

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2210330144



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 动态场景下基于深度学习的
视觉 SLAM 算法研究

论 文 作 者 徐 韬

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: 2210330144

题 目 动态场景下基于深度学习的视觉 SLAM 算法研究

题 目 Research on Visual SLAM Algorithm based on Deep Learning in Dynamic Scene

学生姓名: 徐韬

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 15 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：徐楠

日期：2024年5月25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：徐楠 指导教师签名：王江

日期：2024年5月25日

日期：2024年5月25日

动态场景下基于深度学习的视觉 SLAM 算法研究

摘要

视觉同步定位与地图构建（Visual Simultaneous Localization and Mapping, VSLAM）是指移动机器人在未知环境中通过携带的视觉传感器于运动过程感知外部环境信息并实现自身位置估计和地图构建。然而，当移动机器人面临城市交叉路口、弯曲道路环境等存在许多大曲率弯道运动以及物体间互相遮挡的复杂动态场景时，系统定位精度与建图效果将受到较大影响。为使视觉 SLAM 系统能够应对复杂动态场景，本文主要针对基于深度学习的视觉 SLAM 算法进行深入的理论研究和探索，旨在解决视觉 SLAM 算法在动态场景下遭遇光照变化、图像模糊、尺度变化特征提取与匹配困难、大曲率弯道运动关键帧选取不恰当以及对动态物体分割不完整、难以准确判断潜在动态物体的运动状态等问题。

针对传统视觉 SLAM 算法由于光照变化、图像模糊、尺度变化易导致特征提取与匹配困难降低系统的鲁棒性，以及在大曲率弯道运动下易发生漏选关键帧造成位姿估计不准确、闭环准确性较低等问题，提出一种基于改进关键帧选取的鲁棒视觉 SLAM 算法（Improved keyframe selection-based Robust Visual SLAM algorithm, IR-VSLAM）。该算法在系统前端采用改进 SURF（Speeded Up Robust Features）特征的图像配准算法进行特征提取和匹配，首先，通过局部二维熵进行特征点检测，其次，引入曼哈顿距离构建距离

评价函数进行预处理，最后利用粒子群算法进行匹配优化，增强算法的特征提取与匹配能力；与此同时，针对移动机器人在大曲率弯道运动下丢失有效关键帧的问题，通过创建局部逆向索引窗口并依据窗口内关键帧、当前帧、参考帧三者中的特征点数量、匹配点数量、区域变化空间点数量计算自适应阈值，增补在弯道运动中遗漏的有效关键帧，提升系统在大曲率弯道场景的建图能力。

针对现有动态 SLAM 算法对动态物体分割不完整、难以准确判断潜在动态物体的运动状态等问题，提出一种动态场景下基于改进 Transformer 的视觉 SLAM 算法。该算法通过构建残差坐标注意力模块，引导网络关注漏分割目标，同时融合外部相对位置偏置优化漏分割目标边界语义信息，对潜在动态物体进行准确标记；与此同时，提出一种基于双重约束的物体运动判断方法准确估计潜在动态物体的运动状态，提升算法在动态场景下的定位精度和鲁棒性。

为了验证本文算法的有效性，在公开数据集和真实场景进行测试，结果表明本文算法与 DS-SLAM 算法、Crowd-SLAM 算法以及 DynaSLAM 算法相比绝对轨迹误差 RMSE 分别平均减少了 42.94%、20.79%、11.79%，展现出在复杂动态场景下良好的构图能力。在搭建的移动机器人硬件平台上证明本文算法的可行性，为动态场景下的视觉 SLAM 算法提供重要参考依据。

关键词：动态视觉 SLAM，实例分割，运动判断，关键帧选取，图像配准

RESEARCH ON VISUAL SLAM ALGORITHM BASED ON DEEP LEARNING IN DYNAMIC SCENE

ABSTRACT

Visual Simultaneous Localization and Mapping (VSLAM) refers to the mobile robot in an unknown environment through the visual sensors carried in the movement process to sense the external environment information and achieve its own position estimation and mapping. However, when mobile robots confront complex dynamic settings such as urban intersections and winding road environments, featuring abundant occurrences of large-curvature motion and objects mutually obscuring, the precision of system localization and the effectiveness of mapping are considerably hindered. To enhance the adaptability of visual SLAM systems in coping with intricate dynamic environments, this paper primarily focuses on an extensive theoretical investigation and exploration of deep learning-based visual SLAM algorithms. The main objective is to resolve the issues encountered by visual SLAM algorithms in dynamic scenarios, such as variations in lighting conditions, image blurriness, challenges in feature extraction and matching due to scale variations, inappropriate keyframe selection during large-curvature motion,

incomplete segmentation of dynamic objects, and difficulties in accurately assessing the motion states of potential dynamic objects.

In response to the challenges faced by traditional visual SLAM algorithms, such as difficulties in feature extraction and matching due to variations in lighting conditions, image blurring, and scale changes, resulting in decreased system robustness, as well as issues like inaccurate pose estimation and lower loop closure accuracy, especially during large-curvature motion, an Improved keyframe selection-based Robust Visual SLAM algorithm (IR-VSLAM) is proposed. The algorithm uses an Improved SURF (Speeded Up Robust Features) image alignment algorithm method at the system front end for feature extraction and matching. Initially, feature point detection is conducted using local 2D entropy, followed by the introduction of Manhattan distance to construct a distance evaluation function for preprocessing. Subsequently, the Particle Swarm Optimization algorithm is utilized for matching optimization, thereby enhancing the algorithm's feature extraction and matching capabilities. Concurrently, to address the issue of losing effective keyframes during large-curvature motion of mobile robots, an adaptive threshold is calculated based on the quantity of feature points, matching points, and spatial points of region change among the keyframe, current frame, and reference frame within a locally inverted index window. This adaptive threshold facilitates the supplementation of missed effective

keyframes during curved motion, thereby improving the system's mapping capability in large-curvature scenarios.

In response to the incomplete segmentation of dynamic objects and the challenges in accurately determining the motion states of potential dynamic objects in existing dynamic SLAM algorithms, a visual SLAM algorithm based on improved Transformer in dynamic scenes is proposed. The algorithm uses a residual coordinate attention module to guide the network's focus towards under-segmented targets, while integrating external relative position biases to optimize semantic information at the boundaries of these targets, thus facilitating accurate labeling of potential dynamic objects. Concurrently, a dual-constraint-based approach for object motion assessment is introduced to precisely estimate the motion states of potential dynamic objects, thereby improving the algorithm's localization accuracy and robustness in dynamic scenarios.

In order to verify the effectiveness of the algorithm in this paper, tests on public datasets and real scenarios show that the algorithm in this paper reduces the absolute trajectory error RMSE by an average of 42.94%, 20.79%, and 11.79% compared to the DS-SLAM algorithm, Crowd-SLAM algorithm, and DynaSLAM algorithm, respectively, which demonstrates a good mapping capability in complex dynamic scenes. Additionally, the feasibility of the proposed algorithm was demonstrated on the built mobile robot hardware platform, offering significant insights for the advancement

of visual SLAM algorithms in dynamic environments.

Xu Tao (Electronic Information)

Supervised by Professor Chen Mengyuan

KEY WORDS: Dynamic Visual SLAM, Instance Segmentation, Motion Judgment, Keyframe Selection, Image Alignment

目录

摘要.....	I
ABSTRACT.....	III
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	4
1.2.1 视觉 SLAM 研究现状	5
1.2.2 动态场景下视觉 SLAM 研究现状	7
1.3 论文研究内容与章节安排	8
第 2 章 视觉 SLAM 相关理论与技术	11
2.1 视觉传感器原理	11
2.1.1 相机位姿变换	11
2.1.2 针孔相机模型	12
2.2 SURF 算法与位姿估计	14
2.2.1 SURF 算法	14
2.2.2 位姿估计	17
2.3 动态 SLAM 问题描述	18
2.4 Transformer 结构	20
2.5 本章小结	21
第 3 章 基于改进关键帧选取的鲁棒视觉 SLAM 算法	22
3.1 改进 SURF 图像配准算法	22
3.1.1 基于局部二维熵的特征检测	22
3.1.2 基于距离评价函数的预处理	23
3.1.3 基于粒子群的特征匹配优化	24
3.2 改进关键帧选取策略	25
3.2.1 传统关键帧选取策略	25
3.2.2 自适应阈值关键帧选取策略	26
3.3 实验结果与分析	29
3.3.1 改进 SURF 图像配准对比实验	30
3.3.2 自适应阈值关键帧选取对比实验	32

3.3.3 定位精度对比实验	35
3.4 本章小结	37
第 4 章 动态场景下基于改进 Transformer 的视觉 SLAM 算法	38
4.1 算法整体框架	38
4.2 潜在动态物体分割	39
4.2.1 残差坐标注意力模块	39
4.2.2 融合外部相对位置偏置 Transformer 模块	42
4.3 基于双重约束的物体运动判断	43
4.4 实验结果与分析	44
4.4.1 潜在动态物体分割对比实验	45
4.4.2 基于双重约束的物体运动判断对比实验	46
4.4.3 公开 TUM 数据集评估实验	47
4.4.4 真实场景实验	51
4.5 本章小结	54
第 5 章 全文总结与展望	55
5.1 全文总结	55
5.2 研究展望	56
参考文献	57
攻读学位期间发表的学术论文和科研成果目录	63
致谢	64

第1章 绪论

1.1 研究背景及意义

随着机器视觉、人工智能和深度学习等多领域的蓬勃发展，各种高新技术不断涌现，进一步改善了人类的生产生活方式，推动社会进入了更加智能化的时代。在诸多的高新技术中，机器人技术作为一门多学科交叉融合的新型学科，虽然至今只有几十年的发展历史，但是机器人技术却将人类从繁琐甚至危险的工作中解放了出来、给世界各国工业生产效率带来了极大的提升。与此同时，机器人技术的发展水平在一定程度上代表着一个国家高新技术发展和工业自动化的先进程度，机器人技术以其蕴含的巨大价值受到世界各国的重视并得到了大量的政策支持。我国在《“十四五”机器人产业发展规划》中就曾明确提出要推动机器人产业实现高质量发展，为打造制造业强国、构建健康中国，以及创造美好生活提供坚实支撑。



(a) 海康 Q2-400D 潜伏机器人



(b) 小米“铁蛋”四足机器人



(c) 科沃斯扫地机器人



(d) 中国“祝融号”火星探测车

图 1-1 智能移动机器人终端

目前，移动机器人技术已经在各个领域取得了巨大的发展和应用。如图 1-1 所示，搭载高精度视觉传感器的海康 Q2-400D 潜伏机器人可以在进行外部环境感知和定位的同时根据地面环境信息以及内部控制系统实现自主路径规划、紧急避障等功能，被广泛应用于仓储物流运输等行业；小米“铁蛋”四足机器人采用 AI 交互相机、双目相机、深度摄像模组等多个传感器协作感知当前环境，面对复杂场景适应能力强，甚至可以进行后空翻等高难度动作；科沃斯扫地机器人在室内环境通过视觉、激光等传感器实现地图构建及精准避障。2021 年 5 月我国“祝融号”火星探测车首次在火星成功着陆，由于受地球至火星的距离限制，地面无法实时控制火星车，因此想要在遥远的火星完成科考任务，火星探测车必须具备在复杂且陌生环境中的自主定位与导航能力。不论是潜伏机器人还是火星车，它们都面临一个需要解决的关键技术就是在未知环境中机器人如何感知自身位置信息并同时完成自主导航，其中 SLAM 这一关键技术发挥着至关重要的作用。

“我在哪？我要去哪里？我通过什么方式去？”是自主移动机器人实现智能化必须解决的三个关键科学问题。“我在哪？”即是精确定位问题，后两个便是路径规划问题。目前，移动机器人 SLAM 技术被广泛运用于解决定位精度方面的挑战，它通过搭载的各种传感器在未知环境中感知外部环境信息，从而实现自身位置估计和地图构建，这也是机器人执行导航、人机交互等更高级别任务的先决条件^[1]。目前，根据携带传感器的不同可以将 SLAM 算法分为激光 SLAM 和视觉 SLAM 两大类。激光 SLAM 主要搭载激光雷达、编码器、惯性传感器 IMU（Inertial Measurement Unit）等进行定位导航。激光 SLAM 的优点是稳定性强、精度高、测距范围大，但在粉尘等多干扰、曲率大的弯道运动下表现不佳且硬件成本造价普遍较高^[2]，致使采用激光 SLAM 算法的移动机器人只能在室内等纹理丰富的小范围简单场景取得较好的表现，而在复杂场景下的定位精度和建图能力有待提高。视觉 SLAM 通过搭载相机获取外部图像信息进行自身定位与建图，其使用的相机类型一般包括单目（Monocular）、双目（Stereo）和深度（RGB-D）这三大类。与激光传感器相比，相机可以获取场景中具有丰富纹理与语义特征的图像信息，并且相机造价更加低廉且体积小易于大规模部署使用。与此同时，移动机器人技术研究背后巨大的应用需求以及在城市中面临的开放式复杂环境，均对 SLAM 算法提出较高的鲁棒性需求。视觉 SLAM 注

重对外部无源环境的主动感知、实现动静态障碍物信息的统一采集与处理，能更加灵活地适用于复杂场景。因此，视觉 SLAM 逐渐成为实现移动机器人自主感知、导航以及路径规划的主流研究方向^[3]。

目前，大多数视觉 SLAM 方法能稳定运行的前提条件是对环境进行静态假设，这类 SLAM 系统在视觉信息相对稳定的无人区域、空地环境、月球表面等场景中能够稳定运行并进行准确定位与地图构建，但在城市、商场、酒店等动态物体丰富的场景中，动态物体会极大干扰相机位姿求解，造成构建的地图出现空间几何结构变形扭曲甚至形成拖影最终导致地图可用性急剧下降^[4]。

随着深度学习的发展，研究者们采用结合深度学习的视觉 SLAM 方法，首先采用语义分割网络检测环境中的动态物体，并且标记出所在区域，然后利用运动一致性判断策略筛选动态物体，有效降低了动态物体对 SLAM 系统的干扰^[5]。然而，当视觉 SLAM 系统面临城市交叉路口、弯曲道路环境等存在许多大曲率弯道运动以及物体互相遮挡的复杂动态场景时，系统定位精度与建图效果较差。一方面，由于移动机器人运行于现实世界场景时常伴随光照、尺度变化以及受相机帧率限制易产生模糊图像帧，进而导致视觉 SLAM 系统对图像信息的特征提取与匹配能力不足^[6]；另一方面，由于大曲率弯道运动引起相机视角差急剧变化易造成关键帧选取不恰当，进而导致视觉 SLAM 系统闭环准确率下降，缺乏克服累计误差较大、准确定位困难以及闭环误差修正不完全的能力^[7]；再一方面，由于物体间相互遮挡造成物体在相机视野中产生不完整外观造成漏分割以及难以准确判断潜在动态物体的运动状态，进而降低系统精度^[8]。因此，复杂动态场景下图像配准、关键帧选取、不完整外观的动态物体识别以及特征点运动判断是目前亟需解决的难题。

本文以增强基于视觉的移动机器人算法在复杂动态场景下的定位准确性和鲁棒性为目标，开展了动态场景下基于深度学习的视觉 SLAM 算法研究，着重研究了基于改进关键帧选取的鲁棒视觉 SLAM 算法以及动态场景下基于改进 Transformer 的视觉 SLAM 算法。在特征提取与匹配和关键帧选取环节，重点研究了基于改进 SURF 特征的图像配准方法和随曲率变化的关键帧选取阈值调控机制；在潜在动态物体分割与运动判断环节，着重研究了如何构建能准确分割由于遮挡导致的外观缺失物体的实例分割网络模型以及准确判断潜在动态物体运动状态的有效方法。

1.2 国内外研究现状

自 1986 年科学家首次提出 SLAM 理论以来^[9]，历经近四十年的发展，SLAM 获得了巨大的发展和应用，并逐渐在日常学习生活领域、自动驾驶和航天航空等科研探索领域以及工业生产等领域扮演着重要角色。早期的视觉 SLAM 大多是以卡尔曼滤波算法、粒子滤波算法等为代表的基于滤波的方法实现^[10-13]。基于滤波的方法能较好的处理环境信息与传感器的关系，因此性能表现较好。然而，由于此类方法假设了 Markov 性，即认为 T 时刻的状态与之前的状态无关，而只和上一时刻相关，这就导致算法仅能将当前帧与上一图像帧进行匹配，而无法将当前帧和上一帧之前的图像帧进行匹配，也就是无法检测闭环，这将使得算法随着时间的长期运行发生无法修正的累计误差漂移。随着计算机视觉技术的飞速发展，搭载视觉传感器获取外界环境信息的视觉 SLAM 方法已经成为当前研究的重点。经典的视觉 SLAM 框架共五个部分，分别是传感器信息获取、前端视觉里程计、后端（非线性）优化、地图构建以及闭环检测，如图 1-2 所示^[14]。传感器信息获取主要利用携带的相机获取外部环境信息并进行图像预处理，为 SLAM 后续的进程提供数据来源。前端视觉里程计（Visual Odometry, VO）^[15]主要采用多视图几何^[16]的方法利用相邻图像帧之间的关联信息进行相机位姿估计，同时创建局部地图。闭环检测主要用来判断移动机器人是否处于之前访问过的熟悉场景，若检测到闭环，则将数据输入后端进行优化，可有效减少长时间运行导致的位姿估计累计误差。后端（非线性）优化是指将前端视觉里程计所估计的相机位姿与闭环检测信息进行数据关联从而优化局部或全局地图，最终建立全局一致性轨迹和地图。建图即根据用户需求并利用前端及后端优化获取的数据建立相应环境地图。

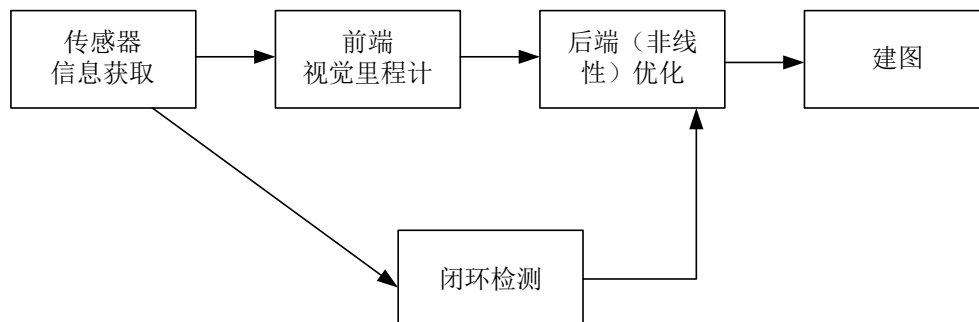


图 1-2 经典视觉 SLAM 框架

1.2.1 视觉 SLAM 研究现状

早在 2007 年, Davison 等人^[17]就提出了第一个基于特征点的视觉 SLAM 算法 MonoSLAM, 该算法通过进行图像相似度的计算判断闭环是否发生, 在一定程度上修正了位姿并减小地图累计误差, 但是该方法计算量较大实时性有所不足。同年, Klein 等人^[18]首次提出了跟踪与建图线程并行的 PTAM (Parallel Tracking and Mapping) 算法, 该算法打破用滤波进行后端优化的传统思路, 创造性地提出将图像帧序列中具有代表性的关键帧进行串联而不是对每一帧图像都进行处理, 最终通过非线性 BA (Bundle Adjustment) 优化实时获取图像关键帧的姿态和轨迹, 为解决大规模场景视觉 SLAM 算法实时性不足的问题奠定了基础。但是该算法采用的 FAST (Feature from Accelerated Segment Test)^[19]算法在面对发生旋转、尺度和光照变化情况时鲁棒性较差, 使得 SLAM 系统可适应场景小, 且易发生跟踪丢失。2015 年, Mur-Artal 等人^[20]在 PTAM 算法的基础上增加了闭环检测线程, 提出了一种基于 ORB (Oriented FAST and Rotated BRIEF) 特征^[21]的 ORB-SLAM 算法, 得益于 ORB 算法良好的尺度和旋转不变性, 使得 ORB-SLAM 算法在面对旋转、平移与缩放等环境所提取的特征具有较高的匹配准确率, 从而增强了系统的定位精度和鲁棒性, 此外 ORB-SLAM 算法还首次提出跟踪、建图和闭环检测三个线程并行作业, 实现了 SLAM 系统的准确定位与建图。ORB-SLAM 算法框架如图 1-3 所示。

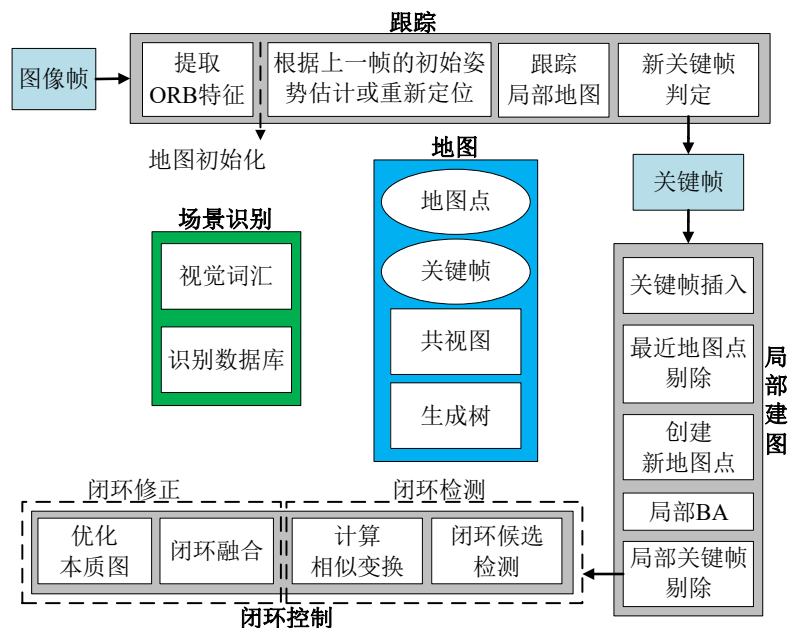


图 1-3 ORB-SLAM 算法框架

2017 年 Engel 等人^[22]提出了 DSO (Direct Sparse Odometry) 算法, 该算法在速度和准确性上都比 ORB-SLAM 更好, 特别是在缺少特征点区域仍然具有较强的鲁棒性, 但是当发生快速运动、图像模糊时易产生跟踪丢失。2017 年, Mur-Artal 等人^[23]对 ORB-SLAM 算法进行改进并提出了 ORB-SLAM2 算法, 该算法通过在闭环检测之后增加全局 BA 从而提升建图精度。2020 年, Campos 等人^[24]为增强 ORB-SLAM2 在更多场景稳定运行的能力, 提出了支持多子地图系统的 ORB-SLAM3 算法。该算法无论是在室外大场景还是室内小场景均展现出较强的鲁棒性和较高实时性, 定位精度与 ORB-SLAM2 相比提高了 2~5 倍。

自关键帧选取策略被提出以来, 科研工作者对其做了大量的研究。Engel 等人^[25]提出的 LSD-SLAM 算法通过比较当前关键帧和图像帧的加权距离与所设定距离阈值的大小决定是否选取关键帧, 该方法虽然计算简单但灵活性较差。为了克服这一问题, Leutenegger 等人^[26]提出了基于图像重叠的关键帧选择方法, 当重叠区域的匹配点小于检测点的 50%~60%时选取关键帧, 同时边缘化最远的关键帧, 从而保留最近的帧集合和另一组关键帧。Qin 等人^[27]提出了跟踪质量与平均视差的双重标准进行关键帧的选取, 若跟踪特征的数量小于一定阈值或跟踪特征的平均视差大于一定的阈值, 则将该图像帧确定为新关键帧。虽然图像重叠、平均视差和跟踪质量的关键帧选取方法效果较好, 但是由于需要重复进行特征提取和匹配、协方差以及视差计算从而导致系统实时性较差。为了平衡关键帧选取的灵活性和实时性, ORB-SLAM2 算法通过尽可能快地插入关键帧并去除后面的冗余帧, 以避免额外的开销, 从而增强在困难场景下的鲁棒性。张等人^[28]提出基于相机方向和角度相对变化量的关键帧选择方法, 若相机姿态急剧变化, 则关键帧选择速率增加, 若相机姿态稍微偏移, 则以较低速率选取关键帧。该方法通过减少 40%~60%的冗余帧提高了算法的速度, 同时不降低定位精度, 但却未考虑移动机器人运行于大曲率弯道运动场景易发生漏选关键帧的问题。大曲率弯道运动示意图如图 1-4 所示。 A_1B_1 、 A_2B_2 分别为两段曲率半径为 ρ_1 、 ρ_2 (对应的曲率为 K_1 、 K_2) 的运动轨迹, 且 $\rho_1 > \rho_2$ 、 $K_1 < K_2$ 。横向偏差 d_r 为移动机器人前轴中心指向障碍物边界的恒定距离, 其指向的点为跟踪点, v 为车辆的纵向速度。从现阶段的研究成果来看, 移动机器人无法沿曲率为 K_2 的 A_2B_2 轨迹运动, 以 $v=1\text{m/s}$ 的纵向速度沿曲率为 K_1 的 A_1B_1 轨迹运动且

$d_r < 700\text{mm}$ 时易出现关键帧丢失的问题。近年来，相关文献也指出如何在大曲率弯道运动下实现全局一致性建图仍是研究热点和亟待解决的难点^[29]。

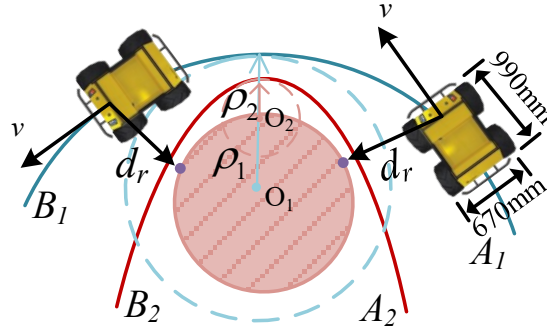


图 1-4 大曲率弯道运动示意图

为了克服大曲率弯道运动下易发生关键帧丢失的问题，陈孟元等人^[30]在传统顺向选取关键帧的基础上首次提出了逆向索引关键帧策略。该策略在顺向选取关键帧的同时，以当前关键帧作为初始点进行反向索引，并计算处于当前关键帧和前一个关键帧之间全部的普通帧与当前关键帧之间的旋转角度，进而将此旋转角度与设定阈值进行比较，若其大于设定阈值则将此普通帧重新增补为曲线运动遗漏的关键帧，提高了算法在曲线运动下的定位精度，但该方法采用的固定阈值难以适应曲率变化情况，当场景中存在曲率变化较大情况时，固定阈值难以准确选取关键帧，进而降低系统闭环准确率和定位精度。

1.2.2 动态场景下视觉 SLAM 研究现状

上述大多 SLAM 方法都是建立在静态环境的假设上。然而，在现实环境中必然存在光照变化和动态物体（如行人、车辆等），其中，光照变化将影响特征提取与匹配的准确性，而动态物体可能会对 SLAM 中的特征识别和数据关联产生显著干扰，从而极大降低系统的定位精度^[31]。因此，如何实现视觉 SLAM 系统在动态场景下的准确性定位和一致性建图已成为了一个重要的课题^[32]。

目前针对动态场景的 SLAM 方法主要分为三类：几何方法、深度学习方法以及几何与深度学习结合的方法。在几何方法中，艾青林等人^[33]通过对特征点状态进行划分，然后采用严格的几何约束筛选静态点从而增强算法在动态场景下适应性。Li 等人^[34]选择深度边缘点来寻找对应关系，并设计了一种静态加权方法来降低动态点的影响。在深度学习方法中，通常采用深度学习中的目标识

别和分割技术。Liu 等人^[35]在 ORB-SLAM3 的基础上增加了基于语义的跟踪和优化线程，减少了动态对象对系统的干扰并进行实时跟踪和地图构建。Soares 等人^[36]提出了一种使用目标检测网络在人群拥挤环境中进行定位和建图的 Crowd-SLAM 算法，该算法将语义动态检测框内的所有特征点均视为动态点并去除，在一定程度上减少了动态物体的干扰，但此种方法同样未充分考虑边界框内的静态特征，从而易发生跟踪丢失。在几何与深度学习相结合的方法中，由目标识别或语义分割网络提供有关动态对象的语义先验信息^[37]，随后通过几何约束进一步细化以过滤动态对象出现的特征。Xiao 等人^[38]提出 Dynamic-SLAM 算法，首先利用 SSD 网络对环境中物体进行检测，然后提出一种选择性跟踪算法对动态对象的特征点进行处理，以降低因匹配错误导致的姿态估计误差，但该方法难以有效判断特征点运动状态。为了解决以上问题，Wu 等人^[39]提出了一种轻量级目标检测网络 Darknet19-YOLOv3 获取动态物体的语义信息，然后采用 Depth-RANSAC 算法筛选动态点提高系统的实时性和精度。为了进一步提升动态场景下 SLAM 系统的定位精度，Yu 等人^[40]提出了 DS-SLAM 算法，该算法结合 SegNet 网络和运动一致性判断策略剔除动态物体，但当动态物沿极线方向运动时，该方法也难以准确判断物体运动状态^[41]。Bescos 等人^[42]提出了 DynaSLAM 算法，使用实例分割网络 Mask R-CNN 将运动物体（行人和车辆等）精准分割，并通过多视角几何方法计算前后帧中同一个特征点的变换角度，判断物体是否发生运动，但在跟踪特征点的过程中视差角易被噪声影响导致部分物体运动状态判断不准确。

1.3 论文研究内容与章节安排

本文以增强复杂动态场景下基于视觉的移动机器人算法的定位准确性和鲁棒性为目标，开展了动态场景下基于深度学习的视觉 SLAM 算法研究，着重研究了基于改进关键帧选取的鲁棒视觉 SLAM 算法以及动态场景下基于改进 Transformer 的视觉 SLAM 算法。在特征提取与匹配环节，重点研究了基于改进 SURF 特征的图像配准方法；在关键帧选取环节，着重研究了随曲率变化的关键帧选取阈值调控机制；在潜在动态物体分割与运动判断环节，着重研究了如何构建能有效分割外观缺失物体的实例分割网络模型以及准确判断潜在动态物体

运动状态的有效方法。论文总体技术路线如图 1-5 所示。

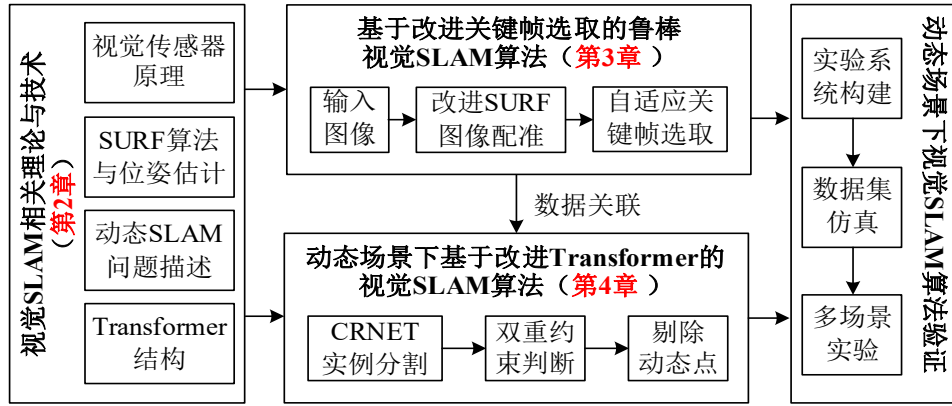


图 1-5 总体技术路线

各章节主要内容如下：

第一章，绪论。介绍了研究的背景、意义及现状。首先介绍了目前移动机器人应用的领域以及移动机器人 SLAM 研究的意义，分析总结了视觉 SLAM 和动态场景下视觉 SLAM 的国内外研究现状，然后对本文主要内容进行了总结，最后对本文内容做了详细介绍。

第二章，视觉 SLAM 相关理论与技术。主要介绍视觉 SLAM 相关原理。首先，通过对视觉传感器原理、SURF 算法与位姿估计进行解析，详细描述了视觉 SLAM 如何通过相机获取外部世界特征点信息并进行位姿估计。其次，从基于优化的视觉 SLAM 算法对动态 SLAM 问题进行描述，分析动态物体对 SLAM 系统产生的影响，最后介绍了 Transformer 结构，为本文后续的动态视觉 SLAM 算法研究奠定理论基础。

第三章，基于改进关键帧选取的鲁棒视觉 SLAM 算法。本章提出改进 SURF 图像配准方法，该方法采用局部二维熵提高特征点的独特性，同时引入曼哈顿距离构建距离评价函数进行预处理，剔除冗余的特征点，增强特征点分布的均匀性，并利用粒子群算法进行特征匹配优化，加速匹配过程，增强算法在光照变化、图像模糊和尺度变化场景特征提取与匹配的准确率和实时性。然后，提出自适应阈值关键帧选取策略，结合角度阈值与自适应阈值进行关键帧选取，补充大曲率弯道运动下遗漏的有效关键帧，提升系统在大曲率弯道场景下的全局一致性建图能力。最后，在公开数据集对本章算法进行了验证。

第四章，动态场景下基于改进 Transformer 的视觉 SLAM 算法。提出了一种潜在动态物体分割网络 CRNET (Coordinate attention and Relative position bias

NETwork), 该网络通过构建残差坐标注意力模块, 引导网络关注漏分割目标, 同时融合外部相对位置偏置优化漏分割目标边界语义信息, 对潜在动态物体进行准确标记; 然后, 提出基于双重约束的物体运动判断策略实现对物体运动状态的准确估计并剔除动态点。最后, 在公开的 TUM 动态数据集进行仿真实验并利用自主搭建的移动机器人实验平台进行真实场景测试, 验证了本文算法的有效性。

第五章, 全文总结与展望。对本论文开展的研究内容和工作进行总结, 分析本文所提算法的优缺点, 并对今后的研究计划进行展望。

第2章 视觉 SLAM 相关理论与技术

本章将介绍本文开展动态场景下基于深度学习的视觉 SLAM 算法研究所涉及的视觉 SLAM 理论与技术，主要包括视觉传感器原理、SURF 算法与位姿估计、动态 SLAM 问题描述以及 Transformer 结构相关内容。

2.1 视觉传感器原理

视觉传感器可以将真实世界的观测信息转换为图像信息，如图 2-1 所示。首先，使用旋转和平移矩阵将三维空间点从世界坐标系映射至相机坐标系。接着，根据相机小孔成像原理进行比例缩放，利用相机内参矩阵将相机坐标系下的点映射到图像坐标系，得到畸变图像坐标。最后，经过平移和量化缩放的处理，将畸变图像坐标转换为像素坐标系中的像素坐标。

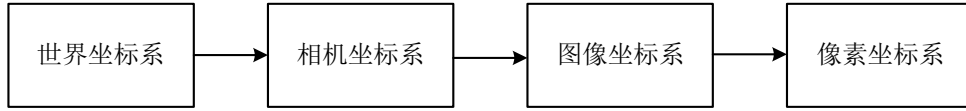


图 2-1 空间点从世界坐标系转换为像素坐标流程图

2.1.1 相机位姿变换

在视觉 SLAM 中通常采用欧式变换来表示相机的运动。假设存在世界坐标系 O_w ，相机坐标系 O_c ，如图 2-2 所示。假设空间点 A 在世界坐标系 O_w 的坐标为 $(a'_1, a'_2, a'_3)^T$ ，在相机坐标系 O_c 的坐标为 $(a_1, a_2, a_3)^T$ 。单独考虑旋转则有：

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

其中， $[e_1, e_2, e_3]$ 和 $[e'_1, e'_2, e'_3]$ 分别为相机和世界坐标系下的单位正交基，将式 (2-1) 两边同时左乘 $[e_1^T, e_2^T, e_3^T]^T$ 得到：

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

其中，将等号右边定义为旋转矩阵。考虑相机运动会伴随平移运动，则相机的

运动可以表示为:

$$a_c = \mathbf{R}a_w + \mathbf{t} \quad (2-3)$$

其中, $\mathbf{R}$ 为旋转矩阵, $\mathbf{t}$ 为平移向量, a_w 和 a_c 分别为变换前后的坐标。为了统一 $\mathbf{R}$ 和 $\mathbf{t}$ 的维度, 可利用齐次坐标简化上式:

$$\begin{bmatrix} a_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} a_w \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} a_w \\ 1 \end{bmatrix} \quad (2-4)$$

其中, $\mathbf{T}$ 为变换矩阵, 用来描述坐标系之间的转换关系。至此就完成了点 A 从世界坐标系到相机坐标系的转换。 $\mathbf{R}$ 和 $\mathbf{t}$ 表示相机的运动信息, 它们会随相机运动而不断更新, 得到它们也就获得了相机运动的轨迹, 因此也将其称之为相机位姿。

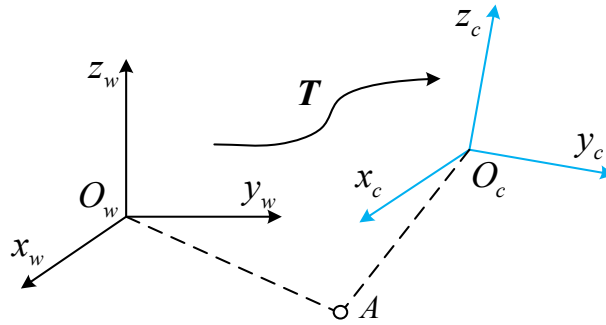


图 2-2 坐标系间的欧式变换

2.1.2 针孔相机模型

相机将空间点映射到二维图像平面的过程如图 2-3 所示。 (x_w, y_w, z_w) 为世界坐标系 O_w , (x_c, y_c, z_c) 为相机坐标系 O_c , (x, y) 为图像坐标系, 单位均为米。 (u, v) 为像素坐标系, 单位为像素。 A' 为空间点 A 在图像坐标系上成像点。

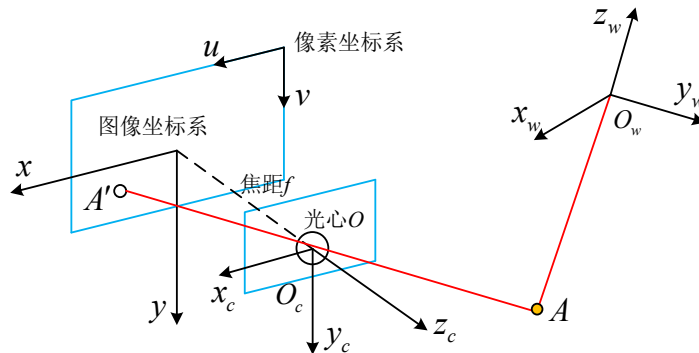


图 2-3 针孔相机模型

假设 a_w 为空间点 A 在 O_w 的坐标, 由 2-4 式可得到 a_w 在 O_c 的坐标 a_c 为:

$$a_c = T_{cw} a_w \Rightarrow \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2-5)$$

根据针孔相机模型的几何关系可将空间点在 O_c 的坐标映射至图像坐标系，如公式2-6所示：

$$\frac{z_c}{f} = \frac{x_c}{x} = \frac{y_c}{y} \quad (2-6)$$

整理后得到：

$$\begin{aligned} x &= f \frac{x_c}{z_c} \\ y &= f \frac{y_c}{z_c} \end{aligned} \quad (2-7)$$

由公式（2-7）可得空间点 A 在图像坐标系下的坐标，单位为米，接着对其进行采样和量化转换为像素坐标。计算方式如下：

$$\begin{aligned} u &= \alpha x + c_x \\ v &= \beta y + c_y \end{aligned} \quad (2-8)$$

其中， u ， v 分别为像素坐标系的坐标轴， α 和 c_x 分别为 u 轴上的缩放系数和平移量， β 和 c_y 分别为 v 轴上的缩放系数和平移量。将公式（2-7）和（2-8）进行合并，即可得到相机坐标系转换为像素坐标系计算公式：

$$\begin{cases} u = \alpha f \frac{x_c}{z_c} + c_x \\ v = \beta f \frac{y_c}{z_c} + c_y \end{cases} \Rightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \stackrel{def}{=} \frac{1}{z_c} K a_c \quad (2-9)$$

其中， K 为相机内参，通常由相机厂家设置，也可以采用张正友标定法^[43]进行标定得到。矩阵 K 中的 f_x 和 f_y 分别由 αf 和 βf 合并得到。最后将公式（2-5）代入公式（2-9）即可得到世界坐标系到像素坐标系的转换公式，如式（2-10）所示，其中 T 被称作相机的外参矩阵。

$$a = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z_c} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \stackrel{def}{=} \frac{1}{z_c} K a_c = \frac{1}{z_c} K T a_w \quad (2-10)$$

2.2 SURF 算法与位姿估计

位姿估计是视觉 SLAM 实现定位与建图的关键环节，本文采用 SURF 特征点进行特征提取与匹配，然后利用匹配点进行位姿估计，因此本节对 SURF 算法和位姿估计进行介绍。

2.2.1 SURF 算法

(1) 提取特征点

SURF 算法通过引入积分图像技术，从根本上改善了其运行效率，这使得其与 SIFT 算法相比时表现出明显的速度优势。具体而言，SURF 算法通过将原始图像与高斯二阶微分模板的滤波操作转化为积分图像的加减运算过程，实现了更为高效的特征提取。这一改进显著提升了算法在搜索和检测特征点时的速度，从而使得整体计算效率得到了显著提升。

积分图像的计算方法如图 2-4 所示，假设在图像中存在一个像素点 A ， A 点也可用其像素坐标表示为 $I(x, y)$ ，则 A 点的积分图像 $I_{\Sigma}(x, y)$ 可表示为 A 点和原点 O 所围成的橙色区域内的像素点灰度值之和，其计算公式如下：

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i < x} \sum_{j=0}^{j < y} I(x, y) \quad (2-11)$$

求取积分图后，计算矩形区域 Σ 的像素灰度和 I_{Σ} 时，只需把该区域的四个顶点 A 、 B 、 C 、 D 的积分图像值 $I_{\Sigma}(A)$ 、 $I_{\Sigma}(B)$ 、 $I_{\Sigma}(C)$ 、 $I_{\Sigma}(D)$ 进行加减运算：

$$I_{\Sigma} = I_{\Sigma}(A) - I_{\Sigma}(B) + I_{\Sigma}(C) - I_{\Sigma}(D) \quad (2-12)$$

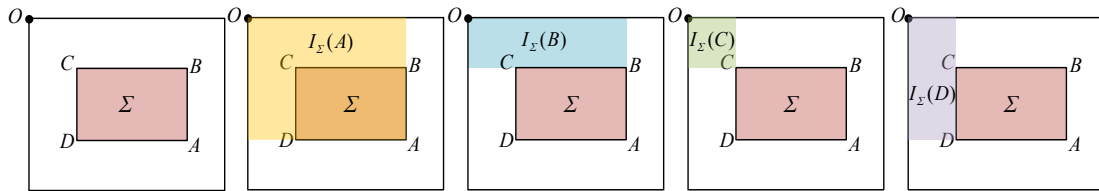


图 2-4 积分图像计算图

SURF 算子基于尺度空间检测，并使用高斯滤波 Hessian 矩阵来提取点特征。具有尺度 σ 的像素点 $I(x, y)$ 的 Hessian 矩阵被定义为：

$$\mathbf{H}(l, \sigma) = \begin{bmatrix} L_{xx}(l, \sigma) & L_{xy}(l, \sigma) \\ L_{xy}(l, \sigma) & L_{yy}(l, \sigma) \end{bmatrix} \quad (2-13)$$

其中， $L_{xx}(l, \sigma)$ 、 $L_{xy}(l, \sigma)$ 和 $L_{yy}(l, \sigma)$ 分别表示在点 $l(x, y)$ 处，高斯二阶偏导数

$\frac{\partial^2 g(\sigma)}{\partial x^2}$, $\frac{\partial^2 g(\sigma)}{\partial xy}$, $\frac{\partial^2 g(\sigma)}{\partial y^2}$ 和图像 I 做卷积。

为了提高算法的实时性。SURF 使用盒式滤波器（Box filters）近似替换高斯滤波器（Gaussian filter），并通过引入权重因子 w 来减少引入盒式滤波器导致的误差， w 通常取 0.9，则 Hessian 矩阵可近似简化为：

$$\det(\mathbf{H}) = D_{xx} * D_{yy} - (0.9 * D_{xy})^2 \quad (2-14)$$

其中， $\det(\mathbf{H})$ 表示点 l 邻域的滤波响应， D_{xx} 、 D_{xy} 和 D_{yy} 分别为使用图 2-5 所示盒式滤波器在 X、Y 和 Z 方向的估计值。

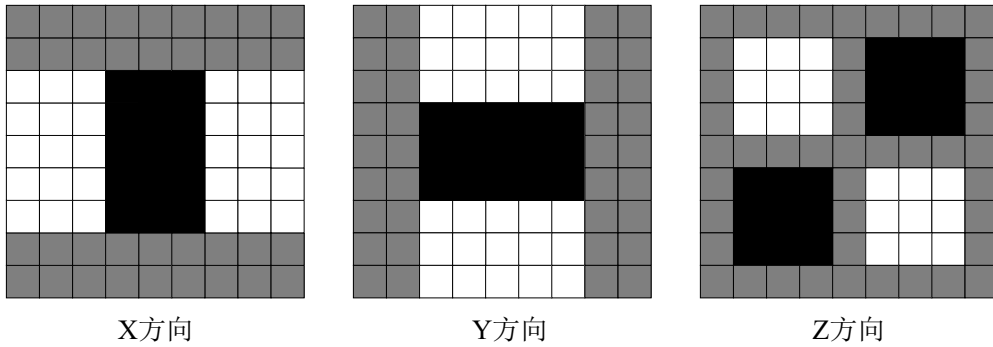


图 2-5 盒式滤波器示意图

为了增强算法对尺度变化的鲁棒性。SURF 算法构造了图像金字塔进行特征点的提取，具体方法是通过计算不同尺寸滤波器下 Hessian 矩阵的响应值，从而获得各个尺度的响应图像。SURF 的尺度空间由多组（O 组）多层（L 层）组成，这种金字塔结构图展示了其层次化的特征提取过程，确保了算法在不同尺度下都能有效工作。传统金字塔和 SURF 金字塔尺度空间对比图如图 2-6 所示。

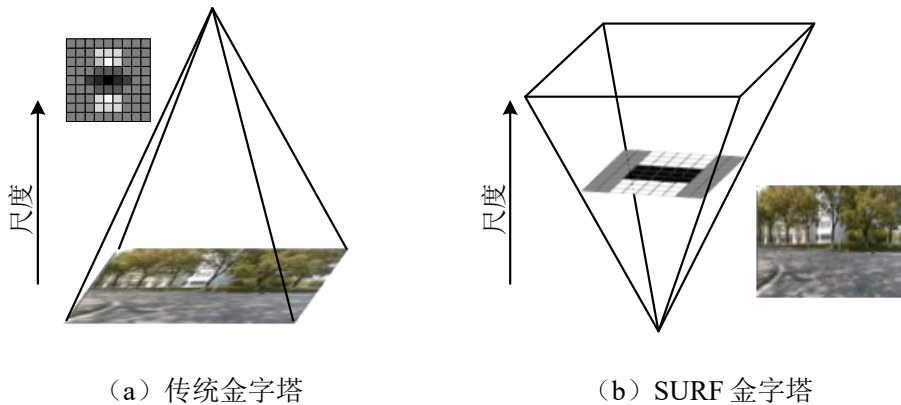


图 2-6 尺度空间对比

SURF 算法通过邻域非极大值抑制技术，精确地确定特征点的位置。通过将像素点与其空间邻域内的点进行对比从而找出局部极值点并将其确定为候选特

征点，如图2-7所示。然后，采用三维线性插值获取候选点精确空间位置以及尺度信息，最终完成特征点的提取，为后续的特征描述和匹配打下了坚实的基础。

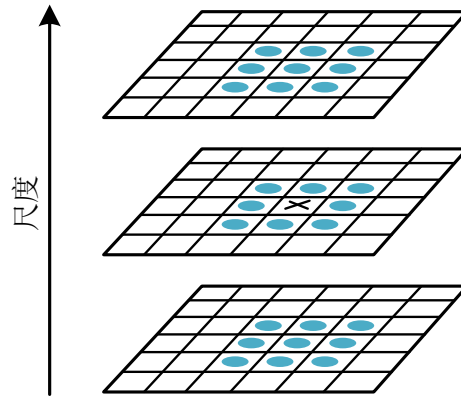


图 2-7 极值点检测示意图

(2) 确定特征点方向

为了实现算法的旋转不变性。首先，以被检测到的特征点中心为圆心，以 $6s$ 为半径（ s 为特征点所在尺度值）选取一个圆形邻域，其次，使用角度为 60° 的扇形区域对圆形邻域进行扫描，并计算扇形区域内全部点的 x （水平）和 y （垂直）方向 Haar 小波响应矢量和^[44]，最终，特征点主方向与矢量和最长的方向一致。这一过程有效地为每个特征点分配了一个主方向，如图2-8所示。

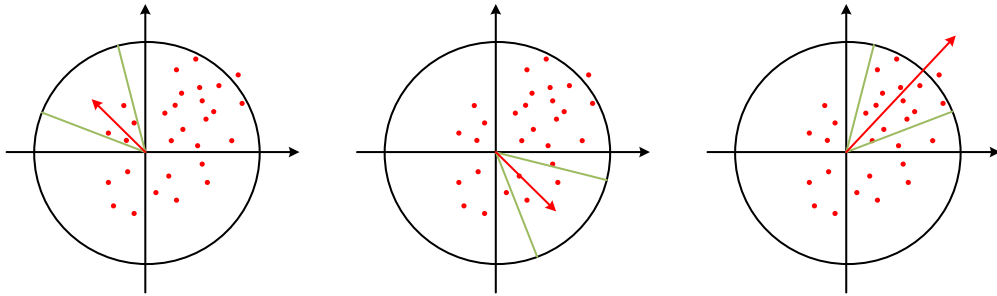


图 2-8 特征点主方向示意图

(3) 构建特征描述子完成匹配

特征点主方向被确定之后，即可开始构建描述子，如图2-9所示。以该特征点为中心、 $20s$ 为边长，构建正方形区域，该区域被平均分为 $4 \times 4 = 16$ 个子区域，每一个子区域含有 $5 \times 5 = 25$ 个像素。在每个子区域中，计算 25 个像素点的 x （水平）和 y （垂直）方向（以确定的主方向为参照）的 Haar 小波响应 d_x 、 d_y 。并采用高斯函数对小波响应进行加权处理从而提升算法对几何形变和位置误差的鲁棒性，最终将每个子区域的 $\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|$ 合并成向量，组成

本质矩阵 $\mathbf{E}$ 反向求解旋转矩阵 $\mathbf{R}$ 和平移矩阵 $\mathbf{t}$ 。从理论上来看，利用五组特征匹配点即可求解自由度为 5 的本质矩阵 $\mathbf{E}$ ，但是由于本质矩阵 $\mathbf{E}$ 具有非线性，仅采用五点法求解较为困难，因此有学者提出采用八点法求解本质矩阵。

求出本质矩阵 $\mathbf{E}$ 后即可利用 SVD (Singular Value Decomposition) 方法恢复相机位姿 $\mathbf{R}$ 和 $\mathbf{t}$ 。假设 SVD 分解为：

$$\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (2-16)$$

其中， $\mathbf{U}$ 和 $\mathbf{V}$ 为正交矩阵， $\mathbf{\Sigma}$ 为奇异值矩阵。此时，对于任意的本质矩阵 $\mathbf{E}$ 可能存在的解如下：

$$\begin{aligned} \mathbf{t}_1^{\wedge} &= \mathbf{U}\mathbf{R}_z\left(\frac{\pi}{2}\right)\mathbf{\Sigma}\mathbf{U}^T, & \mathbf{R}_1 &= \mathbf{U}\mathbf{R}_z^T\left(\frac{\pi}{2}\right)\mathbf{V}^T \\ \mathbf{t}_2^{\wedge} &= \mathbf{U}\mathbf{R}_z\left(-\frac{\pi}{2}\right)\mathbf{\Sigma}\mathbf{U}^T, & \mathbf{R}_2 &= \mathbf{U}\mathbf{R}_z^T\left(-\frac{\pi}{2}\right)\mathbf{V}^T \end{aligned} \quad (2-17)$$

又因为 $\mathbf{E}$ 和 $-\mathbf{E}$ 等价，所以对任意的 $\mathbf{t}$ 取负号计算的结果都相同。因此， $\mathbf{E}$ 经过 SVD 分解后会得到四组可能解，如图 2-11 所示。最后通过将点代入四组解中验证其在相机下的深度值来确定相机位姿 $\mathbf{R}$ 和 $\mathbf{t}$ 。

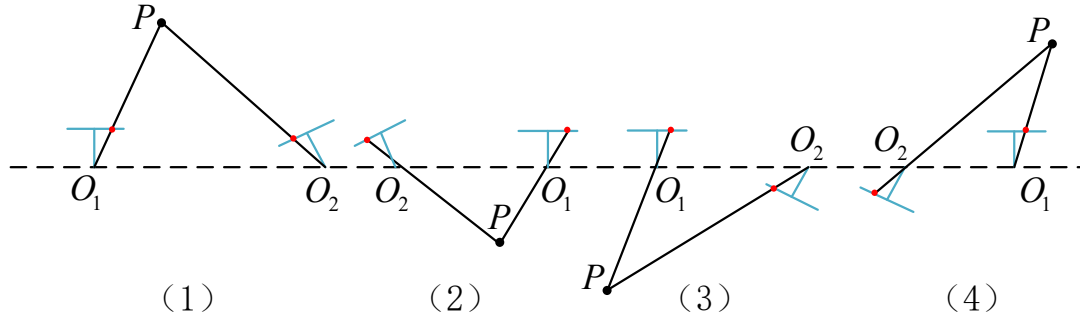


图 2-11 分解本质矩阵所得的四组解

2.3 动态 SLAM 问题描述

在基于优化的视觉 SLAM 算法中，两帧图像进行特征匹配之后，会采用 BA 方法对 3D 空间点和 6 自由度的相机位姿进行同步优化，目标函数为：

$$(\mathbf{T}^*, p^*) = \arg \min_{(\mathbf{T}, p)} \sum_{i=1}^m \sum_{j=1}^n \|z_{i,j} - h(\mathbf{T}_{i,i-1}, p_j)\|^2 \quad (2-18)$$

其中， $z_{i,j}$ 表示通过特征提取与匹配获取的空间点 p_i 在第 i 帧图像中的像素坐标， $h()$ 反映的是空间点 p_j 经过变换矩阵 $\mathbf{T}_{i,i-1}$ 变换后在第 i 帧图像中的像素坐标预测值，具体来说就是求解使得重投影误差达到最小时的变换矩阵 $\mathbf{T}_{i,i-1}$ 。

尽管 BA 优化可以减小系统误差，但该方法的有效性建立在一个关键前提之上，即所匹配的特征点必须属于静态目标。因此当环境中存在动态物体（如行走的人，运行的车辆等）且动态物体在图像中所占比例较高时，将会使得目标函数的优化过程沿着错误的方向进行，最终会导致机器人出现极大的轨迹漂移，并严重影响系统定位精度。为了更直观地理解动态物体的干扰，借助文献[5]的动态场景下投影及图优化模型来进行描述，如图 2-12 所示。

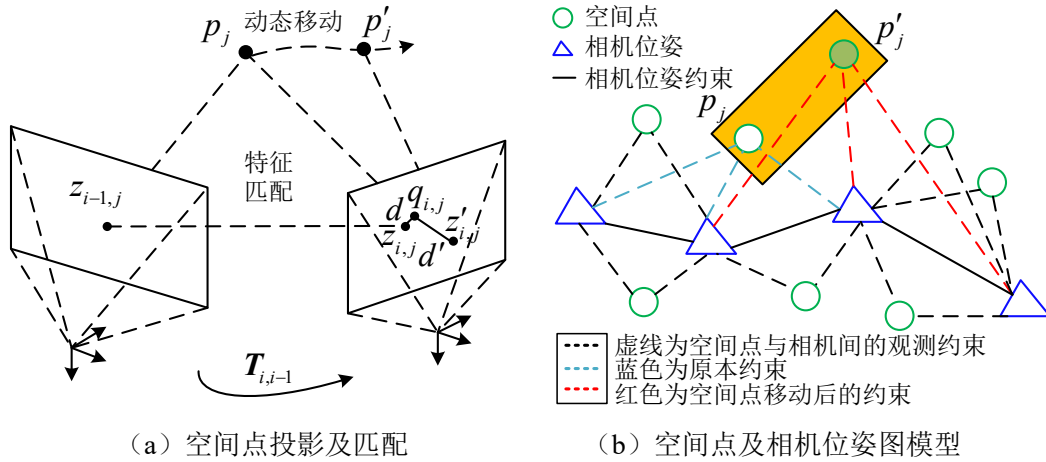


图 2-12 动态场景下投影及图优化模型^[5]

在图 2-12 (a) 中，上一帧中像素点 $z_{i-1,j}$ 通过特征匹配得到在当前帧中的像素点位置 $z_{i,j}$ ， $q_{i,j}$ 为采用 BA 优化得到的 $z_{i,j}$ 所对应的预测值， $q_{i,j}$ 与 $z_{i,j}$ 之间偏差为 d ，通过优化 $T_{i,i-1}$ 使偏差 d 减小，从而提升系统精度。然而，当空间点 p_j 属于动态物体时， p_j 运动到 p'_j 后， $z_{i-1,j}$ 在当前帧中匹配点变成了 $z'_{i,j}$ ，若此时仍然采用变换矩阵 $T_{i,i-1}$ 进行预测，则预测点位置变为了 $q_{i,j}$ ，此时 $q_{i,j}$ 与 $z'_{i,j}$ 之间的偏差变成了更大的 d' ，BA 优化沿着减小 d' 的方向前进，这将导致最终的 $T_{i,i-1}$ 不会是最优解。与此同时，可通过图模型表示优化过程，如图 2-12 (b) 所示，当空间点 p_j 运动至 p'_j ，不仅会改变空间点与相机原本的约束，还可能增加（减少）新约束，这将会对信息矩阵的结构以及迭代优化产生影响，同时新发生（消失）的约束还将会对闭环检测产生极大影响。因此，大多数学者通过在相机位姿估计之前剔除图像帧中的动态区域以避免动态物体影响相机位姿估计的准确性，从而提升系统定位精度。目前较为普遍的方法是采用实例分割网络对物体进行分割，然后利用先验信息区分动态物体，并采用运动判断方法进一步判断物体的运动状态并剔除动态点，最终采用静态点进行位姿估计和优化。

2.4 Transformer 结构

自 2017 年 Vaswani 等人^[45]首次提出 Transformer 网络以来，其已经成为自然语言处理和其他序列建模任务中的重要工具。该模型的创新之处在于引入了自注意力机制，这使得模型能够同时处理长距离依赖性，而无需像传统的循环神经网络（Recurrent Neural Network, RNN）或长短期记忆网络（Long Short-Term Memory, LSTM）那样依赖于顺序处理。自注意力机制使得 Transformer 能够并行计算，极大地提高了训练效率，使其成为处理大规模数据集的理想选择。除了在自然语言处理领域取得了巨大成功之外，Transformer 模型还被广泛应用于图像处理领域，特别是对场景中的动态物体进行识别分割取得了较好的效果。Transformer 模型结构示意图如图 2-13 所示。

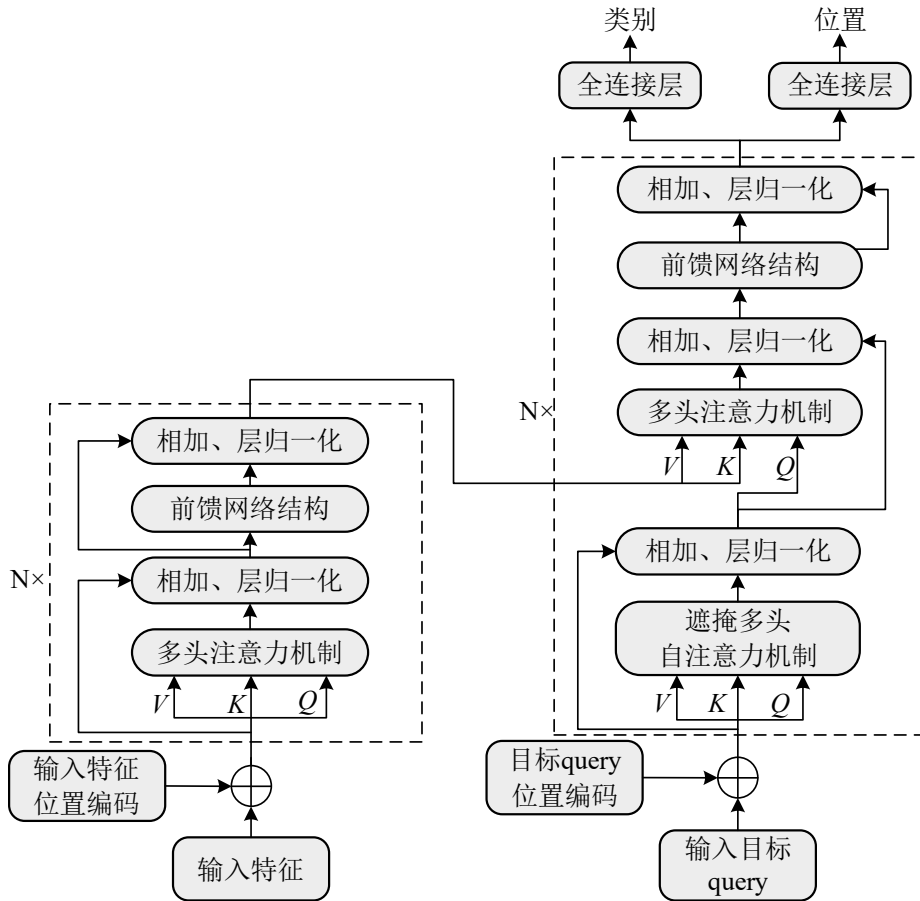


图 2-13 Transformer 模型结构示意图

多头注意力机制层以点积方式将多个注意力层进行组合实现。其输入由维度为 d_k 的查询（Query）和键（Key）以及维度为 d_v 的值（Value）组成。通过计算 Query 和 Key 的点积并除以 $\sqrt{d_k}$ 以避免点积运算导致计算量过大的问题，

接着经过 soft max 对注意力权重进行处理，如式 (2-19) 所示。多头注意力层通过将 Query、Key、Value 进行多次拆分并行计算，并保持总体参数不变，最终合并注意力权重。该方法采用的多个独立的头部可以对不同部分的输入信息进行关注，更易于获得全面、丰富的特征信息，如式 (2-20) 所示。

$$\text{Attention}(Q, K, V) = \text{soft max} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2-19)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^O \quad (2-20)$$

其中， $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ ， $Q \in \mathbb{R}^{n \times d_k}$ ， $K \in \mathbb{R}^{m \times d_k}$ ， $V \in \mathbb{R}^{m \times d_v}$ ， $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ， $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ， $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ ， $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ ， d_{model} 表示模型维度。

前馈连接层包含 RELU 激活函数以及两个线性变换，则当输入为 x 时，其计算公式为：

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2-21)$$

在 Transformer 中残差连接的作用与卷积神经网络相同，重点是要解决网络层层数过多导致的梯度消失问题。而归一化层的作用旨在保持不同信息在标准化过程中的独立性。

2.5 本章小结

本章主要介绍了视觉 SLAM 相关理论与技术。首先，通过对视觉传感器原理、SURF 算法与相机位姿估计进行解析，详细描述了视觉 SLAM 通过相机获取外部世界特征点信息并进行位姿估计的过程。其次，从基于优化的视觉 SLAM 算法对动态 SLAM 问题进行描述，分析动态物体对 SLAM 系统产生的影响，最后介绍了 Transformer 结构，为本文后续的动态视觉 SLAM 算法研究奠定了理论基础。

第3章 基于改进关键帧选取的鲁棒视觉 SLAM 算法

本章针对现有视觉 SLAM 算法由于光照变化、图像模糊、尺度变化易导致特征提取与匹配困难以及大曲率弯道运动拐点附近易发生漏选关键帧等问题，提出一种基于改进关键帧选取的鲁棒视觉 SLAM 算法（IR-VSLAM）。该算法在特征提取与匹配阶段，利用局部二维熵进行特征点检测提升特征点的独特性，同时引入曼哈顿距离构造距离评价函数进行预处理，最后采用粒子群算法优化匹配过程；在关键帧选取与闭环检测阶段，为避免大曲率弯道运动漏选关键帧，通过创建局部逆向索引窗口并依据窗口内关键帧、当前帧、参考帧三者中的特征点数量、匹配点数量、区域变化空间点数量计算自适应阈值，增补在曲线运动中遗漏的有效关键帧，提升算法定位精度。最后，在公开的数据集验证本章算法的有效性。IR-VSLAM 算法整体框架如图 3-1 所示。

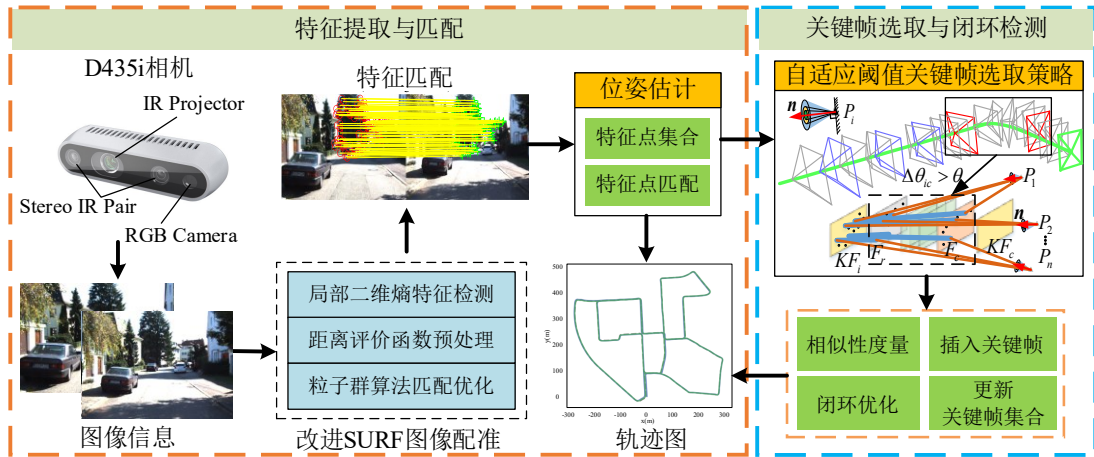


图 3-1 IR-VSLAM 算法整体框架

3.1 改进 SURF 图像配准算法

3.1.1 基于局部二维熵的特征检测

图像熵^[46]是一种衡量图像所含平均信息量大小的指标，其数值越大代表图像平均信息量越大，所包含的信息也越丰富^[47]。一维熵能反映图像分布的灰度信息，在一维熵的基础上引入二维熵^[48]，能反映特征点的空间信息，使检测的特征点更具独特性。具体做法是：计算图像邻域的灰度均值并定义为空间特征

向量，再将其与该图像的像素灰度值合成为特征二元组 (i, j) ，其中 i 、 j 分别是像素灰度值和邻域灰度均值， $0 \leq i, j \leq 255$ 。则局部二维熵定义为：

$$H = -\sum p_{ij} \log p_{ij} \quad (3-1)$$

其中， p_{ij} 能够反映某像素与它周围像素之间分布的相互关系，其定义为：

$$p_{ij} = f(i, j) / N^2 \quad (3-2)$$

其中， $f(i, j)$ 表示 (i, j) 出现的次数， N 表示尺度。

利用二维熵处理图像后，所提取的特征更加稳定准确。特征提取步骤为：

(1) 计算特征点 3×3 邻域的二维熵和所有特征点的局部邻域二维熵 M 。(2) 设置二维熵阈值 T_1 ， T_2 ($T_1 < T_2$)。当采样点的局部邻域二维熵 M 满足 $T_1 < M < T_2$ ，则确认该点是特征点，否则将该点剔除，这样就实现了对特征点的筛选功能。经过一些学者实验证明，发现当 $T_1=2$ ， $T_2=4.5$ 时实验的效果较好，因此本文将图像的二维熵设置在 $[2, 4.5]$ 之间^[49]。

3.1.2 基于距离评价函数的预处理

传统的 SURF 算法利用欧氏距离来实现特征匹配，由于其描述符有 64 维，若直接对特征点间的欧式距离进行遍历求解，必然会因为巨大的计算量降低匹配效率，因此本文引入计算量较小的曼哈顿距离来实现 SURF 特征匹配过程。

对于 n 维点 $X = (x_1, x_2, \dots, x_n)$ 和 $X' = (x'_1, x'_2, \dots, x'_n)$ 之间的欧式距离 D 与曼哈顿距离 D' 可定义为：

$$D = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2} \quad (3-3)$$

$$D' = \sum_{i=1}^n |x_i - x'_i| \quad (3-4)$$

由公式 (3-3) 和 (3-4) 可知，欧式距离计算时有平方和开方，而曼哈顿距离只需进行简单的加减以及绝对值运算，因此可减少计算量，提高算法速度。

接着，基于距离信息对提取的预匹配点对进行进一步的预处理，旨在剔除可能存在的错误匹配对和部分冗余信息，剩下的数据作为匹配的二次输入。具体剔除原理为：假设 (x_i, y_i) 和 (x'_i, y'_i) 是两帧图像上的正确匹配点对，则 x_i 和 x'_i

的距离 $d(x_i, x'_i)$ 应与 y_i 和 y'_i 的距离 $d(y_i, y'_i)$ 相似，根据此关系构建距离评价函数：

$$\begin{cases} \sigma_x = \frac{\sum_{i=1}^{n'-1} (|x_i + x'_i| / |y_i + y'_i|)}{n' - 1} \\ \sigma_y = \frac{\sum_{i=1}^{m'-1} (|y_i + y'_i| / |x_i + x'_i|)}{m' - 1} \end{cases} \quad (3-5)$$

其中， σ_x 为以横坐标为分子，纵坐标为分母的评价函数； $n'-1$ 为去除最大的 $|x_i + x'_i| / |y_i + y'_i|$ 后的比值； σ_y 为横坐标为分母，纵坐标为分子的评价函数； $m'-1$ 为去除最大的 $|y_i + y'_i| / |x_i + x'_i|$ 后的比值； $m' = n'$ 为匹配点对个数。若匹配点对满足： $|x_i + x'_i| / |y_i + y'_i| < \sigma_x$ ， $|y_i + y'_i| / |x_i + x'_i| < \sigma_y$ 则该点对保留，反之剔除。

3.1.3 基于粒子群的特征匹配优化

在图像匹配过程，可以通过粒子群算法（Particle Swarm Optimization, PSO）^[50] 优化极值搜索的过程，减少时间消耗。PSO 是一种随机搜索寻优算法，其核心在于利用粒子个体历史最优解 $qbest$ 和群体历史最优解 $pbest$ 的信息不断更新粒子位置和速度从而实现寻找最优值的目标。

粒子群算法的粒子速度和位置更新公式分别如公式（3-6）和（3-7）所示。

$$v_{id}^{n+1} = wv_{id}^n + c_1r_1(qbest_{id}^n - x_{id}^n) + c_2r_2(pbest_{id}^n - x_{id}^n) \quad (3-6)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1} \quad (3-7)$$

其中， n 为迭代次数， x_{id}^{n+1} 为当前时刻粒子在解空间中的位置向量， v_{id}^{n+1} 为速度分量， w 为惯性因子，可以调整局部和全局寻优性能， c_1 、 c_2 表示加速度常数， r_1 、 r_2 为区间 $[0,1]$ 的随机函数，用来增加搜索的随机性。

在算法的寻优过程中，解空间用等待匹配图像帧的所有特征向量表示，粒子数设置为 n ，随机分布在解空间中。为解决寻优过程中特征向量的离散化问题，将最优解 X 最小邻域内的离散解 X^* 视作最终成功匹配的特征向量。离散解 X^* 计算方法为：

$$X^* = \arg_{X'} \min [d(X, X')] \quad (3-8)$$

其中， X' 为解空间。第一次对特征向量方向相似性进行筛选的适应度函数为：

$$f(x) = \frac{(x, x')}{\|x\| \|x'\|} = \frac{\sum_{i=1}^n x_i x'_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n (x'_i)^2}} \quad (3-9)$$

其中, x , x' 分别为原始图像和待匹配图像点特征的特征向量。在算法初始化时计算粒子适应值, 将粒子初始位置设为个体最佳位置 $qbest_i$, 将适应值最优的个体位置设为最优种群位置 $pbest$ 。适应函数的值越大则特征向量的方向越相似。若 $f(x_i) > f(qbest_i)$, 则更新 $qbest_i$; 若 $f(x_i) > f(pbest)$, 则更新 $pbest = x_i$ 。在方向相似度配比完成后, 进行曼哈顿距离的寻找, 适应函数为:

$$f(x) = \sum_{i=1}^n |x_i - x'_i| \quad (3-10)$$

将粒子的初始位置定义为 $qbest_i$, 将适应值最好的个体位置定义为 $pbest$; 若 $f(x_i) < f(qbest_i)$, 则更新 $qbest_i$; 若 $f(x_i) < f(pbest)$, 则更新 $pbest = x_i$ 。若两次匹配的结果均为同一特征点, 则匹配成功, 进一步减小了误匹配的概率。基于改进 SURF 特征的图像配准方法流程图如图 3-2 所示。

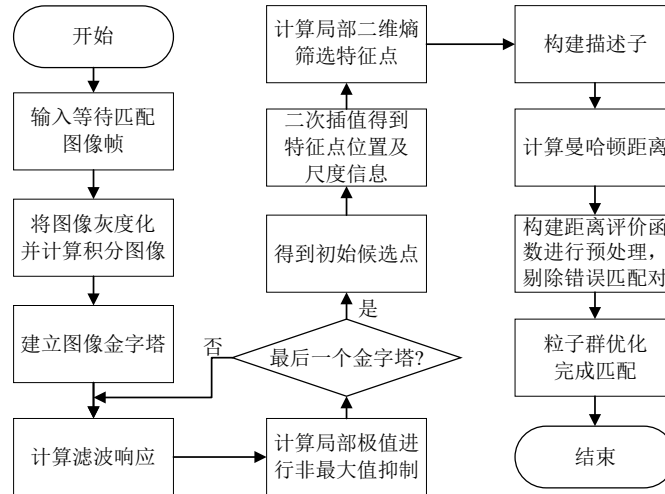


图 3-2 基于改进 SURF 特征的图像配准方法流程图

3.2 改进关键帧选取策略

3.2.1 传统关键帧选取策略

传统关键帧选取方法较多采用基于位置变化量和空间变化量的选取方法, 如图 3-3 所示。该方法将 SLAM 系统初始化后的第一帧图像设置为关键帧, 若当前帧与上一帧之间的旋转矩阵 R 大于阈值 θ , 说明两帧间旋转角度较大, 此

时选取当前帧为关键帧；否则，比较当前帧与上一关键帧之间的平移矩阵 $\mathbf{t}$ 是否大于平移阈值 E ，若 $\mathbf{t} > E$ 则说明两帧间平移距离较大，此时选取当前帧为关键帧。旋转矩阵 $\mathbf{R}$ 和平移矩阵 $\mathbf{t}$ 计算公式如下：

$$\begin{cases} p_j = \mathbf{R}p'_j + \mathbf{t}, \forall_j \\ e_j = p_j - (\mathbf{R}p'_j + \mathbf{t}) \\ \min_{\mathbf{R}, \mathbf{t}} \frac{1}{2} \sum_{j=1}^n \|p_j - (\mathbf{R}p'_j + \mathbf{t})\|_2^2 \end{cases} \quad (3-11)$$

其中， p_j 和 p'_j 分别为第一帧和第二帧图像中的第 j 点， e_j 为第 j 对点的误差项， $\mathbf{R}$ 和 $\mathbf{t}$ 分别为两帧间的旋转和平移矩阵。为求解 $\mathbf{R}$ 、 $\mathbf{t}$ ，使得 p_j 和 p'_j 一一对应，定义第 j 对点的误差项为 e_j ，运用 SVD 求解式中的 ICP 问题，构建最小二乘方程，求出使误差平方和达到最小的 $\mathbf{R}$ 和 $\mathbf{t}$ 。

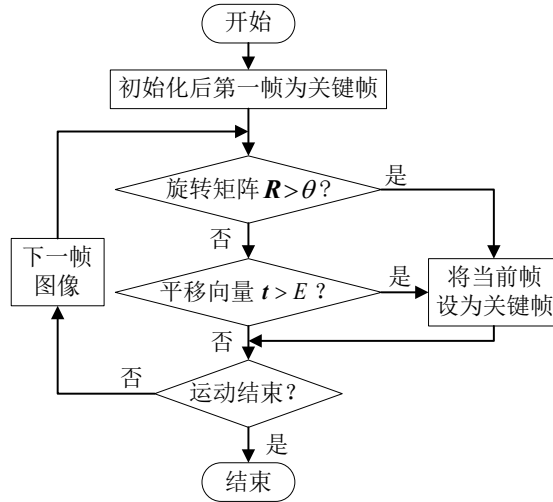


图 3-3 传统关键帧选取策略流程图

基于空间点位置变化的关键帧选取策略在实现时较为简便且运行稳定，在较大程度上克服了图像的冗余存储和计算量过大等问题。然而当移动机器人进行大曲率弯道运动时该方法易发生漏选关键位置处的关键帧，导致位姿估计失准、影响系统闭环准确率和轨迹全局一致性。

3.2.2 自适应阈值关键帧选取策略

关键帧选取是 SLAM 系统的重要环节之一，为 SLAM 系统的后端优化和闭环提供了重要的原始数据信息，因此关键帧选取的正确与否将直接影响整个 SLAM 系统的精度。为解决现有关键帧选取算法在面临大曲率弯道运动时由于

视角差急剧变化导致未能准确选取关键帧的问题，提出一种基于自适应阈值的关键帧选取策略。当移动机器人进行旋转变换量 $\Delta\theta$ 变化较小的直线运动时，运动变化主要由位移变化量 Δp 体现，因此直线运动时根据位移变化量 Δp 与阈值 τ 的大小关系进行关键帧的选取，当 $\Delta p > \tau$ 时，选取当前帧为关键帧。当机器人进行旋转变换量 $\Delta\theta$ 变化较大的曲线运动时，运动主要由旋转变换量 $\Delta\theta$ 体现，因此曲线运动时根据旋转变换量 $\Delta\theta$ 与阈值 γ 的大小关系进行关键帧的选取，当 $\Delta\theta > \gamma$ 时，选取当前帧为关键帧。当机器人进行旋转变换量 $\Delta\theta$ 变化更大的大曲率弯道运动时，由于在短距离和时间内机器人的旋转变换量 $\Delta\theta$ 较大引起视差角急剧变化易造成算法漏选关键帧，影响系统闭环检测精度和定位精度。因此本文在机器人进行大曲率弯道运动时，结合大曲率角度阈值 θ ($\theta > \gamma$) 与自适应阈值选取关键帧。计算机器人运动时获取的当前图像帧与关键帧 KF_i 之间的旋转变换量 $\Delta\theta_{ic}$ ，当 $\Delta\theta_{ic}$ 大于大曲率角度阈值 θ 时，则认为此时机器人正在进行大曲率弯道运动，选取该帧为新的关键帧，记为 KF_c ，同时将关键帧 KF_i 与 KF_c 之间的普通帧定义为局部逆向索引窗口，使用自适应阈值对该窗口内漏选的关键帧进行逆向索引。自适应阈值关键帧选取策略示意图如图 3-4 所示。

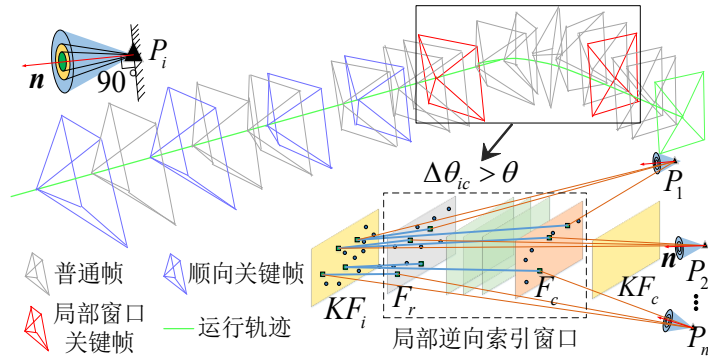


图 3-4 自适应阈值关键帧选取策略示意图

局部逆向索引窗口内与关键帧 KF_i 相似度最高的帧定义为参考帧 F_r ，将当前帧 F_c 中特征点与参考帧 F_r 中特征点（图 3-4 中圆点）进行匹配，匹配点为图中正方形点。根据 KF_i 、 F_c 、 F_r 中匹配关系计算形成空间点 P_1 、 P_2 、...、 P_n ，以空间点 P_i ($i=1,2,\dots,n$) 为顶点做出锥角为 9 度的圆锥体，其主轴所在向量 n 垂直于空间点 P_i 位置处的表面，方向指向相机方向，圆锥体底面由同心圆划分为三个区域，每个区域与顶点 P_i 形成一个新的圆锥，从最大圆锥到最小圆锥顶角依次为 9 度、6 度、3 度，相邻区域之间角度为 3 度。圆锥顶角选取 9 度是依据

大多数特征点检测算法能够在此角度内找到对应特征点^[51]。将空间点 P_i 分别与 KF_i 、 F_c 、 F_r 中各对应匹配点连接形成相应的空间点连接线，空间点连接线通过圆锥底面的不同区域则定义该空间点为区域变化空间点。利用 KF_i 、 F_c 、 F_r 中的特征点数量、匹配点数量、区域变化空间点数量等参数计算自适应阈值 A ，其计算公式为：

$$A = \lambda(R_c Q_r / R_r + T_c Q_r / T_r) \quad (3-12)$$

其中， λ 为自适应阈值的动态系数， R_c 为当前帧 F_c 与关键帧 KF_i 匹配点数量， R_r 为参考帧 F_r 与关键帧 KF_i 匹配点数量， T_c 为当前帧 F_c 特征点数量， T_r 为参考帧 F_r 特征点数量， Q_r 为关键帧 KF_i 、参考帧 F_r 区域变化空间点数量。

为了增强自适应阈值 A 对大曲率弯道运动场景的适应能力，设计随曲率变化的参数调节自适应阈值的动态系数 λ ，自适应阈值的动态系数计算公式为：

$$\lambda = (1 + \gamma - \beta - \alpha) \quad (3-13)$$

其中， γ 、 β 与 α 为动态系数 λ 的调节参数。

γ 通过降低在关键帧 KF_i 之后的前两个帧中选择关键帧的概率，并确保在从关键帧 KF_i 经过 4 个帧之后插入新的关键帧的方式使自适应阈值 A 更具有适应性，同时通过减少从参考帧 F_r 到当前帧 F_c 的匹配点数量使自适应阈值 A 的初始参数更严格， γ 计算公式为：

$$\gamma = (2 - N/2)(1 - R_c / R_r) \quad (3-14)$$

其中， N 为当前帧 F_c 与关键帧 KF_i 间图像帧数量。

虽然采用参数 γ 在理论上可保证关键帧的均匀分布，但是在实际过程中，区域变化空间点数量从未超过自适应阈值以至于未能选取关键帧，因此采用参数 β 简化自适应阈值 A 。而参数 α 用于通过增加区域变化空间点数量（大于匹配点数量的半数）来调节阈值， β 和 α 计算公式为：

$$\begin{cases} \beta = Q_c / R_c - Q_c / R_r \\ \alpha = Q_c / R_c - 0.5 \end{cases} \quad (3-15)$$

其中， Q_c 为关键帧 KF_i 、当前帧 F_c 区域变化空间点数量。

为避免区域变化空间点数量太多导致当前帧与上一关键帧差别较大，造成两帧之间难以对齐，因此设定区域变化空间点的阈值上限 ω 。然后根据区域变

化空间点数量与自适应阈值 A 的大小关系确定是否增补曲线运动遗漏的关键帧。在局部逆向索引窗口中若区域变化空间点数量大于自适应阈值 A 且小于区域变化空间点的阈值上限 ω 时，则定义当前帧 F_c 为大曲率弯道运动漏选关键帧。基于自适应阈值的关键帧选取数学表达式如式 (3-16) 所示。当新的关键帧是根据相机旋转变量选择且旋转变量大于阈值 θ 时，创建局部逆向索引窗口执行自适应阈值的关键帧选取策略。自适应关键帧选取策略如算法 1 所示。

$$KF_i = \left\{ F_j \left| \left([\Delta p_{ic} > \tau] \cup [\Delta \theta_{ic} > \gamma] \cup [\Delta \theta_{ic} > \theta] \cup [\omega > Q_c + Q_r > A] \right) \right. \right\} \quad (3-16)$$

算法 1 基于自适应阈值的关键帧选取策略

Input: 图像帧序列 F_j

Output: 关键帧 KF_i

Parameters: 位移阈值 τ , 角度阈值 γ 、大曲率角度阈值 θ , 自适应阈值 A , 区域变化空间点的阈值上限 ω , 动态系数 λ , 其他参数: R_c, R_r, T_c, Q_c, Q_r

$KF_1 \leftarrow F_1$

FOR new F_c THEN

 计算上一关键帧 KF_i 与 F_c 之间的旋转变量 $\Delta \theta_{ic}$ 和位移变化量 Δp_{ic}

 IF $\Delta p_{ic} > \tau$ THEN

 插入新的关键帧, $KF_i \leftarrow F_j, i = i + 1$

 ELSEIF $\Delta \theta_{ic} > \gamma$ THEN

 插入新的关键帧, $KF_i \leftarrow F_j, i = i + 1$

 IF $\Delta \theta_{ic} > \theta$ THEN

 创建局部逆向索引窗口

 计算自适应阈值 A 和动态系数 λ :

$$A = \lambda(R_c Q_r / R_r + T_c Q_r / T_r)$$

$$\lambda = (1 + \gamma - \beta - \alpha)$$

 IF $\omega > Q_c + Q_r > A$ THEN

 插入新的关键帧, $KF_i \leftarrow F_c, i = i + 1$

 ELSE

 RETURN NULL

 END IF

 END IF

END FOR

3.3 实验结果与分析

为了验证所提 IR-VSLAM 算法的有效性，在公开的 KITTI^[52]、EuRoC^[53]以及 TUM^[54]数据集进行改进 SURF 图像配准对比实验、自适应阈值关键帧选取对比实验以及定位精度对比实验。实验平台电脑配置为：Ubuntu18.04 操作系统，CPU 为 I5-11400F，GPU 为 RTX3060，内存 16G。

3.3.1 改进 SURF 图像配准对比实验

为了验证改进 SURF 算法对光照变化的适应性，将改进 SURF 算法与 SIFT、ORB 和 SURF 算法进行图像配准对比，4 种算法在光照变化场景下的匹配对比如图 3-5 所示。由图可知，改进的 SURF 算法在 4 种算法中图像匹配效果最优。改进的 SURF 算法对于两种不同光照强度图像的匹配效果相比于传统的 SURF 算法具有明显的改善，其中 ORB 的匹配点数最少，其中也存在一些误匹配点，同样 SIFT 以及 SURF 算法相对改进后的 SURF 算法而言也存在许多误匹配点。

4 种算法在光照变化场景下的匹配结果对比如表 3-1 所示。由表中数据可知，SIFT 算法速度最慢，ORB 算法速度最快。改进的 SURF 算法运行速度是传统 SURF 算法的 3 倍。由于引入了二维熵提高了所检测特征点的独特性，所以改进的 SURF 算法提取的特征点个数少于传统 SURF 算法，但检测精度与之相比提高了 9%。与 ORB 算法相比，虽然改进的 SURF 算法计算时间有所增加，但精度远高于 ORB 算法，且满足实时性需求。

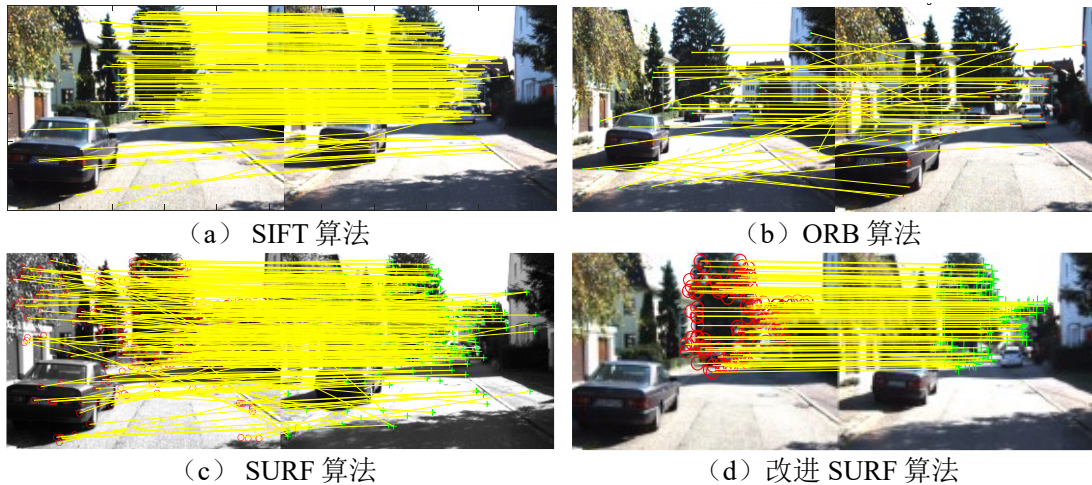


图 3-5 4 种算法在光照变化场景下的匹配对比图

表 3-1 4 种算法在光照变化场景下的匹配结果对比

算法	SIFT	ORB	SURF	改进后的 SURF
时间 (s)	30.41	0.25	1.09	0.36
左图特征点数 (个)	2139	804	718	643
右图特征点数 (个)	1918	699	376	304
匹配点数 (对)	273	48	179	110
正确率 (%)	80.1	74.2	82.3	89.7

为了验证改进的 SURF 算法对模糊图像的处理能力，将清晰度不同的两张图像分用 SIFT 算法、ORB 算法、SURF 算法以及改进后的 SURF 进行处理，4

种算法在图像模糊场景下的匹配对比如图 3-6 所示。由图可知，SIFT、ORB 以及 SURF 算法在图像模糊场景下存在较为明显的错误匹配对，匹配效果较差。而改进后的 SURF 算法由于采用局部二维熵筛选高质量特征点，并采用距离评价函数剔除了错误匹配对，因此算法整体匹配效果较好，未出现明显的错误匹配。

4 种算法在图像模糊场景下的匹配结果对比如表 3-2 所示。由表中数据可知，SURF、SIFT 以及 ORB 算法由于图像模糊影响在待匹配图像帧中所提取的特征点数量均较少，且存在较多的错误匹配，匹配准确率较低。改进的 SURF 算法有效剔除了错误特征点，与传统 SURF 算法相比准确率提高了 5.7%，且在四种算法中正确率最高。在运行速度方面，改进后的 SURF 算法与传统的 SURF 算法相比速度提高了 5 倍，仅用 0.16s 即可完成特征提取与匹配的过程。

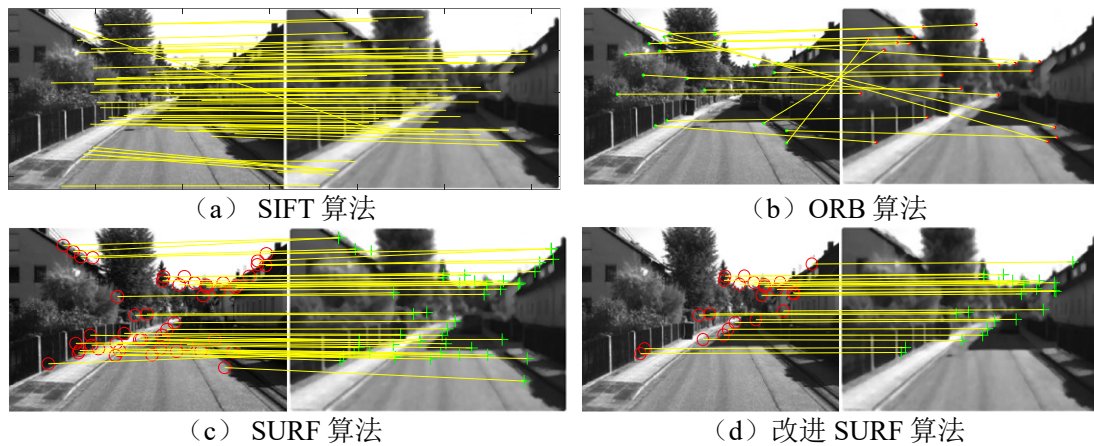


图 3-6 4 种算法在图像模糊场景下的匹配对比图

表 3-2 4 种算法在图像模糊场景下的匹配结果对比

算法	SIFT	ORB	SURF	改进后的 SURF
时间 (s)	3.41	0.10	0.88	0.16
左图特征点数 (个)	468	238	174	159
右图特征点数 (个)	270	92	141	122
匹配点数 (对)	74	23	49	32
正确率 (%)	80.1	74.2	82.3	89.7

在图像特征提取与匹配的过程中，由于视觉传感器与目标物体的距离有所差异，因此相机所获取的图像往往尺寸不同，为了验证改进的 SURF 算法对尺度变化图像的应对能力，分别用 SIFT、ORB、SURF 以及改进的 SURF 算法对尺寸不同的两张图像进行特征提取与匹配对比，4 种算法在尺度变化场景下的匹配对比如图 3-7 所示。由图可知，SIFT、ORB 以及 SURF 算法由于图像尺度变化导致发生了错误的匹配。而改进的 SURF 算法未发生错误匹配，表现出对尺度变化场景较强的鲁棒性。

4 种算法在尺度变化场景下的匹配结果对比如表 3-3 所示。由表中数据可知，由于右图尺寸较大，所以右图提取的特征点数多于左图。从速度方面分析，4 种算法中 SIFT 算法速度最慢，ORB 算法速度最快，是 SIFT 算法的 105 倍，而改进的 SURF 算法相比于传统的 SURF 算法运行速度提高了 2.5 倍。在精度方面，改进后的 SURF 算法正确率达到了 88.4%，相较于传统 SURF 算法提高了 4%，也高于 SIFT 算法和 ORB 算法。

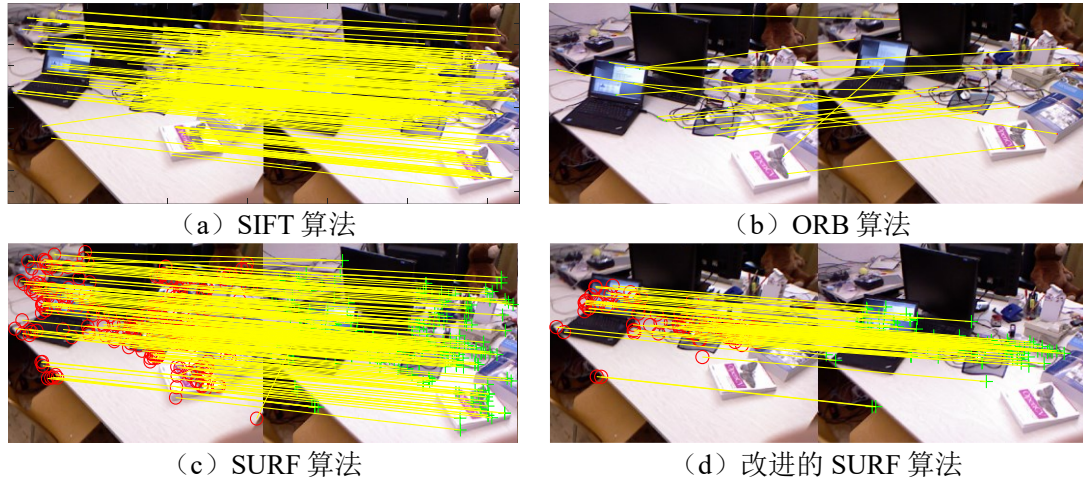


图 3-7 4 种算法在尺度变化场景下的匹配对比图

表 3-3 4 种算法在尺度变化场景下的匹配结果对比

算法	SIFT	ORB	SURF	改进后的 SURF
时间 (s)	59.73	0.57	1.87	0.75
左图特征点数 (个)	1563	554	725	659
右图特征点数 (个)	1777	653	839	733
匹配点数 (对)	308	23	246	90
正确率 (%)	82.3	77.0	84.8	88.4

3.3.2 自适应阈值关键帧选取对比实验

为了验证所提自适应阈值关键帧选取策略的有效性，将自适应阈值关键帧选取策略与文献[30]所提采用固定阈值的关键帧策略进行关键帧选取对比实验，实验数据选用 KITTI 和 EuRoC 数据集中具有丰富曲率变化的序列。使用固定阈值 (FT-SLAM) 和使用自适应阈值 (AT-SLAM) 的关键帧选取算法在 KITTI 4 个序列下获取的轨迹对比如图 3-8 所示。从图 3-8 中曲率变化明显的位置 1 和位置 2 局部放大图可看出，FT-SLAM 采用固定阈值的关键帧选取策略难以适应曲率变化，而 AT-SLAM 充分考虑了帧间特征点的关系，在局部逆向索引窗口内依据关键帧、当前帧、参考帧三者中的特征点数量、匹配点数量、区域变化空间

点数量设定的自适应阈值相比于固定阈值更能适应曲率变化，从而增补大曲率弯道运动时遗漏的有效关键帧，因此本文算法能更好的跟踪曲线运动的轨迹并提升整体的建图质量。

FT-SLAM 和 AT-SLAM 算法在 KITTI 数据集 4 个序列中运行获取的对比数据如表 3-4 所示。由表可知，AT-SLAM 的平均绝对轨迹误差相比于 FT-SLAM 减小了 26%。这是由于 AT-SLAM 采用基于自适应阈值的关键帧选取策略增补了在大曲率弯道运动中遗漏的关键帧，提升了算法的闭环准确率和定位精度。

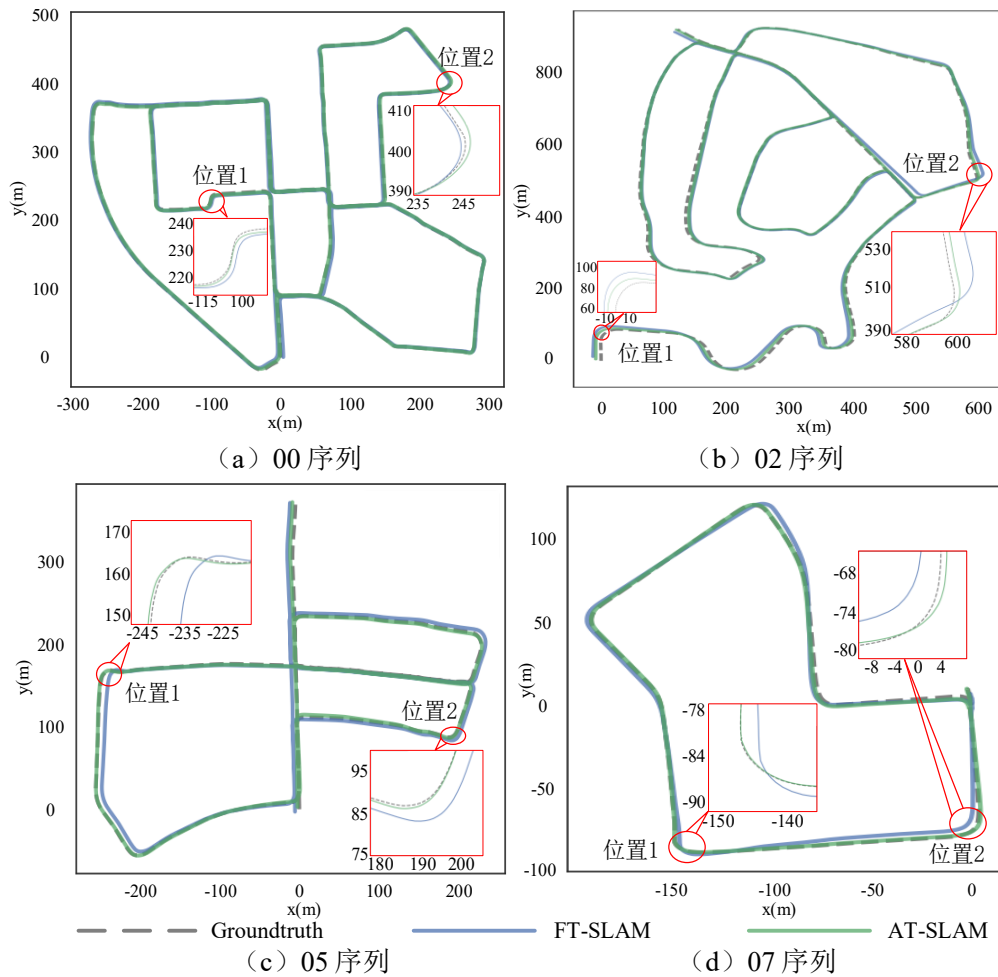


图 3-8 KITTI 数据集轨迹对比

表 3-4 2 种算法在 KITTI 数据集 4 个序列下的结果对比

KITTI 序列	位置 1 提取关键帧数 (个)		位置 1 轨迹均方根误差 RMSE (m)		位置 2 提取关键帧数 (个)		位置 2 轨迹均方根误差 RMSE (m)		全局 提取关键帧数 (个)		全局 轨迹均方根误差 RMSE (m)	
	FT- SLAM	AT- SLAM	FT- SLAM	AT- SLAM	FT- SLAM	AT- SLAM	FT- SLAM	AT- SLAM	FT- SLAM	AT- SLAM	FT- SLAM	AT- SLAM
00	80	121	7.62	5.99	32	54	9.84	9.91	2136	2363	8.97	7.44
02	75	86	1.72	0.34	102	125	29.11	10.10	2285	2476	17.86	12.02
05	55	67	1.93	1.32	34	52	7.34	2.65	1123	1236	4.76	4.47
07	135	147	5.31	0.18	56	88	3.32	0.06	541	679	2.66	1.01
平均值	86	105	4.15	1.96	56	79	12.4	5.68	1521	1688	8.56	6.24

使用固定阈值（FT-SLAM）和使用自适应阈值（AT-SLAM）的关键帧选取算法在 EuRoC 数据集 4 个序列下轨迹对比如图 3-9 所示。从图 3-9（a）局部放大图可以看出，AT-SLAM 算法和 FT-SLAM 算法均能较好的完成弯道处的跟踪与地图构建，但 AT-SLAM 算法由于采用自适应阈值关键帧选取策略所构建地图的轨迹误差相比于 FT-SLAM 算法更小。从图 3-9（b）、3-9（c）和 3-9（d）中可以看出，在曲率变化较大的 MH03、MH05 和 V203 序列，由于移动机器人在局部放大区域和红色框内曲率变换明显且剧烈，造成 FT-SLAM 算法分别在 $(4.1\text{m}, -0.8\text{m})$ 、 $(-2.3\text{m}, 8.4\text{m})$ 和 $(-2\text{m}, 0\text{m})$ 处产生轨迹跟踪丢失，而本文所提 AT-SLAM 算法采用基于自适应阈值的关键帧选取策略对曲率变化具有更好的适应能力，所建立的轨迹更加接近真实轨迹。

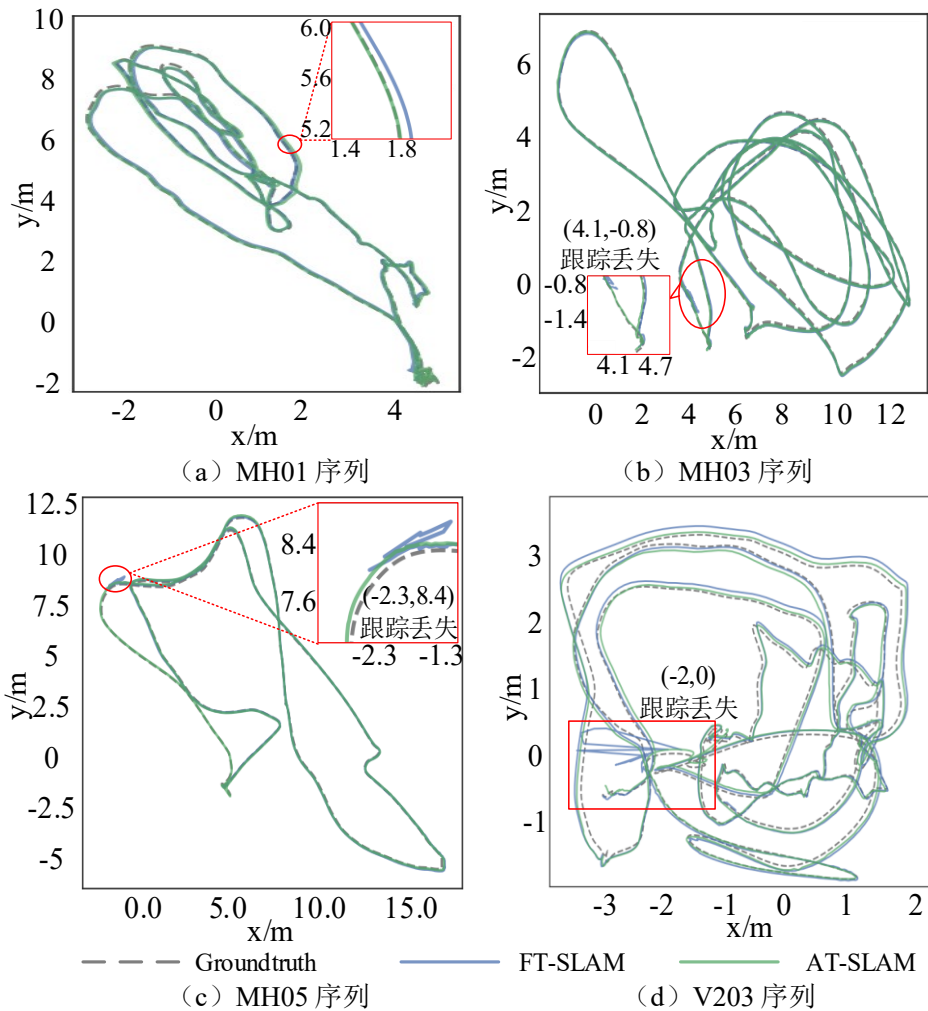


图 3-9 EuRoC 数据集轨迹对比

FT-SLAM 和 AT-SLAM 算法在 EuRoC 数据集 4 个序列结果对比如表 3-5 所示。由表可知，由于 AT-SLAM 算法采用的自适应阈值关键帧选取策略增补了在

大曲率运动中遗漏的有效关键帧，因此 AT-SLAM 与 FT-SLAM 算法相比绝对轨迹误差减小了 22%。因此在大曲率运动下自适应阈值关键帧选取策略在定位精度方面与固定阈值关键帧选取策略相比有较大优势。

表 3-5 2 种算法在 EuRoC 数据集 4 个序列下的结果对比

EuRoC 序列	提取关键帧数（帧）		轨迹均方根误差 RMSE（m）	
	FT-SLAM	AT-SLAM	FT-SLAM	AT-SLAM
MH01	2753	3082	0.036	0.032
MH03	2237	2400	0.052	0.034
MH05	1571	1873	0.072	0.065
V203	1722	1921	0.038	0.023
平均值	2057	2319	0.050	0.039

ORB-SLAM3、FT-SLAM 和 AT-SLAM 三种算法在 KITTI-05 和 EuRoC-V203 序列下的闭环准确率-召回率对比图如图 3-10 所示。ORB-SLAM3 在 100%准确率下，召回率分别为 73.5%和 71.7%；FT-SLAM 在 100%准确率下，召回率分别为 82.3%和 80.5%；AT-SLAM 在 100%准确率下，召回率分别为 88.3%和 85.4%。

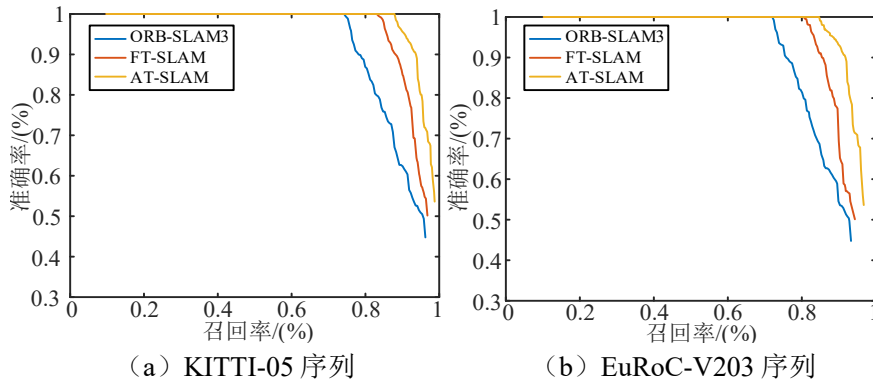


图 3-10 准确率-召回率对比

3.3.3 定位精度对比实验

为了进一步验证所提 IR-VSLAM 算法的有效性，将本文算法与 ORB-SLAM2 算法在公开 TUM 数据集进行对比实验，并进行定量和定性分析。IR-VSLAM 算法与 ORB-SLAM2 算法在 TUM 数据集所建立的轨迹对比如图 3-11 所示。IR-VSLAM 算法在特征提取与匹配阶段，利用局部二维熵进行特征点检测提升特征点的独特性，并引入距离评价函数进行预处理剔除错误匹配对，最后采用粒子群算法优化匹配过程，增强了算法在复杂场景下特征提取与匹配的准确性；在关键帧选取阶段，充分考虑帧间特征点的关系，在局部逆向索引窗口

内依据关键帧、当前帧、参考帧三者中的特征点数量、匹配点数量、区域变化空间点数量设定的自适应阈值选取关键帧，增补了在弯道运动时遗漏的关键帧，提升了定位精度，因此本章算法建立的地图轨迹误差更小。

本章算法与 ORB-SLAM2 在 TUM 4 个序列下的 ATE 和 RPE 结果对比如表 3-6 所示，从表中数据可知，与 ORB-SLAM2 相比，本章算法绝对轨迹误差降低 22.22%~36.52%，平移相对误差降低 17.49%~21.9%，具有较大优势。

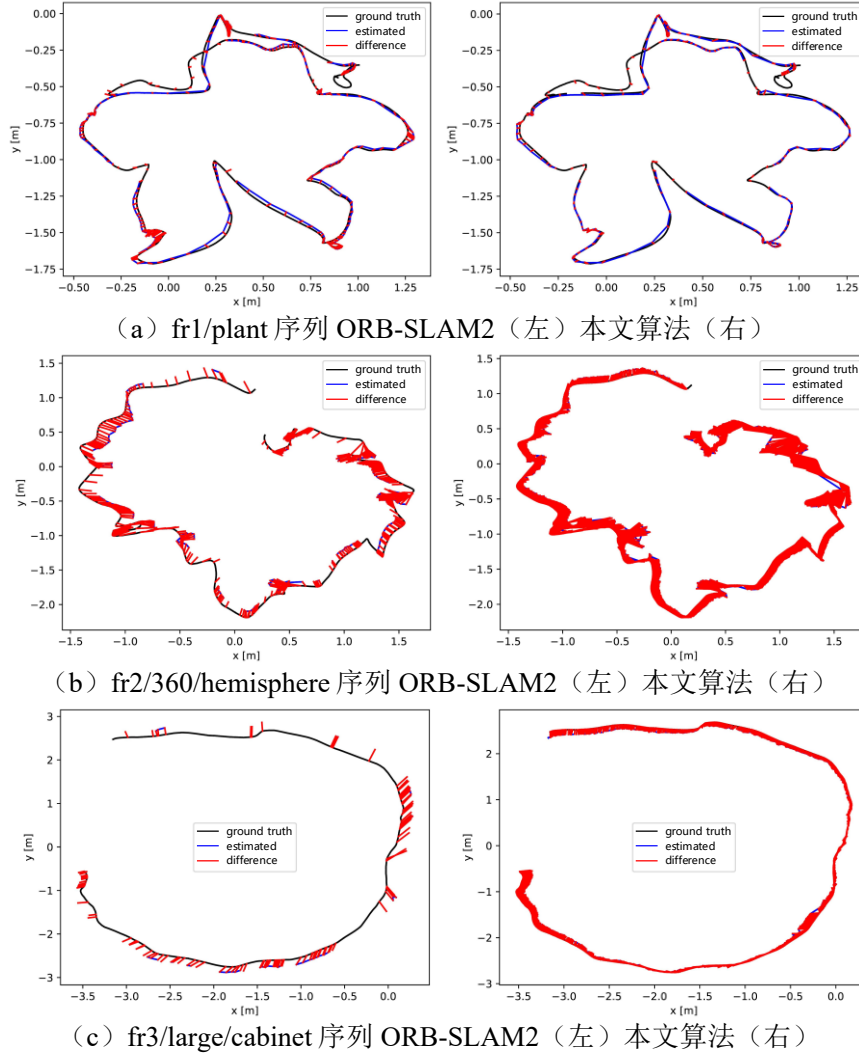


图 3-11 TUM 数据集轨迹对比

表 3-6 2 种算法在 TUM 数据集 4 个序列下的结果对比

TUM 序列	ORB-SLAM2		本章算法		提升幅度	
	ATE (m)	RPE (°)	ATE (m)	RPE (°)	ATE (%)	RPE (%)
fr1/plant	0.014	1.588	0.010	1.240	28.57	21.91
fr2/360/hemisphere	0.115	1.664	0.073	1.213	36.52	27.10
fr2/desk	0.009	0.772	0.007	0.637	22.22	17.49
fr3/large/cabinet	0.059	1.026	0.038	0.804	35.60	21.63

3.4 本章小结

本章针对现有视觉 SLAM 算法在光照变化、图像模糊、尺度变化场景易发生特征提取与匹配困难以及在大曲率弯道运动下易发生漏选关键帧的问题，提出了一种 IR-VSLAM 算法，该算法在特征提取与匹配环节，引入局部二维熵、距离评价函数以及粒子群算法增强图像配准的准确性和实时性。在关键帧选取阶段，采用基于自适应阈值的关键帧选取策略增补大曲率弯道运动下遗漏的有效关键帧，提升系统在大曲率弯道场景下的建图能力。通过在公开的数据集进行定量和定性分析，结果表明本章算法与 ORB-SLAM2 算法相比建立的轨迹地图误差更小，系统定位精度更高。

然而，本章算法所提自适应阈值关键帧选取策略依据局部逆向索引窗口内关键帧、当前帧、参考帧三者中的特征点数量、匹配点数量、区域变化空间点数量等参数计算自适应阈值，当场景中出现动态物体时，由于动态点的错误匹配将造成匹配点数量等参数的准确性下降，从而导致自适应阈值计算不准确，最终影响关键帧选取的准确性。因此，在第 4 章，将融合实例分割网络和物体运动判断方法增强 SLAM 系统在动态场景下的鲁棒性。

第4章 动态场景下基于改进 Transformer 的视觉 SLAM 算法

当移动机器人运行于动态场景时，动态点的误匹配会极大干扰相机位姿求解，造成地图可用性急剧下降。目前，采用深度学习网络分割动态物体并剔除动态点，最终采用静态点进行地图构建已成为动态 SLAM 的有效解决方案之一。本章针对现有动态 SLAM 算法对物体分割不完整、难以准确判断潜在动态物体的运动状态等问题，提出一种动态场景下基于改进 Transformer 的视觉 SLAM 算法。为减少动态物体对 SLAM 系统定位精度的干扰，通过构建残差坐标注意力模块，引导网络关注漏分割目标，同时融合外部相对位置偏置优化漏分割目标边界语义信息，对潜在动态物体进行准确标记；与此同时，提出一种基于双重约束的物体运动判断方法准确估计潜在动态物体的运动状态。最后在公开的 TUM 动态数据集和真实场景验证本文算法的有效性。

4.1 算法整体框架

本文算法整体框架如图 4-1 所示。在潜在动态物体分割阶段，通过所提 CRNET 网络分割输入图像中的潜在动态物体并生成语义信息，同时提取图像特征点。在潜在动态物体运动判断与建图阶段，采用基于双重约束的物体运动判断策略准确区分潜在动态物体的运动状态并剔除动态点，最终提升系统的定位精度。

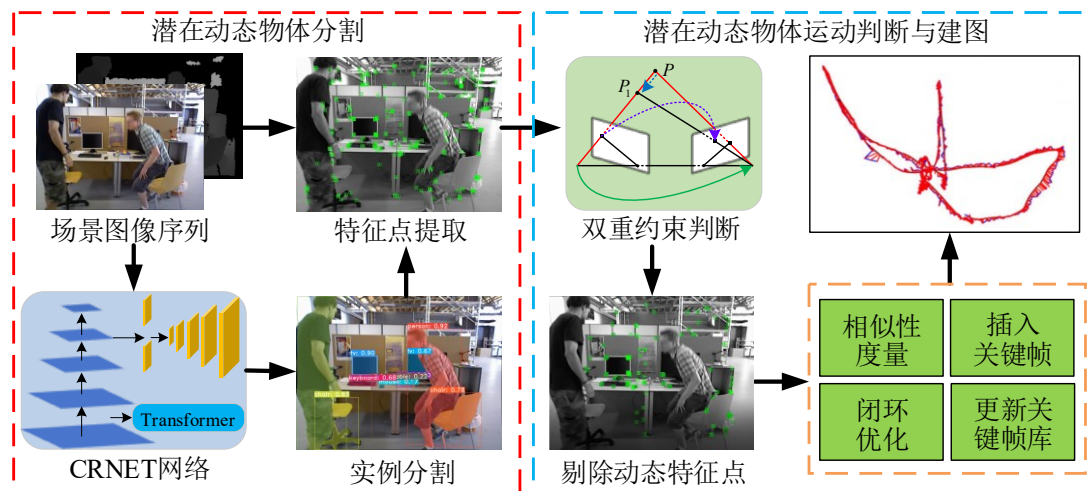


图 4-1 算法整体框架

4.2 潜在动态物体分割

为解决传统物体分割算法在进行动态物体分割时由于物体间相互遮挡导致物体在相机视野中产生不完整外观造成分割效果较差甚至漏分割等问题，提出一种融合坐标注意力（Coordinate Attention, CA）机制^[55]与相对位置偏置（Relative position bias）^[56]的实例分割网络 CRNET。该网络主要由主干网络模块、残差坐标注意力模块以及融合外部相对位置偏置 Transformer 模块共同组成。主干网络模块的基础为 ResNet-101，并在其基础上添加 FPN（Feature pyramid networks for object detection）^[57]构成 ResNet-FPN 网络。与此同时，残差坐标注意力模块接收来自主干网络携带高维信息的 P4、P5 特征图，而融合外部相对位置偏置 Transformer 模块接收边界信息丰富的 P3 特征图，此方法可以更好的融合来自主干网络的不同类型特征。为了加强外观缺失物体区域的像素权重进而提升对外观缺失物体的识别率，在残差坐标注意力模块使用对通道和位置信息均敏感的坐标注意力机制。与此同时，为了加强遮挡物体与外观缺失物体之间的边界语义信息，提升对外观缺失物体的边界分割准确性，将外部相对位置偏置融入 Transformer 从而构建融合外部相对位置偏置的 Transformer 模块。CRNET 网络结构如图 4-2 所示。

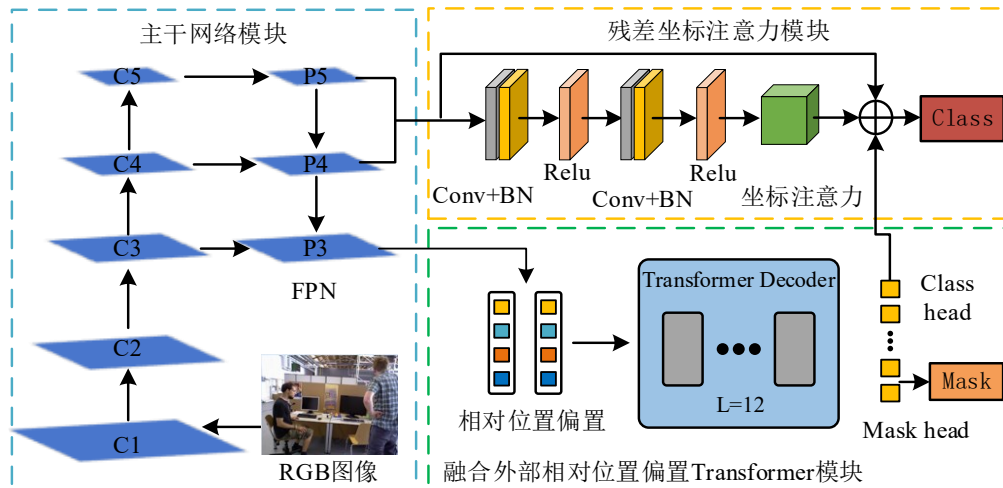


图 4-2 CRNET 网络结构图

4.2.1 残差坐标注意力模块

视觉 SLAM 系统运行过程中，由于相机视野中外观缺失的物体占据图像区域较大，如果不能对其中的潜在动态物体进行准确识别，将严重影响 SLAM 系

统的鲁棒性。本文通过在残差模块中加入坐标注意力机制，从而构造出对外观缺失物体更具特征表达能力的残差坐标注意力模块。将特征图 M' 经过两次卷积、BN 层、Relu 激活操作，最后与输入特征 M' 跳跃链接实现残差学习得到对通道和位置均敏感的特征图 M'' 。残差坐标注意力模块如图 4-3 所示。

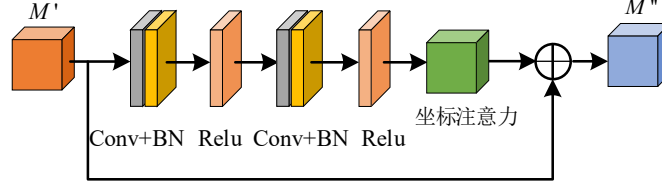


图 4-3 残差坐标注意力模块

残差坐标注意力模块不但增加了感兴趣区域的特征通道权重，还能获得该区域空间位置，充分突出外观缺失区域有效特征信息，同时又避免了干扰信息影响。具体计算方式为：

$$M'' = M' + C(\sigma(Bf_{3 \times 3}(\sigma(Bf_{1 \times 1}(M'))))) \quad (4-1)$$

其中， M' 为输入特征， M'' 为残差坐标注意力模块的输出特征， $C(\cdot)$ 为使用坐标注意力提取特征， $f_{1 \times 1}$ 和 $f_{3 \times 3}$ 分别为 1×1 和 3×3 的卷积层， $\sigma(\cdot)$ 为 Relu 激活函数， B 为归一化参数。

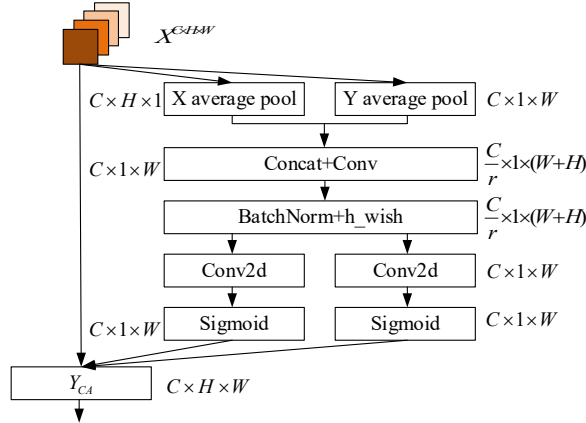


图 4-4 坐标注意力模块

本文坐标注意力模块如图 4-4 所示。现有的 SE (squeeze-and-excitation) 注意力机制^[58]对内部通道信息进行关注，但忽视了空间位置信息。而卷积注意力模块 (convolutional block attention module, CBAM)^[59]为引入空间位置信息，在通道上使用全局池化的方法，但此方法却又只考虑了局部范围信息。本文通过在通道注意力中融入空间位置信息，生成对位置信息和通道信息均敏感的特征图从而使网络更加关注高维特征信息，提高网络对外观缺失区域的注意力。CA 注意力机制首先对输入的特征图 $X^{C \times H \times W}$ 使用尺寸 $(H, 1)$ 和 $(1, W)$ 的池化核，分别

对水平和垂直方向的每个通道进行池化，得到一对大小为 $C \times H \times 1$ 和 $C \times 1 \times W$ 的特征图。高度为 h 的第 c 个通道和宽度为 w 的第 c 个通道的输出如下：

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c(h, i) \quad (4-2)$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j \leq H} x_c(j, w) \quad (4-3)$$

其中， $x_c(h, i)$ 和 $x_c(j, w)$ 分别为水平和竖直方向的特征， $z_c^h(h)$ 和 $z_c^w(w)$ 分别是编码后的水平张量和竖直张量， H ， W 和 C 分别为输入特征图的长、宽和通道数。接着对这两个张量进行融合操作并且用 1×1 卷积函数实现降维，如下式所示：

$$F = \delta(f_{1 \times 1}[z^h, z^w]), F \in \mathbb{R}^{C/r \times (H+W)} \quad (4-4)$$

其中， $[z^h, z^w]$ 为融合操作， $f_{1 \times 1}$ 为 1×1 卷积函数， $F \in \mathbb{R}^{C/r \times (H+W)}$ 为空间信息在水平和竖直方向上进行编码的中间特征映射， $\delta(\bullet)$ 为 h_wish 激活函数， r 是下采样步长。

接着将 F 沿着空间维度重新分解为两个单独的张量 F^h 和 F^w ，然后分别使用 1×1 卷积函数将 F^h 和 F^w 通道数变为与 X 一致，得到如下结果：

$$g^h = \sigma(f_{1 \times 1}(F^h)), g^h \in \mathbb{R}^{C/r \times H} \quad (4-5)$$

$$g^w = \sigma(f_{1 \times 1}(F^w)), g^w \in \mathbb{R}^{C/r \times W} \quad (4-6)$$

其中， $\sigma(\bullet)$ 为 *sigmoid* 激活函数， g^h 和 g^w 分别为水平和竖直方向的权重。最终 CA 注意力模块的输出表示为：

$$Y_{CA}(i, j) = X_c(i, j) \times g^h(i) \times g^w(j) \quad (4-7)$$

表 4-1 主干网络与残差坐标注意力模块主要参数设置

序号	层类型	区域划分	输出
1	ConvC1	Resnet-101,C1	(64,H/4,W/4)
2	ConvC2	Resnet-101,C2	(256,H/4,W/4)
3	ConvC3	Resnet-101,C3	(512,H/8,W/8)
4	ConvC4	Resnet-101,C4	(1024,H/16,W/16)
5	ConvC5	Resnet-101,C5	(2048,H/32,W/32)
6	ConvP3	FPN,P3	(512,H/8,W/8)
7	ConvP4	FPN,P4	(1024,H/16,W/16)
8	ConvP5	FPN,P5	(2048,H/32,W/32)
9	FC	Coordinate attention	(C,1,1)
10	Activation Function	Coordinate attention	Sigmoid

4.2.2 融合外部相对位置偏置 Transformer 模块

Transformer 在广泛的视觉任务中表现优异，在计算机视觉领域受到极大关注。传统 Transformer 位置编码应用于 soft max 函数内部以点积方式重新评估注意力得分。然而，由于相似向量的点积较大导致注意力权重值较大，这会使得注意力集中在相似区域而不是真正相关的区域，从而产生无法准确表征像素间相关性的问题。因此，本文设置自适应参数 AP 和相对位置偏置 B 来增强位置信息，标记非相关区域。在 soft max 函数之外采用外部相对位置偏置 B_o 对非相关区域的注意力权重值进行限制。从而限制相似区域的大注意力权重，增加外观缺失物体与遮挡其物体的边界语义相关性，提高外观缺失物体分割准确性。

相对位置偏置生成流程图如图 4-5 所示。具体步骤如下：首先计算相对位置索引。输入 2×2 特征图，用绝对位置索引代表特征图中每个像素块的位置，将红色块的绝对位置索引依次减去其他位置索引，即可得到其他位置与红色块之间的相对位置索引，依次计算就能得到橙色、蓝色和绿色像素块的相对位置索引，将其按行展开即得到一个 4×4 的矩阵。相对位置偏置参数保存在长度为 $(2M-1)(2M-1)$ 的相对位置偏置表中，参数可通过查表得到并且参数可训练。

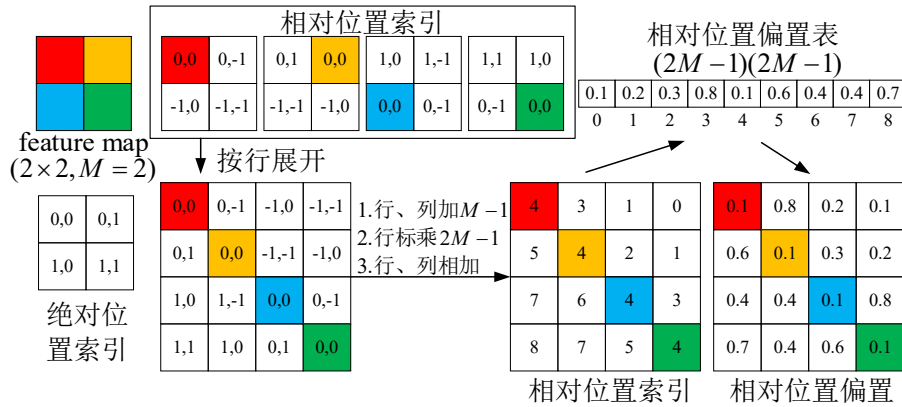


图 4-5 相对位置偏置生成流程图

为了增强网络获取位置信息的能力，采用一种外部相对位置偏置算法，该算法对输入元素之间的相关性分数进行点乘运算：

$$e_{ij} = x_i W^Q (x_j W^K)^T \cdot AP + B \quad (4-8)$$

其中， x_i 、 x_j 分别为图像块的输入； W^Q ， W^K 为可训练参数矩阵；

$AP \in \mathbb{R}^{M^2 \times M^2}$ 是与相对位置偏置一致，包含一组可学习参数的自适应参数；

$B \in \mathbb{R}^{M^2 \times M^2}$ 为相对位置偏置； M^2 为图像块数量。

采用 soft max 和外部相对位置偏置 B_o 计算权重系数 α_{ij} ，并进行线性变换求取每个输出的加权和 z_i ，如下式所示。

$$\alpha_{ij} = \text{soft max}(e_{ij}) + B_o = \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{ik})} + B_o \quad (4-9)$$

$$z_i = \sum_{j=1}^n \alpha_{ij} x_j W^V \quad (4-10)$$

其中， W^V 为参数矩阵。

4.3 基于双重约束的物体运动判断

虽然利用实例分割能区分动态物体，但是当动态物体处于静止状态时（如坐着的人、停在路边的车），若仅采用先验信息直接过滤动态物体则会剔除大量的静态点，因此，为进一步判断物体运动状态，通常采用运动一致性判断方法判断实例分割掩码内的特征点的运动状态。几何约束示意图如图 4-6 所示。

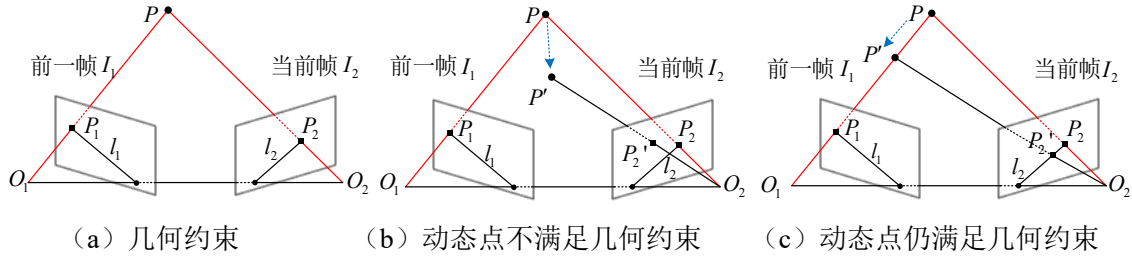


图 4-6 几何约束示意图

O_1 和 O_2 分别为前一帧 I_1 和当前帧 I_2 相机光学中心， P 为空间中的某一点，其在 I_1 、 I_2 上的投影分别为 P_1 、 P_2 ， P_1 、 P_2 的归一化像素坐标分别为 $X_1 = [u_1, v_1, 1]$ 、 $X_2 = [u_2, v_2, 1]$ ， F 为基础矩阵，应该满足：

$$X_2^T F X_1 = 0 \quad (4-11)$$

若 F 矩阵的计算足够准确，则 P_2 在极线 l_2 上，满足 (4-11) 式约束条件。假设 $Ax + By + C = 0$ 为极线所在方程， D 为 P_2 到极线距离，则此时满足：

$$D = \frac{|Au_2 + Bv_2 + C|}{\sqrt{\|A\|^2 + \|B\|^2}} = 0 \quad (4-12)$$

通过将 D 与设定阈值进行比较，从而判断空间点是否处于运动状态。然而在实际情况中，存在以下三种情况：(1)、由于特征提取和 F 矩阵估计具有不

确定性， P_2 可能不会准确地落在极线 l_2 上，不满足极线约束。(2)、空间点发生运动，如图 4-6 (b) 所示，相机由前一帧移动至当前帧位置时， P 点运动至 P' ， P' 在当前图像帧 I_2 的投影点 P'_2 未落在极线 l_2 上，因此不满足以上约束。(3)、当特征点沿相机坐标系的 Z 轴（光轴）方向运动时，如图 4-6 (c) 所示，空间点 P 点运动到 P' ，此时 P' 在当前图像帧 I_2 上的投影 P'_2 仍然处于极线 l_2 上，满足极线约束，但实际情况是 P 点为动态点。综上所述，只使用极线约束来确定动态点并不严谨，具有一定的局限性。因此本文提出基于双重约束的物体运动判断方法准确识别动态点。定义空间中 3D 点 S 在关键帧 K 上的实际投影点为 S_1 ，在当前帧 C 中的投影点为 $S_2(x_1, y_1)$ ，由当前帧 C 中与 S_1 匹配得到的点为 $S'_2(x_2, y_2)$ ，基于双重约束的物体运动判断方法如式 (4-13) 所示。

$$\begin{cases} T = \lambda D + \mu(\Delta d \vee \Delta z) \\ \Delta d = \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2} \\ \Delta z = |z_1 - z_2| \end{cases} \quad (4-13)$$

其中， T 表示静态点得分， D 为 S_2 到极线的距离，距离越接近 0，说明投影点 S_2 越有可能在当前帧的极线 l_2 上，则该点是动态点的可能性越小； Δd 为 S_2 与 S'_2 的欧式距离， Δz 为 P 点在当前帧的投影深度 z_1 与实际深度 z_2 的差值。 λ 和 μ 为权重，当距离 D 大于设定阈值 τ_D 时，取 $\lambda=1$ ， $\mu=0$ ，表示不需计算位置差与深度差便可确定动态点。否则取 $\lambda=0$ ， $\mu=1$ ，若 Δd 大于阈值 τ_d 或 Δz 大于阈值 τ_z 时，则认为 S 点为动态点。最后计算临界点 T 的得分并设定临界阈值 τ_T ，若 $T > \tau_T$ ，则确定 S 点为动态点。

4.4 实验结果与分析

为了验证本文算法的有效性，在公开 TUM 动态数据集分别进行潜在动态物体分割对比实验、基于双重约束的物体运动判断对比实验以及系统定位精度定量分析和定性评估。实验平台电脑配置为：Ubuntu18.04 版本 Linux 操作系统，Intel (R) Core (TM) i5-11400F @ 2.60GHz，显卡为 RTX3060，16G 内存。此外，搭建移动机器人实验平台进行真实环境建图对比实验，进一步验证本文算法的可行性。

4.4.1 潜在动态物体分割对比实验

为了验证所提 CRNET 在外观严重和轻微缺失下的实例分割性能，选取 TUM 数据集进行测试。CRNET 与 YOLACT++^[60]、Mask R-CNN 算法在 TUM 数据集下分割效果对比如图 4-7 所示。由图 4-7 第一行可知，由于 YOLACT++ 和 Mask R-CNN 缺少对物体外观缺失区域特征增强网络层，导致在外观严重缺失情况下发生了漏分割。而 CRNET 设计的残差坐标注意力模块，使其在关注物体外观缺失区域信息的同时对无用信息进行了有效抑制，从而增强了对物体的分割效果。由图 4-7 第二行可知，外观轻微缺失物体在图像帧中所占比例较大，YOLACT++ 和 Mask R-CNN 在模型中只使用了针对局部特征有效的卷积模块。CRNET 采用融合外部相对位置偏置的 Transformer 模块，其中外部相对位置偏置根据位置信息重新分配权重，增加外观缺失物体的边界语义信息。而 Transformer 能有效获取像素全局依赖关系，提升对图像中所占比例较大目标的识别率，因此 CRNET 对避免由于物体外观缺失造成的漏分割具有显著效果。

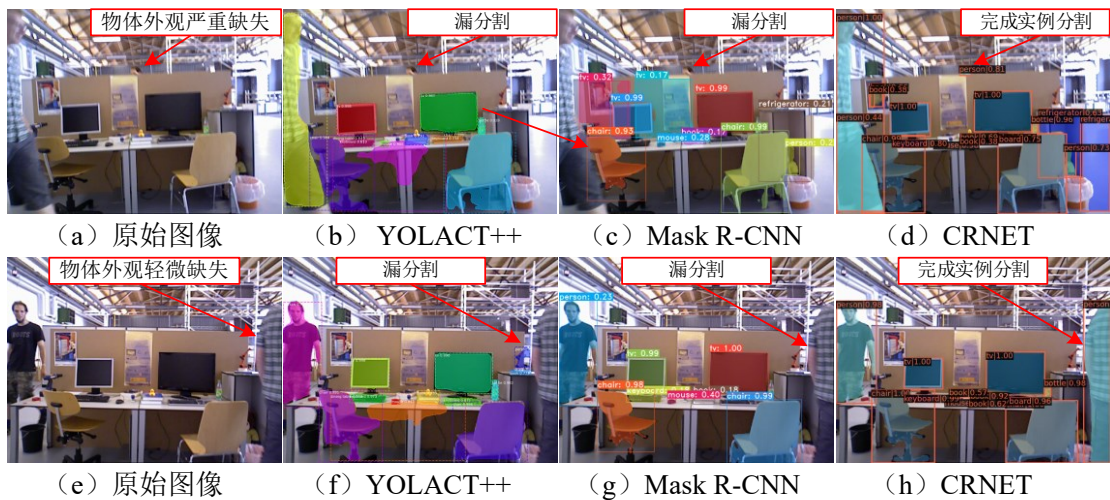


图 4-7 潜在动态物体分割对比

CRNET 与不同算法的 AP 值对比如表 4-2 所示，由表中数据可知，CRNET 算法的 AP 值与 YOLACT++、Mask R-CNN 算法相比分别提高了 14.7%、11.2%。

表 4-2 CRNET 与不同算法的 AP 值对比

算法	mAP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
PA-NET ^[61]	36.6	58.0	39.3	16.3	38.1	53.1
FCIS ^[62]	29.5	51.5	30.2	8.0	31.0	49.7
YOLACT++	34.6	53.8	36.9	11.9	36.8	55.1
Mask R-CNN	35.7	58.0	37.8	15.5	38.1	52.4
CRNET	39.7	60.3	43.1	18.5	56.7	71.2

4.4.2 基于双重约束的物体运动判断对比实验

为了验证所提基于双重约束的物体运动判断策略的有效性，将本文算法与 ORB-SLAM3、Crowd-SLAM 以及 DynaSLAM 算法特征点提取效果进行对比，如图 4-8 所示。由图 4-8 (a) ~ (d) 可知，ORB-SLAM3 由于缺乏识别和判断动态物体的有效策略，动态点剔除效果较差且连续剔除动态点能力较弱。由图 4-8 (e) ~ (h) 可知，Crowd-SLAM 利用 YOLO 网络检测动态物体，再剔除检测框内的特征点，减少了动态点的干扰，但该方法未考虑检测框内也存在静态点。由图 4-8 (i) ~ (l) 可知，DynaSLAM 可将动态物体进行剔除，但其通过计算前后帧中同一个特征点的变换角度判断该点是否运动的方法易受到角度、外部光照变化干扰导致无法准确去除如图红色矩形框内的动态点。图 4-8 (m) ~ (p) 为本文算法特征提取图，该算法通过 CRNET 实例分割网络分割潜在动态物体，并利用基于双重约束的物体运动判断策略判断物体运动状态并对动态点进行剔除，由图可知本文算法能准确判断物体运动状态并对动态点进行剔除。

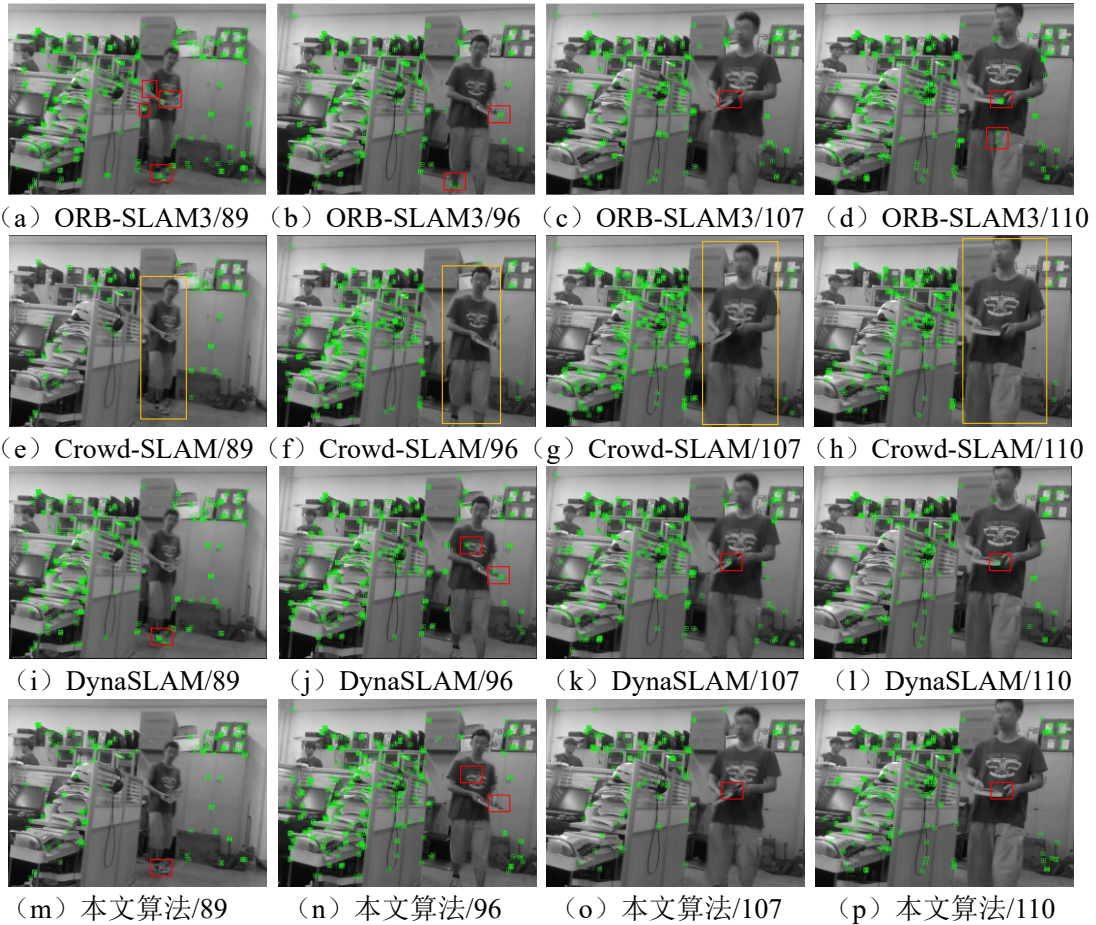


图 4-8 特征点提取效果对比 (89~110 帧)

4.4.3 公开 TUM 数据集评估实验

为了验证本文算法在动态场景下的有效性，采用 TUM 数据集中的 fr3_walking 动态序列对本文所提算法进行仿真实验验证。TUM 数据集包含 RGB 图像、深度图像以及序列所对应的相机运动真实轨迹文件。在 fr3_walking 序列中，有两个人来回的走动并移动椅子，因此该序列属于高动态序列。本次实验选取 fr3_walking_half、fr3_walking_rpy、fr3_walking_xyz 以及 fr3_walking_static 四组序列验证本文所提算法在动态环境下的鲁棒性，其中，half 表示相机在半球形轨迹上移动，rpy 表示相机绕主轴旋转，xyz 表示相机沿着空间中 x、y 和 z 轴运动，static 表示将相机保持在适当位置。为了便于表达，使用 fr3/w/half，fr3/w/rpy，fr3/w/xyz 以及 fr3/w/static 来表示以上四组序列。本次实验采用绝对轨迹误差（ATE）与其均方根误差（RMSE）和相对位姿误差（RPE）与其均方根误差（RMSE）进行定量分析，其中 ATE 为位姿真值和估计位姿间的差值，反映算法的定位精度和轨迹全局一致性，RPE 为每隔固定时间位姿真值和估计位姿的误差，反映了视觉里程计误差。RMSE 反应轨迹位姿和真实位姿的欧式距离，值越小，精度越高。设 $Q_1, \dots, Q_n \in SE(3)$ 表示真实位姿， $P_1, \dots, P_n \in SE(3)$ 表示位姿估计值，下标为时间 t （或帧），相隔固定时间用 Δ 表示，则第 i 帧的 ATE F_i 与其 RMSE 以及第 i 帧的 RPE E_i 与其 RMSE 分别定义如下：

$$F_i = X_i^{-1} S Q_i$$

$$RMSE(F_{1:n}, \Delta) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|trans(F_i)\|^2} \quad (4-14)$$

$$E_i = (Q_i^{-1} Q_{i+\Delta})(P_i^{-1} P_{i+\Delta})$$

$$RMSE(E_{1:n}, \Delta) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|trans(E_i)\|^2} \quad (4-15)$$

本文算法与 ORB-SLAM3 算法三维空间绝对轨迹误差热力图对比如图 4-9 所示，其中，灰色 Reference 为真实轨迹，有颜色的线条为估计轨迹，颜色从深蓝色偏向深红色表示系统误差由小变大。由图可知，由于 ORB-SLAM3 算法缺乏识别潜在动态物体的有效策略导致其地图构建效果较差，在四个序列上的误差值明显较大。本文利用 CRNET 网络分割动态物体，并通过基于双重约束的物体运动判断策略准确剔除动态点，减少了动态物体对位姿估计的干扰，因此本文算法所建立的轨迹图更接近真实轨迹。

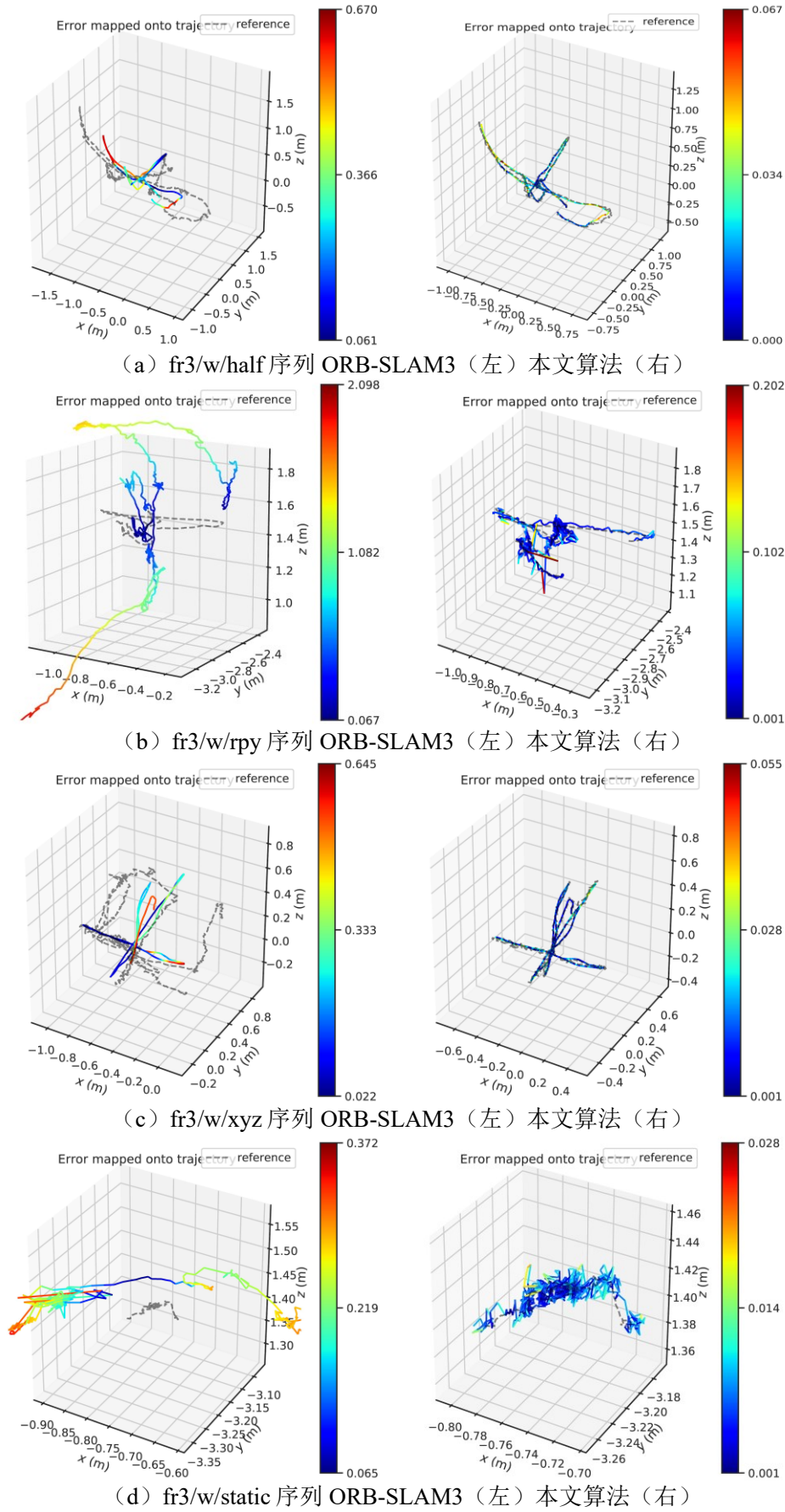


图 4-9 三维空间绝对轨迹误差热力图对比

本文算法与 ORB-SLAM3 算法在 TUM 四个序列的平移相对位姿误差对比如图 4-10 所示。其中，横坐标表示算法运行的时间，单位为秒 (s)；纵坐标表示真实位姿与算法在某时刻所估计位姿之间的平移相对位姿误差，单位为米 (m)。由图可知，与 ORB-SLAM3 算法相比，本文算法位姿估计精度更高。

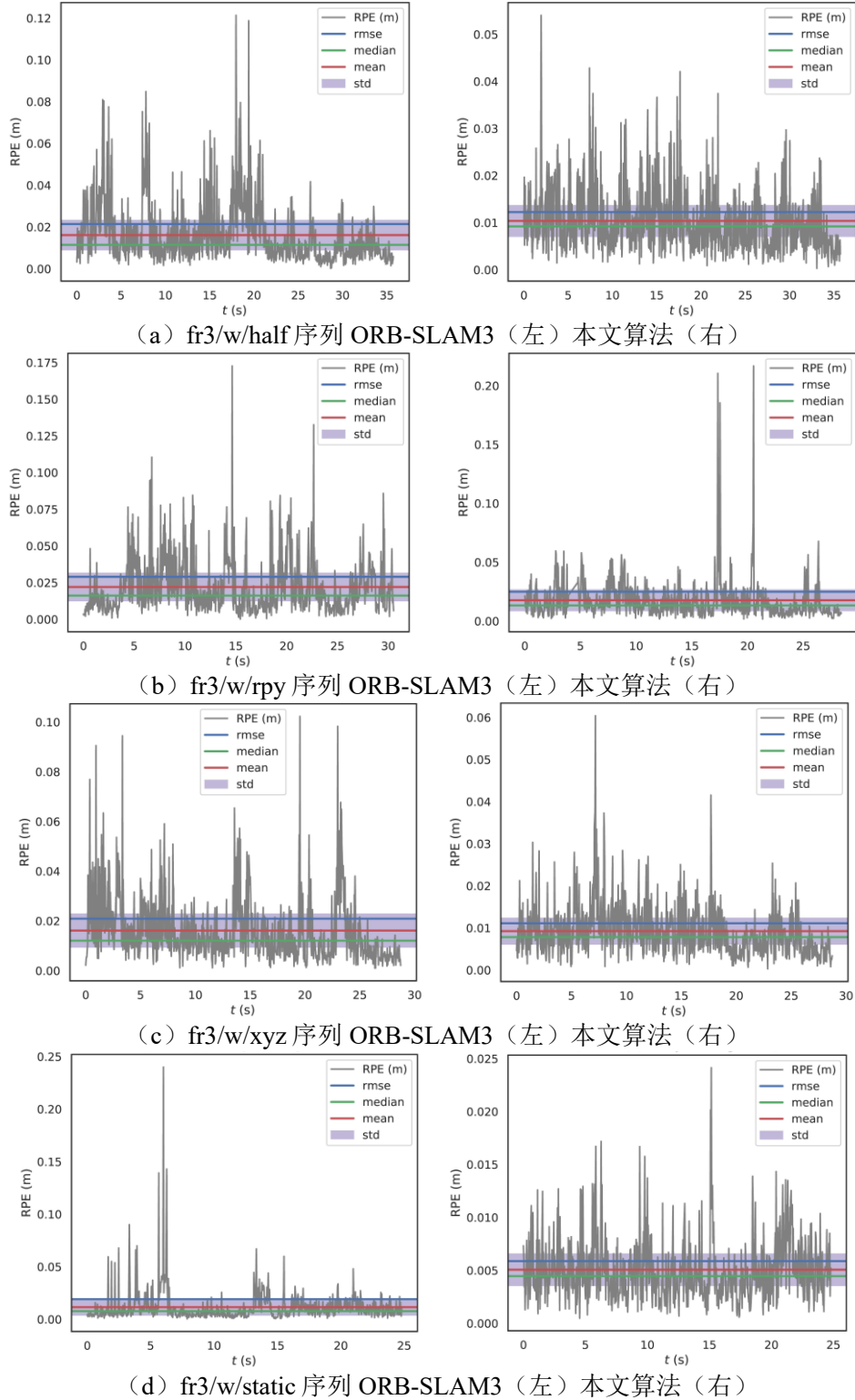


图 4-10 平移相对位姿误差对比

本文算法与 ORB-SLAM3 算法的 ATE、平移 RPE 以及旋转 RPE 在 TUM 四个序列上的对比结果如表 4-3~表 4-5 所示。由表可知，本文算法与 ORB-SLAM3 相比 ATE RMSE、平移 RPE RMSE、旋转 RPE RMSE 分别降低了 93%~97.39%、13.79%~68.42%、16.01%~55.26%。

表 4-3 本文算法与 ORB-SLAM3 算法的 ATE 对比结果（单位：m）

TUM 序列	ORB-SLAM3			本文算法			提升幅度/(%)		
	RMSE	Median	Mean	RMSE	Median	Mean	RMSE	Median	Mean
w/half	0.357	0.283	0.321	0.025	0.020	0.022	93.00	92.93	93.15
w/rpy	0.914	0.640	0.761	0.032	0.020	0.025	96.50	68.75	96.71
w/xyz	0.324	0.249	0.262	0.015	0.012	0.013	95.37	95.18	95.04
w/static	0.268	0.263	0.261	0.007	0.006	0.006	97.39	97.72	97.70

表 4-4 本文算法与 ORB-SLAM3 算法的 RPE（平移部分）对比结果（单位：m）

TUM 序列	ORB-SLAM3			本文算法			提升幅度/(%)		
	RMSE	Median	Mean	RMSE	Median	Mean	RMSE	Median	Mean
w/half	0.021	0.014	0.016	0.012	0.009	0.010	42.86	35.71	37.50
w/rpy	0.029	0.016	0.022	0.025	0.013	0.018	13.79	18.75	18.18
w/xyz	0.021	0.012	0.016	0.011	0.011	0.008	47.62	8.33	50.00
w/static	0.019	0.008	0.012	0.006	0.005	0.005	68.42	37.50	58.33

表 4-5 本文算法与 ORB-SLAM3 算法的 RPE（旋转部分）对比结果（单位：°）

TUM 序列	ORB-SLAM3			本文算法			提升幅度/(%)		
	RMSE	Median	Mean	RMSE	Median	Mean	RMSE	Median	Mean
w/half	0.537	0.357	0.435	0.396	0.298	0.340	26.26	16.53	21.84
w/rpy	0.662	0.444	0.534	0.556	0.313	0.404	16.01	29.50	24.34
w/xyz	0.513	0.291	0.377	0.383	0.226	0.273	25.34	22.34	27.59
w/static	0.380	0.213	0.268	0.170	0.131	0.146	55.26	38.50	45.52

为了进一步验证本文算法的有效性，将本文算法与 DS-SLAM、DynaSLAM 等先进动态 SLAM 算法的 ATE 进行对比，如表 4-6 所示。由表中数据可知，本文算法在大部分序列表现最好。DynaSLAM 和 DS-SLAM 算法先利用语义分割网络检测潜在动态物体，然后采用运动判断策略过滤动态点从而提升动态场景下的位姿估计精度，但这两种算法在物体外观缺失环境下存在潜在动态物体识别率不高、以及难以准确估计潜在动态物体运动状态的问题，最终影响系统定位精度。而本文算法先采用 CRNET 准确分割动态物体以及外观缺失物体，然后采用基于双重约束的物体运动判断方法准确去除动态点，因此本文算法与 DS-

SLAM、Crowd-SLAM 以及 DynaSLAM 算法相比 ATE RMSE 分别平均减少 42.94%、20.79%、11.79%。

表 4-6 本文算法与其他先进算法的 ATE 对比结果（单位：m）

TUM 序列	Crowd-SLAM		DS-SLAM		DynaSLAM		本文算法	
	RMSE	Median	RMSE	Median	RMSE	Median	RMSE	Median
w/half	0.026	0.024	0.034	0.022	0.032	0.025	0.025	0.020
w/rpy	0.055	0.027	0.444	0.284	0.037	0.035	0.032	0.020
w/xyz	0.020	0.014	0.025	0.015	0.017	0.015	0.015	0.012
w/static	0.008	0.006	0.008	0.005	0.007	0.007	0.007	0.006

4.4.4 真实场景实验

为了进一步验证本文算法的有效性，搭建 Husky 移动机器人实验平台，如图 4-11 所示。Husky 移动机器人搭载 D435i 相机、360° 激光雷达等用于采集外部环境信息以及移动机器人自身的运动信息，主要参数设置如表 4-7 所示。



图 4-11 Husky 移动机器人实验平台

通过内置 24V 20Ah 密封铅酸电池为 Husky 的续航能力提供保障。Husky 使用 ROS 消息-订阅机制创建相机节点，通过 D435i 相机实时获取外部环境图像信息，可实现在线调参，实时建图与导航。Husky 移动机器人硬件平台由用户层、运动控制层以及传感器层共同组成。用户层采用 CPU i7-10875H 处理器，内存 8G，GPU 为 GTX1080，操作系统为 Ubuntu18.04 PC 端，使用 PC 端接收 Husky 获取的数据信息并进行实时处理，同时，在 PC 端可通过指令控制机器人运行的速度和方向；运动控制层是指 Husky 控制的运动设备；传感器层使用不同传感器获取外部环境信息和机器人自身运动信息。本章真实场景实验利用 D435i 相

机获取外部环境图像信息，D435i 相机和主要参数分别如图 4-12 和表 4-7 所示。

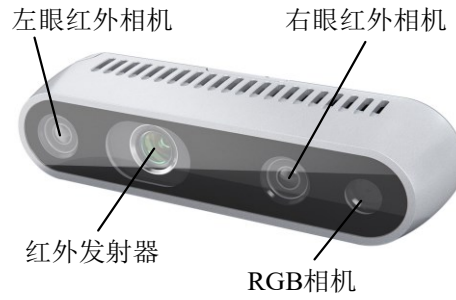


图 4-12 D435i 相机

表 4-7 主要参数设置

变量名称	变量名称	参数
机器人参数	运行速度	$1m/s$
	转速	$0.48rad/s$
	方向变化范围	$[0, 2\pi]$
相机参数	相机采样频率	$30fps$
	RBB 分辨率 (PPI)	$\leq 1920 \times 1080$
	RGB 传感器技术	卷帘快门

为了进一步验证所提算法在真实场景中的有效性，设置室内和室外动态场景进行测试，如图 4-13 所示。在室内场景测试中，移动机器人从起点绕会议桌 1 和会议桌 2 进行绕“8”字大曲率弯道运动且在运行途中遭遇动态物体，最后到达终点位置。在室外场景测试中，移动机器人从 A 点出发沿红色箭头路线运行，最终回到 A 点，其中 B、C、D 等字母标注位置为机器人相机视角画面，室外场景存在大量光照变化、大曲率弯道运动以及动态物体干扰，总行程约 500m。本次实验以移动机器人的激光雷达数据作为真实参照轨迹。

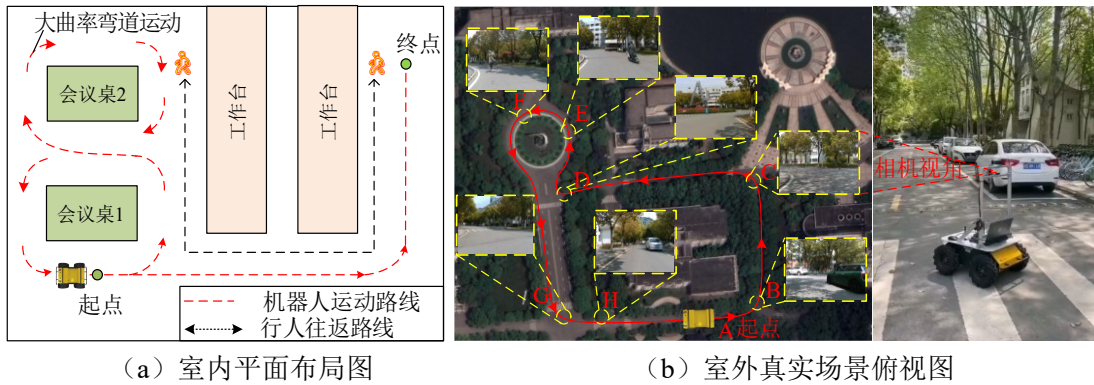


图 4-13 真实场景实验环境

移动机器人在室内运行过程中获取的两帧分割效果图如图 4-14 所示，本文提出的 CRNET 算法对物体外观严重缺失和轻微缺失情况均能完成分割。移动机器人特征点提取效果图如图 4-15 所示。本文算法首先采用所提 CRNET 分割网

络对动态物体进行准确分割，并采用基于双重约束的物体运动判断策略准确判断物体运动状态并过滤动态点，增强了算法在动态环境下的鲁棒性。

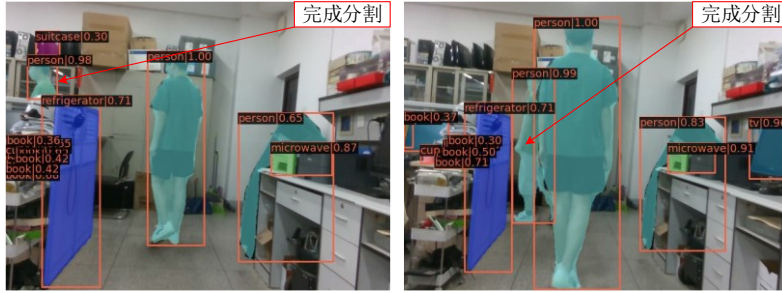


图 4-14 潜在动态物体分割效果图



图 4-15 特征点提取效果

本文算法和 DynaSLAM 算法在室内和室外真实场景构建的轨迹对比图如图 4-16 所示。由图可知，本文算法更加接近真实轨迹，而 DynaSLAM 算法在运行中出现了轨迹漂移最终导致建图误差较大。本文算法采用 CRNET 实例分割网络精确标记动态物体，并采用基于双重约束的物体运动判断策略去除动态点，因此本文算法定位精度更高。

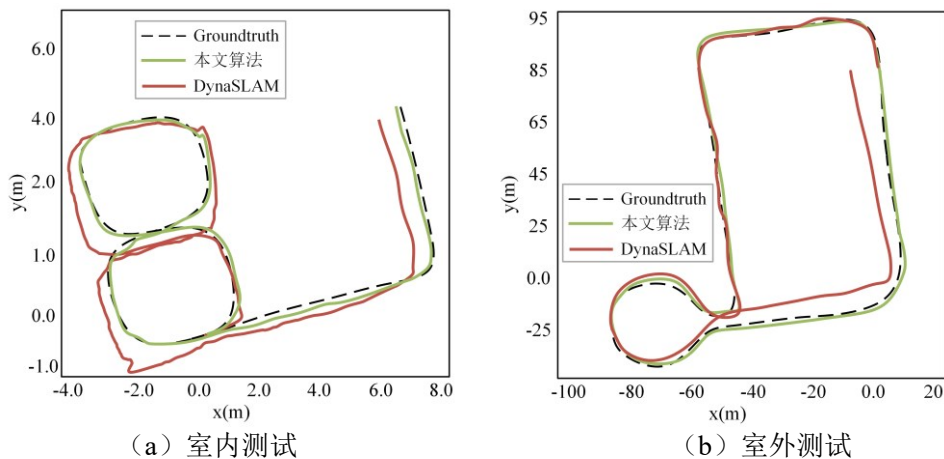


图 4-16 2 种算法轨迹对比

真实场景实验的关键帧选取与绝对轨迹误差 RMSE 对比如表 4-8 所示，DynaSLAM 算法在大曲率弯道运动下易发生关键帧丢失，而本文算法采取自适应阈值关键帧选取策略，增补了曲线运动漏选的关键帧。此外本文算法前端采

用改进的 SURF 算法更能适应光照变化等复杂场景、动态 SLAM 线程采用 CRNET 网络识别潜在动态物体并利用双重约束策略进一步判断潜在动态物体的运动状态，因此本文算法定位精度更高。

表 4-8 真实场景关键帧选取与绝对轨迹误差 RMSE 对比

场景	DynaSLAM		本文算法	
	关键帧 (个)	RMSE (m)	关键帧 (个)	RMSE (m)
室内	867	0.033	924	0.027
室外	1526	2.014	1725	1.624

4.5 本章小结

本章提出了一种动态场景下基于改进 Transformer 的视觉 SLAM 算法，针对现有分割网络易出现动态物体分割不完整的问题，提出一种融合坐标注意力机制与相对位置偏置的实例分割网络 CRNET，增强对外观缺失物体的分割能力避免漏分割；与此同时，提出一种双重约束的物体运动判断策略解决了潜在动态物体运动状态判断不准确的问题。在公开 TUM 动态数据集和真实环境进行测试，结果显示本文算法与 ORB-SLAM3、DS-SLAM、Crowd-SLAM 以及 DynaSLAM 算法相比定位精度更高，展现出在动态场景下良好的构图能力。

第5章 全文总结与展望

5.1 全文总结

本文从视觉 SLAM 和动态场景下视觉 SLAM 研究现状出发, 以实现复杂动态场景下移动机器人准确性定位与高精度建图为目标, 深入分析复杂动态场景移动机器人面临的挑战, 通过图像特征提取与匹配、关键帧选取以及潜在动态物体分割与判断线程共同构建一种面向动态场景的视觉 SLAM 算法。该算法有效改善了移动机器人在动态场景下遭遇光照变化、图像模糊、尺度变化以及大曲率弯道运动导致跟踪丢失、轨迹漂移等问题, 提高了视觉 SLAM 系统的定位精度和鲁棒性。

论文主要内容及创新点概括如下:

(1) 针对现有视觉 SLAM 算法由于光照变化、图像模糊、尺度变化易导致特征提取与匹配困难的问题, 提出一种基于改进 SURF 特征的图像配准算法, 首先利用局部二维熵进行特征点检测提升特征点的独特性, 其次引入曼哈顿距离构建距离评价函数进行预处理, 剔除误匹配的特征点, 然后利用粒子群算法进行匹配优化, 提升算法的实时性。经过实验验证, 改进的 SURF 算法与改进前相比在光照变化、图像模糊以及尺度变化场景特征提取与匹配的准确率分别提升了 9%、5.7%、4%, 增强了算法在复杂场景下的鲁棒性。

(2) 针对移动机器人在大曲率弯道运动下易发生漏选关键帧造成位姿估计失准、闭环准确性较低等问题, 提出了一种基于自适应阈值的关键帧选取策略进行关键帧的选取, 增补大曲率弯道运动拐点附近遗漏的有效关键帧, 增强算法对大曲率弯道运动场景信息的感知能力。通过在公开的 KITTI 和 EuRoC 数据集进行验证, 结果表明本文算法与固定阈值关键帧选取策略相比绝对轨迹误差降低了 22%。

(3) 针对现有动态 SLAM 算法对物体分割不完整、难以准确判断潜在动态物体的运动状态等问题, 提出了一种 CRNET 实例分割网络模型, 有效避免了物体分割不完整甚至漏分割; 与此同时, 提出一种基于双重约束的物体运动判断策略准确估计潜在动态物体的运动状态, 提高了移动机器人在复杂动态场景下

的建图能力。在公开的 TUM 数据集进行仿真实验，结果显示所提算法与 DS-SLAM 算法、Crowd-SLAM 算法以及 DynaSLAM 算法相比绝对轨迹误差 RMSE 分别平均减少 42.94%、20.79%、11.79%，展现出在复杂动态场景下良好的构图能力。与此同时，利用自主搭建的移动机器人实验平台在真实环境对算法进行测试，结果表明本文算法与 DynaSLAM 算法相比在定位精度方面具有更大优势。

5.2 研究展望

本文通过对“动态场景下基于深度学习的视觉 SLAM 算法”展开研究，旨在解决视觉 SLAM 算法在动态场景下遭遇光照变化、图像模糊、尺度变化特征提取与匹配困难、大曲率弯道运动关键帧选取不恰当以及对动态物体分割不完整难以准确判断物体运动状态等问题。然而，本文还存在一些问题需要解决，主要包括以下两个方面。

(1) 虽然本文视觉 SLAM 系统前端所采用的改进 SURF 图像配准方法能在一定程度上应对光照变化、图像模糊以及尺度变化，但是在遭遇如长走廊、大白墙等低纹理场景时，特征点提取的重复性低，匹配性能较差。解决此问题可考虑如下两种方案：一是利用线特征或者面特征实现低纹理环境感知；二是融合惯性测量单元 (IMU) 数据，增强算法在低纹理场景下的鲁棒性。

(2) 在当前的工作中，将动态物体的信息视为干扰并将其全部剔除，但在实际应用中，准确估计动态物体的运行信息是至关重要的。移动机器人在运行过程中难免会遭遇动态物体，移动机器人需要通过动态物体的速度、大小、运动趋势等信息准确估计与其发生碰撞的概率并及时调整自身运动状态或重新进行路径规划。因此如何对动态物体进行准确的跟踪，有效利用其信息帮助移动机器人实现更高级别的任务是下一步的研究方向。

参考文献

- [1] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on robotics, 2016, 32(6): 1309-1332.
- [2] PLANTARD P, H. SHUM H P, MULTON F. Filtered pose graph for efficient kinect pose reconstruction[J]. Multimedia Tools and Applications, 2017, 76: 4291-4312.
- [3] PLACED J A, STRADER J, CARRILLO H, et al. A survey on active simultaneous localization and mapping: State of the art and new frontiers[J]. IEEE Transactions on Robotics, 2023, 39(3): 1686-1705.
- [4] YIN H, LI S, TAO Y, et al. Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments[J]. IEEE Transactions on Robotics, 2022, 39(1): 289-308.
- [5] 王柯赛, 姚锡凡, 黄宇, 等. 动态环境下的视觉 SLAM 研究评述[J]. 机器人, 2021, 43(6): 715-732.
- [6] XIE H, ZHANG D, WANG J, et al. Semi-Direct Multimap SLAM System for Real-Time Sparse 3-D Map Reconstruction[J]. IEEE Transactions on Instrumentation and Measurement, 2023, 72: 1-13.
- [7] PIAO J-C, KIM S-D. Real-time visual-inertial SLAM based on adaptive keyframe selection for mobile AR applications[J]. IEEE Transactions on Multimedia, 2019, 21(11): 2827-2836.
- [8] 陈孟元, 韩朋朋, 刘金辉, 等. 动态遮挡场景下基于改进 Transformer 实例分割的 VSLAM 算法[J]. 电子学报, 2023, 51(7): 1812-1825.
- [9] SMITH R C, CHEESEMANN P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56-68.
- [10] 赵功洋, 闫利. 移动机器人 SLAM 位姿估计的改进四元数无迹卡尔曼滤波[J]. 测绘学报, 2022, 51(2): 212-223.
- [11] 李天成, 范红旗, 孙树栋. 粒子滤波理论、方法及其在多目标跟踪中的应用[J]. 自动化学报, 2015, 41(12): 1981-2002.
- [12] LI Y, BRASCH N, WANG Y, et al. Structure-slam: Low-drift monocular slam in indoor

- p>environments[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6583-6590.
- [13]JULIER S, UHLMANN J, DURRANT-WHYTE H F. A new method for the nonlinear transformation of means and covariances in filters and estimators[J]. IEEE Transactions on Automatic Control, 2000, 45(3): 477-482.
- [14]PU H, LUO J, WANG G, et al. Visual SLAM integration with semantic segmentation and deep learning: A review[J]. IEEE Sensors Journal, 2023, 23(19): 22119-22138.
- [15]MOURAGNON E, LHUILLIER M, DHOME M, et al. Real time localization and 3D reconstruction[C]. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA: IEEE, 2006: 363-370.
- [16]SCHONBERGER J L, FRAHM J-M. Structure-from-motion revisited[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA: IEEE, 2016: 4104-4113.
- [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, Nara, Japan: IEEE, 2007: 225-234.
- [19]VISWANATHAN D G. Features from Accelerated Segment Test (FAST)[C]. Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK, 2009: 6-8.
- [20]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.
- [21]RUBLEE E, RABAU V, KONOLIGE K, et al. ORB: An efficient alternative to SIFT or SURF[C]. 2011 International conference on computer vision, Barcelona, Spain: IEEE, 2011: 2564-2571.
- [22]ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 611-625.
- [23]MUR-ARTAL R, TARDÓ S 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]ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European conference on computer vision, Springer, 2014: 834-849.
- [26]LEUTENEGGER S, LYNEN S, BOSSE M, et al. Keyframe-based visual-inertial odometry using nonlinear optimization[J]. The International Journal of Robotics Research, 2015, 34(3): 314-334.
- [27]QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
- [28]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(1): 45-52.
- [29]GUO S, RONG Z, WANG S, et al. A LiDAR SLAM with PCA-based feature extraction and two-stage matching[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-11.
- [30]陈孟元, 丁陵梅, 张玉坤. 基于改进关键帧选取策略的快速 PL-SLAM 算法[J]. 电子学报, 2020, 50(3): 608-618.
- [31]UL ISLAM Q, IBRAHIM H, CHIN P K, et al. FADM-SLAM: a fast and accurate dynamic intelligent motion SLAM for autonomous robot exploration involving movable objects[J]. Robotic Intelligence and Automation, 2023, 43(3): 254-266.
- [32]XING Z, ZHU X, DONG D. DE-SLAM: SLAM for highly dynamic environment[J]. Journal of Field Robotics, 2022, 39(5): 528-542.
- [33]艾青林, 刘刚江, 徐巧宁. 动态环境下基于改进几何与运动约束的机器人 RGBD SLAM 算法[J]. 机器人, 2021, 43(2): 167-176.
- [34]LI S, LEE D. RGB-D SLAM in dynamic environments using static point weighting[J]. IEEE Robotics and Automation Letters, 2017, 2(4): 2263-2270.
- [35]LIU Y, MIURA J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods[J]. IEEE Access, 2021, 9: 23772-23785.
- [36]SOARES J C V, GATTASS M, MEGGIOLARO M A. Crowd-SLAM: visual SLAM towards crowded environments using object detection[J]. Journal of Intelligent & Robotic

- Systems, 2021, 102(2): 50.
- [37]张继贤, 刘飞. 视觉 SLAM 环境感知技术现状与智能化测绘应用展望[J]. 测绘学报, 2023, 52(10): 1617-1630.
- [38]XIAO L, WANG J, QIU X, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.
- [39]WU W, GUO L, GAO H, et al. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint[J]. Neural Computing and Applications, 2022, 34: 6011-6026.
- [40]YU C, LIU Z, LIU X-J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain: IEEE, 2018: 1168-1174.
- [41]王梦瑶, 宋薇. 动态场景下基于自适应语义分割的 RGB-D SLAM 算法[J]. 机器人, 2023, 45(1): 16-27.
- [42]BESCOS B, FÁCIL J M, CIVERA J, et al. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.
- [43]ZHANG Z. Flexible camera calibration by viewing a plane from unknown orientations[C]. Proceedings of the seventh IEEE international conference on computer vision, Kerkyra, Greece: IEEE, 1999: 666-673.
- [44]BAY H, TUYTELAARS T, VAN GOOL L. SURF: Speeded Up Robust Features[C]. Computer Vision—ECCV 2006, Graz, Austria: Springer, 2006: 404-417.
- [45]VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [46]DE LA ROSA J I, GUTIÉRREZ O, VILLA-HERNÁNDEZ J, et al. Entropy estimation for robust image segmentation in presence of non Gaussian noise[J]. Multimedia Tools and Applications, 2021, 80(5): 6991-7021.
- [47]YANG W, CAI L, WU F. Image segmentation based on gray level and local relative entropy two dimensional histogram[J]. Plos one, 2020, 15(3): e0229651.
- [48]SILVA L E, DUQUE J J, FELIPE J C, et al. Two-dimensional multiscale entropy analysis: Applications to image texture evaluation[J]. Signal Processing, 2018, 147: 224-232.

- [49]陈雪松, 陈秀芳, 毕波, 等. 基于改进 SURF 的图像匹配算法[J]. 计算机系统应用, 2020, 29(12): 222-227.
- [50]KENNEDY J, EBERHART R. Particle swarm optimization[C]. Proceedings of ICNN'95-international conference on neural networks, Perth, WA, Australia: IEEE, 1995: 1942-1948.
- [51]HOSSEININAVEH A, SERPICO M, ROBSON S, et al. Automatic image selection in photogrammetric multi-view stereo methods[C]. IEEE Conference on Visual Analytics Science and Technology, Eurographics Association, 2012.
- [52]GEIGER A, LENZ P, STILLER C, et al. Vision meets robotics: The kitti dataset[J]. The International Journal of Robotics Research, 2013, 32(11): 1231-1237.
- [53]BURRI M, NIKOLIC J, GOHL P, et al. The EuRoC micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157-1163.
- [54]SCHUBERT D, GOLL T, DEMMEL N, et al. The TUM VI benchmark for evaluating visual-inertial odometry[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain: IEEE, 2018: 1680-1687.
- [55]HOU Q, ZHOU D, FENG J. Coordinate attention for efficient mobile network design[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021: 13713-13722.
- [56]LIU Z, LIN Y, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]. Proceedings of the IEEE/CVF international conference on computer vision, 2021: 10012-10022.
- [57]LIN T-Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA: IEEE, 2017: 2117-2125.
- [58]HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA: IEEE, 2018: 7132-7141.
- [59]WOO S, PARK J, LEE J-Y, et al. CBAM: Convolutional block attention module[C]. Proceedings of the European conference on computer vision (ECCV), Munich, Germany: Springer, 2018: 3-19.
- [60]ZHOU C. Yolact++ Better Real-Time Instance Segmentation [M]. University of California,

Davis, 2020.

[61]LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C].

Proceedings of the IEEE conference on computer vision and pattern recognition, 2018:
8759-8768.

[62]LI Y, QI H, DAI J, et al. Fully convolutional instance-aware semantic segmentation[C].

Proceedings of the IEEE conference on computer vision and pattern recognition, 2017:
2359-2367.

攻读学位期间发表的学术论文和科研成果目录

一、 发表学术论文

- 【1】 **Tao Xu**, Mengyuan Chen*, Jinhui Liu. TKO-SLAM: Visual SLAM algorithm based on time-delay feature regression and keyframe pose optimization, Journal of Field Robotics (SCI), 2024. (WOS: 001216071100001).
- 【2】 陈孟元*, **徐韬**, 张坦坦. 大曲率运动下 SLAM 算法关键帧选取策略, 中国惯性技术学报 (EI 收录) (录用).
- 【3】 **徐韬**, 陈孟元*, 刘晓晓, 韩朋朋. 动态场景下基于注意力机制与几何约束的 VSLAM 算法. 传感技术学报(CSCD 核心库收录), 2023, 36 (09): 1395-1406.

二、 授权发明专利

- 【1】 **徐韬**, 陈孟元. 一种基于双重约束的机器人地图构建方法、存储介质及设备, 授权公告号: CN 116067360 B, 专利号: 2023100312943.
- 【2】 陈孟元, **徐韬**, 韩朋朋. 一种基于自适应关键帧选取的机器人地图构建方法、存储介质及设备, 授权公告号: CN 116399326 B, 专利号: 2023103839309.

三、 参加学科竞赛

- 【1】 **徐韬**, 杨苏朋, 等. 第十八届“挑战杯”全国大学生课外学术科技作品竞赛全国三等奖.

致谢

硕士研究生生涯即将结束，在这段时间收获颇多，借此机会对在求学生涯进行帮助的人们表示衷心的感谢。

感谢我的导师陈孟元教授。本论文从选题到开题以及最终成文都是在陈老师的精心指导下完成的。陈老师是我的良师益友，他平时乐于和我讨论研究进展，总能在我遇到困难时引导我找到解决问题的方法，不仅教会了我学术研究的思维方式，还教会我许多为人处事的方法；在生活中，陈老师关心学生，鼓励大家在进行科研的同时也要健康生活，以丰富的人生经验为我指明前进的方向，令我受益终生。

感谢我的师兄张玉坤、韩朋朋以及刘金辉对我的帮助和指导，在你们的帮助下，我得以快速适应研究生的角色。感谢实验室同门刘晓晓、姚满宇等，在你们身上我学习到了很多，遇到难题也会一起讨论，同时也感谢实验室师弟杨苏朋、张坦坦、唐哲等，你们的到来为实验室增加了活力。

感谢安徽工程大学电气工程学院老师们的辛勤付出，在这里我收获了知识和成长，每一位老师的教诲我都铭记在心。感谢电气研 21 的篮球小伙伴们，感谢你们陪我共同度过这段岁月，科研之余与你们在风雨操场、宿舍篮球场挥汗如雨的日子是如此让人放松。

感谢我的家人，感谢他们为我创造的良好生活环境，让我的求学之路没有后顾之忧，同时感谢李玉雪一直以来对我的支持，让我每一次受挫之后都能重塑信心，继续前行。再次感谢每一个给予我帮助、关心和支持的人，我一定会不忘初心、继续前行，做一个对国家和社会有所贡献的人。

尚德敏学 唯实惟新

