分类号：TP242.6　　学校代码：10363
密　级：公开　　学　　号：2210320109

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目　动态场景下基于改进Transformer实例分割的视觉SLAM算法研究

论 文 作 者　钱润邦
指 导 教 师　陈孟元 教授
学科（专业）　控制科学与工程
研 究 方 向　模式识别与智能系统

论文提交日期：2024 年 5 月 31 日

分类号：TP242.6　　单位代码：10363
密　级：公 开　　学　　号：2210320109

题　目　动态场景下基于改进 Transformer 实例分割的视觉 SLAM 算法研究

题　目　Research on Visual SLAM Algorithm Based on Improved Transformer Instance Segmentation in Dynamic Scenes

学生姓名： 钱润邦

校内导师： 陈孟元（教授）

专业学位类别： 控制科学与工程

研究方向（领域）： 模式识别与智能系统

论文答辩日期：2024 年 05 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：钱润邦

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：钱润邦　　指导教师签名：

日期：2024 年 5 月 30 日　　日期：2024 年 5 月 30 日

动态场景下基于改进 Transformer 实例分割的视觉 SLAM 算法研究

摘　要

同步定位与地图构建（Simultaneous Localization And Mapping，SLAM）是移动机器人通过配置的传感器获取环境信息，完成在未知环境下的定位，同时构建地图。SLAM 算法主要分为激光 SLAM 与视觉 SLAM(Visual SLAM,VSLAM)两大类。由于激光 SLAM 在价格和获取特征信息方面与视觉 SLAM 相比存在一定的劣势，因此使用视觉传感器的 SLAM 技术逐渐成为主流的研究方向。目前多数视觉 SLAM 算法适用的场景是静态的，若无法准确获取动态场景中物体的运动状态并筛选出动态物体，将提高视觉 SLAM 系统误差，甚至导致视觉 SLAM 系统无法在动态场景中正常运行。目前大多数视觉 SLAM 算法融入了实例分割网络标记场景潜在动态物体，但传统实例分割网络分割精度低，无法完整分割潜在动态物体，从而难以完整剔除动态物体，因此如何完整剔除视觉 SLAM 系统在动态场景中遇到的动态物体是亟待解决的问题。

针对移动机器人在动态场景中易出现有效深度值特征点数量过少和难以高效剔除动态特征点等问题，影响相机位姿估计精度。为此，提出一种基于 D-ORB（Depth guided Oriented FAST and Rotated BRIEF）特征点提取的动态视觉 SLAM 算法。通过设计一种 D-ORB 特征点提取方法，及时划分出有效区域，同时对有效区域进行关联和

ORB 特征点提取，提高有效深度值特征点数量。为进一步减少动态特征点对 SLAM 系统定位精度的影响，联合位姿估计和偏转视角剔除动态点，提高视觉 SLAM 系统在动态场景中的位姿精度。

目前传统的视觉 SLAM 算法在动态场景下易出现难以完整标记潜在动态物体和无法准确判断物体运动状态等问题，为此，提出一种基于实例分割的动态视觉 SLAM 算法。在物体分割环节，通过设计一种 ReT-encoder（Region Transformer encoder）模块，引导模型关注并提取图像局部特征信息和全局特征信息，同时设计一种混合权重特征金字塔网络（Mixed weight Feature Pyramid Network，M-FPN）模块，通过权重系数融合学习大物体特征和小物体特征，精确标记潜在动态物体区域。为降低动态物体对视觉 SLAM 系统的影响，联合位姿矩阵和分割掩码构建出物体投影偏差，判断物体运动状态并及时剔除动态物体，减少动态物体对 SLAM 系统的影响。

使用公开 TUM 数据集分别测试基于 D-ORB 特征点提取的动态视觉 SLAM 方法与基于实例分割的动态视觉 SLAM 方法，并与主流的 ORB-SLAM2 和 DynaSLAM 算法对比分析。对以上算法进行硬件平台的搭建和实验验证，实验结果表明本文所提算方法在动态场景中定位精度与构图效果与上述两种算法相比有一定优势，证明本算法的可行性。

关键词：同步定位与地图构建，实例分割，偏转视角，物体投影偏差

RESEARCH ON VISUAL SLAM ALGORITHM BASED ON IMPROVED TRANSFORMER INSTANCE SEGMENTATION IN DYNAMIC SCENES

ABSTRACT

Simultaneous Localization And Mapping (SLAM) means that the mobile robot obtains environmental information through the configured sensor, locates in an unknown environment, and builds a map at the same time. Laser SLAM and Visual SLAM(VSLAM) are two categories of SLAM algorithms. Since laser SLAM has certain disadvantages compared with visual SLAM in terms of price and obtaining feature information, SLAM technology using visual sensors has gradually become the mainstream research direction. At present, most visual SLAM algorithms are applicable to static scenes. If the motion state of objects in dynamic scenes cannot be accurately obtained and dynamic objects can be screened out, the system error of visual SLAM will be increased, and even the visual SLAM system cannot operate normally in dynamic scenes. At present, most visual SLAM algorithms incorporate the instance segmentation network to mark potential dynamic objects in the scene, but the traditional instance segmentation network has low segmentation accuracy and cannot completely segment potential dynamic objects, so it is difficult to completely eliminate dynamic objects. Therefore, how to completely

eliminate dynamic objects encountered by visual SLAM system in dynamic scenes is an urgent problem to be solved.

The number of effective depth value feature points is too small and it is difficult to efficiently remove dynamic feature points in the pose estimation of the system, which affects the accuracy of the camera pose estimation. Therefore, a dynamic VSLAM algorithm based on Depth guided Oriented FAST and Rotated BRIEF (D-ORB) feature point extraction method is proposed. By designing a D-ORB feature point extraction method, the effective region is divided in time, and the correlation of the effective region and the extraction of ORB feature points are carried out to increase the number of effective depth feature points. In order to further reduce the influence of dynamic feature points on the localization accuracy of SLAM system, the pose estimation and deflection angle are combined to eliminate dynamic points and improve the pose accuracy of visual SLAM system in dynamic scenes.

At present, the traditional visual SLAM algorithm easily cause problems such as difficult to completely label potential dynamic objects and can not accurately judge the motion state of objects in dynamic scenes. Therefore, a dynamic VSLAM algorithm based on instance segmentation is proposed. In object segmentation, a Region Transformer encoder (ReT-encoder) module is designed to guide the model to focus on and extract the local feature information and global feature information of the image. At

the same time, a Mixed weight Feature Pyramid Network (M-FPN) module is designed to learn large object features and small object features by weight coefficient fusion, and accurately label potential dynamic object regions. In order to further reduce the influence of dynamic feature points on the localization accuracy of SLAM system, the projection error of the object is constructed by combining pose matrix and segmentation mask, and the motion state of the object is judged and the dynamic object is deleted in time to reduce the influence of dynamic object on SLAM system.

The open TUM dataset was used to test the dynamic VSLAM method based on D-ORB feature point extraction and the dynamic VSLAM method based on instance segmentation, and compared with the mainstream ORB-SLAM2 and DynaSLAM algorithms. The hardware platform of the above algorithm is built and experimental verification is carried out. The experimental results show that the proposed algorithm has certain advantages compared with the above two algorithms in positioning accuracy and composition effect in dynamic scenes, which proves the feasibility of the proposed algorithm.

Qian Runbang(Control Science and Engineering)

Supervised by Professor Chen Mengyuan

KEYWORDS: Simultaneous Localization and Mapping, Instance Segmentation, Deflection angle, Projection error of the object

目 录

第 1 章 绪论

1.1 本文研究背景及意义

人工智能技术在理论和应用方面的不断探索和进步，使得移动机器人技术能够满足大部分需求并应用于实际场景中[1][2]。移动机器人具有自主性、灵活性、智能化等特点，它可以根据不同场合提供高效便捷、安全可靠的服务。在智能化的背景下，移动机器人的实际应用随着各行各业的迫切需求在不断扩大。例如，在工业自动化领域，移动机器人可以对精密仪器和汽车零部件进行精确的加工和组装，提高产品的一致性和准确性。在家庭服务领域，自动扫地机器人可以自主地清扫地面上的灰尘和垃圾，用户可以通过在线交互功能指导移动机器人监控家庭安全。移动机器人视觉 SLAM 技术的不断探索和实践，让移动机器人对未知环境感知的能力渐渐变强。近年以来，移动机器人的实际运行场景逐渐转变为复杂的动态场景。传统移动机器人技术缺乏应对动态场景的策略，难以在动态场景下保持高鲁棒性，因此动态场景下的移动机器人 SLAM 技术成为目前研究热点之一[3][4]。

移动机器人在运行的过程中，系统会获取传感器依据场景生成的数据，利用数据之间的约束关系计算出移动机器人的当前位置，从而实现移动机器人在陌生环境下的定位功能，同时构建出陌生环境的地图，为探索移动机器人更深层次的决策与应用提供基础[5][6]。移动机器人配置外部传感器类型的选择依据场景需求，然后再构建出相应的算法，从而实现高效率的系统定位和地图构建功能，常见的 SLAM 算法包括激光 SLAM 和视觉 SLAM[7]。

根据激光雷达数据表达形式的差异，将激光雷达主要分为二维和三维。二维激光 SLAM 技术相对成熟，代表算法有 Cartographer[8]算法。但由于二维激光 SLAM 只能获取移动机器人所处陌生环境中的二维平面数据，因此在获取数据信息方面存在一定的劣势。三维激光 SLAM 使用多线激光雷达[9]，可以获取环境的三维数据，因此能够表达更为丰富的空间信息，但在技术上不如二维激光 SLAM 技术的成熟，且价格较为昂贵。虽然激光 SLAM 在静态场景下有着高精度、高稳定性等优势，但在面对动态场景时其系统难以保持较高的定位精度。

视觉 SLAM 通过视觉传感器获取场景中的信息，最常用的视觉传感器是相机。根据相机功能差异，可将相机分为单目相机、双目相机和 RGB-D 相机。相机在价格、体积以及场景信息获取方面与激光发生器相比有一定的优势，因此在一些大规模场景中更适合相机获取场景信息。相机获取的图像帧输入到视觉 SLAM 系统后，通过算法的建模不仅能够获取图像中纹理特征细节，还能够检测环境中物体级别的语义信息，从而增强视觉 SLAM 系统的信息处理能力，因此，视觉 SLAM 成为多数研究者的主要研究方向[10]。

视觉 SLAM 通过相机记录场景信息，并以图像帧的形式输入到视觉 SLAM 系统，然后系统根据当前图像帧的特征信息与上个图像帧建立约束关系，最后计算得到位姿，从而实现系统定位功能。然而大多数传统视觉 SLAM 算法假设移动机器人所处环境是静止不变的，并基于该假设利用场景中的特征信息求解获取移动机器人当前位置。若将动态物体的特征信息加入位姿估计环节中，系统构建的轨迹精度和地图精度将下降，因此传统视觉 SLAM 算法由于缺乏物体运动判断策略将无法高效完成视觉 SLAM 系统任务[11]。近些年来研究者们通过构建几何约束关系来判断场景中物体的运动状态，从而获取动态特征信息。为进一步增加视觉 SLAM 系统在动态场景中运行的稳定性，通过算法的构建以获取场景中更为丰富的语义信息[12][13]。随着深度学习理论和应用的发展，使得实例分割网络在识别物体精度方面取得了极大的进步。实例分割网络对图像帧进行检测，标记出动态场景中出现的潜在动态物体，进一步增强了视觉 SLAM 系统检测并筛选出动态物体的能力[14][15]。目前大多数视觉 SLAM 算法引入了深度学习的方法来获取场景中更为丰富的语义信息，但也存在一定的问题：（1）移动机器人在使用 RGB-D 相机时，易出现因深度图像部分深度值缺失造成系统在位姿估计时有效深度值特征点数量过少和难以高效剔除动态特征点等问题，从而降低移动机器人位姿估计时系统的稳定性。（2）动态场景下实例分割网络难以完整标记潜在动态物体，尤其是环境中的大物体，若无法完整剔除动态物体，将对系统定位和建图功能有较大的影响。

综上所述，动态场景下标记动态物体，对视觉 SLAM 系统的位姿求解和地图构建有着较大的作用。本文针对移动机器人在动态场景中易出现因深度图像部分深度值缺失造成系统在进行位姿估计时有效深度值特征点数量过少以及难以高

效剔除动态特征点等问题，动态场景下难以完整标记潜在动态物体和无法准确剔除动态物体等问题，提出一种动态场景下视觉 SLAM 优化算法。提出的算法有效提高了有效深度值特征点数量，同时剔除动态点，在动态场景中完整标记潜在动态物体，同时剔除动态物体，本文算法有效提高视觉 SLAM 系统在动态场景中的定位精度。

1.2 国内外研究现状

1.2.1 视觉 SLAM 研究现状

视觉 SLAM 技术的应用能够满足大部分实际场景需求，为人们的生活提供了一定的便利。与激光雷达相比，相机在价格和信息量表达方面有着一定的优势[16]。视觉 SLAM 通过配置的相机获取场景图像帧，同时将图像帧与图像帧之间的特征进行数据关联获取位姿，从而得到移动机器人在环境中的位置。根据帧与帧之间获取位姿方式的不同，将 SLAM 系统里面的视觉里程计方法分为直接法和特征点法。

直接法主要依据图像帧上的像素灰度信息建立光度误差方程，通过最小化光度误差来估计相机的运动，从而计算出位姿。在 Newcombe 等人[17]提出的算法中，通过直接法构建最小化光度误差模型获取相机位姿，提高了算法在弱纹理场景中位姿跟踪鲁棒性，但该算法地图构建方法的计算量较大，需要采用 GPU 加速。Engel 等人[18]在 2014 年提出的 LSD-SLAM 算法中使用直接法进行帧间位姿求解，但是该算法在位姿求解过程中对光照变化较为敏感且在大角度弯道运动中难以保持跟踪的稳定性。在 2017 年，Engel 等人[19]提出了一种高效率的稀疏直接法 DSO，该算法首先通过位姿约束将帧间的像素值差值作为误差函数，然后构建最小化误差函数求解出最优位姿，为减少环境光照对算法的影响，DSO 加入了光度标定，进一步提高系统运行的稳定性。在 2023 年，贾嫣晗等人[20]提出一种双目直接法视觉 SLAM 算法，该算法在使用直接法的同时引入双目静态残差约束策略来提升算法帧间跟踪精度，在室内外表现出一定的鲁棒性。

特征点法的理论较为成熟，它在光照变化条件下与直接法相比具有更强的稳定性，因此在视觉 SLAM 算法的视觉里程计中主要使用特征点法记录图像特征。

Davision 等人[21]在 2007 年提出了 MonoSLAM 算法，该算法使用单目相机获取场景中的图像帧，并使用特征点法提取特征点，在位姿优化阶段，采用了卡尔曼滤波器方法，但是滤波器方法容易受环境中噪声的影响，从而降低系统运行的稳定性。同年 Klein 等人[22]设计出 PTAM 算法，该算法提出了相机位姿的非线性优化的策略，并引入了关键帧筛选机制，虽然该方法减少了数据冗余，但是该方法容易出现帧与帧之间跟踪失败的情况。在 2015 年，Mur-Artal 等人[23]提出了 ORB-SLAM，该算法主要包括跟踪、局部建图和闭环检测三个线程，在特征点提取环节采用了 ORB 法进行特征点提取，该特征点提取的方法具有较高的鲁棒性，有效提高帧间特征点跟踪的准确性，同时引入回环检测模块，进一步提高 ORB-SLAM 环境感知能力。同年，付梦印等人[24]提出的算法中将经过特征点法提取的特征点输入高斯混合模型中，结合彩色图像和特征点位置获取颜色、坐标和协方差矩阵，为降低因特征点过少对 ICP（Iterative Closest Point）算法精度的影响，该算法引入粒子采样步骤，有效提高了定位精度。在 2017 年，Mur-Artal 等人[25]在 ORB-SLAM 的基础上对传感器的类型进行细分并提出 ORB-SLAM2，该系统能够高效处理单目、双目和 RGB-D 相机生成的图像，并在系统判定为回环后及时对帧间约束关系进行优化，提高了系统的稳定性。在 2018 年，张涛等人[26]提出一种双目视觉里程计，该算法通过改进的 RANSAC（RANdom SAmple Consensus）算法选取经过特征点法提取的优质特征点，降低特征匹配错误率并提高匹配效率，同时提出一种双向重投影的位姿估计算法，有效降低了视觉里程计累积误差，提升了系统的位姿估计精度。在 2020 年，Carlos 等人[27]提出一种多地图复用的 SLAM 算法，称为 ORB-SLAM3 算法，该算法在室内外与 ORB-SLAM2 算法相比更具有稳定性。在 2023 年，张洪等人[28]在 ORB-SLAM2 算法的基础上引入旋转和平移量策略作为是否创建关键帧的依据，该方法有效降低了系统的误差。

1.2.2 动态场景下视觉 SLAM 研究现状

传统视觉 SLAM 利用图像帧间特征的约束计算得到系统当前位置，从而实现定位功能。当场景中存在动态物体时，由于系统缺乏对动态物体的约束方法，因此会将动态物体上的特征点引入到位姿估计环节中，从而降低轨迹精度。在相

对复杂的场景中会不可避免地存在动态物体，此时需要通过建模合适的算法准确筛选出动态物体来提高视觉 SLAM 系统在动态场景中的稳定性。

基于几何方法主要通过构建图像帧之间的数学几何关系，再结合场景情况判断特征点是否满足约束条件，从而得到动态点和静态点。Zou 等人[29]提出的 CoSLAM 算法首先计算出跟踪特征点重投影距离，再将该距离与阈值进行比较筛选出动态特征点，引入的优化函数使得系统能够更准确获取特征点运动状态，更高效地获取动态特征点，从而提高了算法的鲁棒性，CoSLAM 系统框架如图 1-1 所示。Chen 等人[30]提出的算法中采用 RANSAC 五点法获取相机位姿变换矩阵，并通过相机位姿变换矩阵将帧间特征点进行数据关联获取特征点经过转换后的位置，在跟踪过程中该算法利用了直接法跟踪帧间特征点获取投影位置，在运动状态判断策略中将经过转换的特征点位置与真实特征点位置进行对比，从而判断特征点运动状态，但是该特征点运动判断方法易受噪声和光线因素的干扰。Dai 等人[31]通过帧间特征点的关联来实时地区分出特征地图点中的动态点与静态点，进而有效避免了动态物体对系统的影响。

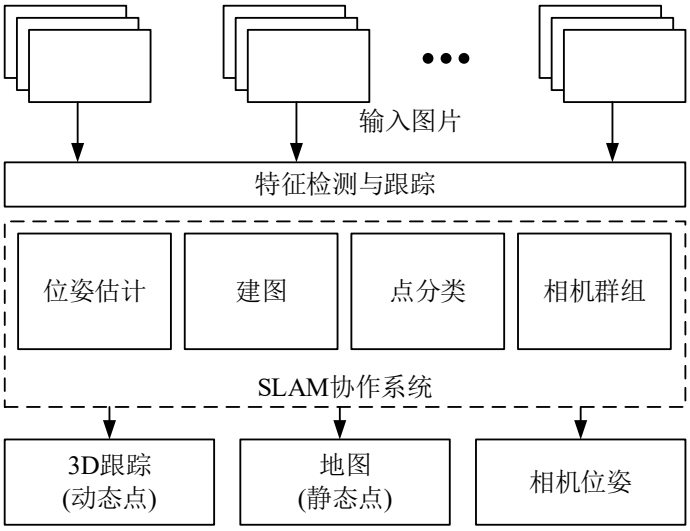


图 1-1 CoSLAM 系统框架

随着基于深度学习的图像分割算法的不断发展与进步，图像分割算法为降低动态场景中动态物体对视觉 SLAM 系统的影响提供了新思路。基于深度学习的图像分割算法能够及时推理图像帧中存在的语义信息，依据语义信息获取潜在动态物体区域，再结合区域内特征点状态分布判断是否为动态物体，从而提高系统在动态场景下的鲁棒性。Kaneko 等人[32]引入了图像分割模型处理图像帧，并根据语义信息对应的掩码标记区域，直接剔除潜在运动物体表面上特征点，从而降

低动态特征点对系统的影响，但该算法只针对某些特定语义信息引导剔除物体，由于缺乏运动判断策略，因此该算法将难以准确剔除场景中的动态物体。Zhong等人[33]提出了 Detect-SLAM 算法，将关键帧输入到 SSD[34]网络中并得到各个物体分割掩码，从而获取物体所在区域，为获取物体运动状态，通过概率传播策略获取特征点运动状态并剔除动态点，从而提高系统定位精度。Xiao 等人[35]在使用 SSD 算法的同时加入了漏检测的方法，在特征点运动状态策略中，对 SSD 算法输出掩码标记区域的特征点进行运动状态判断，及时筛选出动态点，但该算法分割精度较低，难以完整分割物体。在 2018 年，Yu 等人[36]提出 DS-SLAM 算法，该算法利用语义分割网络 SegNet[37]获取图像帧中潜在动态物体所在区域，为获取标记物体的运动状态，通过极线约束判断标记区域特征点运动状态，及时得到动态物体所在区域，但是若场景中出现大物体，该算法的鲁棒性将下降。Bescos等人[38]提出了 DynaSLAM 系统，该算法在 ORB-SLAM2 基础上引入了图像分割模块和特征点运动判断模块，图像分割模块采用 Mask RCNN[39]实例分割网络获取场景中的潜在动态物体，并直接剔除场景中的人，然后将除人之外的特征信息加入线程中进行粗略跟踪，在特征点运动判断环节，利用帧间特征点构造向量，根据向量夹角判断特征点运动状态，该算法降低了动态物体对系统的影响，但实例分割网络分割精度较低，难以完整剔除动态物体，DynaSLAM 使用 RGB-D 相机的系统框架如图 1-2 所示。姚二亮等人[40]提出的算法中，首先使用 YOLOv3[41]网络获取场景中的语义标签，然后结合基于图像中边缘的距离变换误差与光度误差的一致性进一步细分出动静态区域，该方法在构建地图过程中减少了动态特征的加入，降低了动态物体对系统稳定运行的影响。在 2022 年，Wu 等人[42]提出了 Yolo-SLAM 算法，利用设计的 Darknet19-YOLOv3 网络获取图像帧中各个潜在动态物体的标记区域，同时提出一种 depth-RANSAC 方法判断标记区域特征点的运动状态，有效筛选出大部分动态点，但该方法的分割网络难以完整标记潜在动态物体区域。

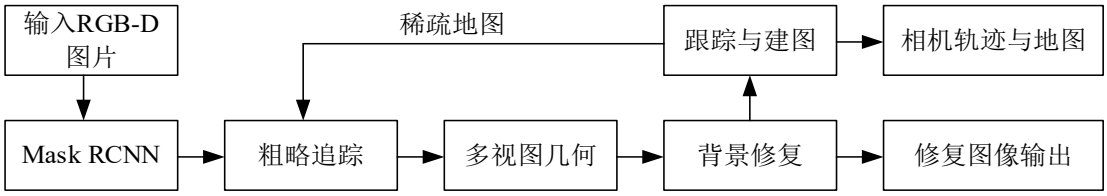


图 1-2 DynaSLAM 算法中使用 RGB-D 相机的系统框架

1.3 研究内容与章节安排

本文主要研究移动机器人在动态场景中运行的情况下，系统定位精度下降的问题，对方法进行改进并提出一种动态场景下基于改进 Transformer 实例分割的视觉 SLAM 算法。本文对视觉 SLAM 系统做了如下工作：（1）针对移动机器人在动态场景中易出现有效深度值特征点数量过少和难以高效剔除动态特征点的问题，设计一种基于 D-ORB 特征点提取的动态视觉 SLAM 算法。通过设计一种 D-ORB 特征点提取方法，提高有效深度值特征点数量。同时联合位姿估计和偏转误差剔除动态点，提高视觉 SLAM 系统在动态场景中的位姿精度。（2）目前传统的视觉 SLAM 算法在动态场景下易出现难以完整标记潜在动态物体和无法准确判断物体运动状态等问题，基于此，提出一种基于实例分割的动态视觉 SLAM 算法。通过设计一种实例分割网络，提高潜在动态物体分割精度。通过设计一种物体投影偏差策略，及时降低动态物体对 SLAM 系统的影响。论文技术路线如图 1-3 所示。

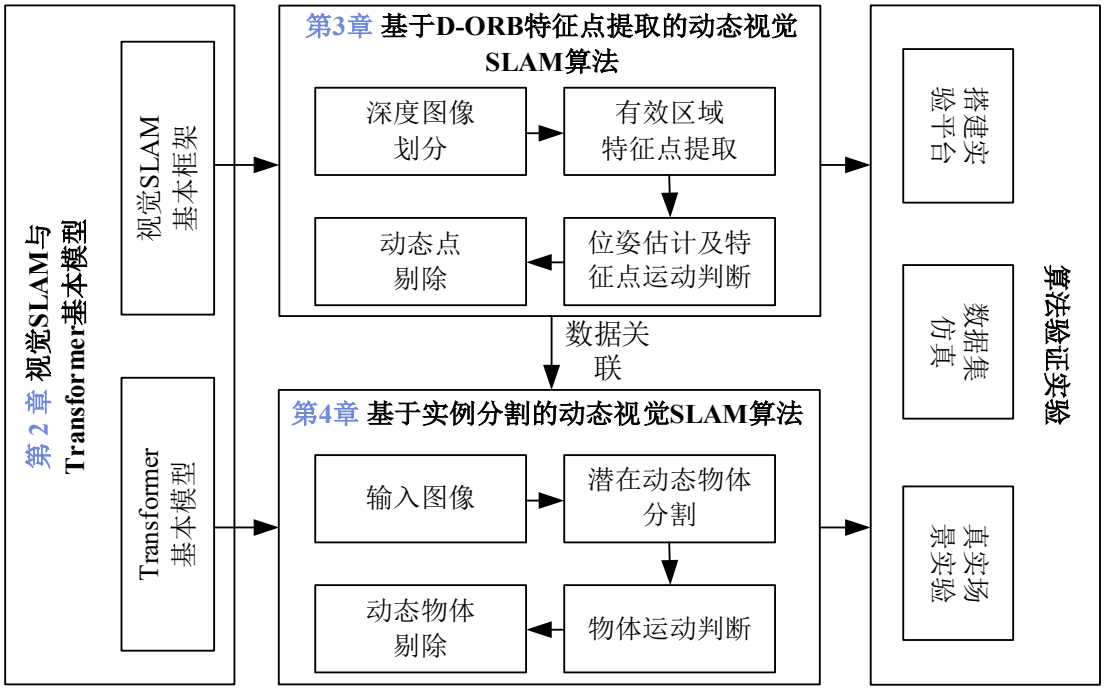


图 1-3 论文技术路线

各章节的主要内容如下：

第 1 章：介绍基于视觉 SLAM 技术的重要性以及其研究背景、意义和现状。并对目前视觉 SLAM 技术以及动态场景下视觉 SLAM 技术的研究现状进行了介

绍。

第 2 章：介绍视觉 SLAM 与 Transformer 基本框架。主要介绍了视觉 SLAM 系统特征点提取与匹配、相机位姿求解、后端优化、闭环检测与建图五个部分。根据动态场景需要，探索 Transformer 模型，介绍了一种基于 Transformer 图像分割网络，在动态场景下将基于 Transformer 实例分割的视觉 SLAM 算法与传统视觉 SLAM 算法 ORB-SLAM2 进行静态特征点分布对比。

第 3 章：基于 D-ORB 特征点提取的动态视觉 SLAM 算法。针对移动机器人在动态场景中易出现有效深度值特征点数量过少和难以高效剔除动态特征点等问题，影响相机位姿估计精度。为此，提出一种基于 D-ORB 特征点提取的动态视觉 SLAM 算法。通过设计一种 D-ORB 特征点提取方法，及时划分出有效区域，同时对有效区域进行关联和 ORB 特征点提取，提高有效深度值特征点数量。为进一步减少动态特征点对 SLAM 系统定位精度的影响，联合位姿估计和偏转视角剔除动态点，提高视觉 SLAM 系统在动态场景中的位姿精度。

第 4 章：基于实例分割的动态视觉 SLAM 算法。目前传统的视觉 SLAM 算法在动态场景下易出现难以完整标记潜在动态物体和无法准确判断物体运动状态等问题，为此，提出一种基于实例分割的动态视觉 SLAM 算法。在潜在动态物体实例分割环节，通过设计一种 ReT-encoder 模块，引导模型关注并提取图像局部特征信息和全局特征信息，同时设计一种混合权重 FPN 模块，通过权重系数融合学习大物体特征和小物体特征，精确标记潜在动态物体区域。在潜在动态物体运动状态判断环节，联合第 3 章算法剔除动态点后估计的位姿与本章实例分割网络输出的掩码构建物体投影偏差模型获取并及时剔除动态物体，降低动态物体对视觉 SLAM 系统的影响。

第 5 章：总结与展望。总结了本文研究工作及解决方案，并总结了当前方法的优缺点，展望了下一步研究计划。

第 2 章 视觉 SLAM 与 Transformer 基本模型

本章首先对视觉 SLAM 中的模块展开了研究，然后对 Transformer 基本框架进行简单介绍，并简要分析一种基于 Transformer 实例分割网络。最后依据动态场景的需要建立模型用于提高视觉 SLAM 系统运行的稳定性，在动态场景下将基于 Transformer 实例分割的视觉 SLAM 算法与传统视觉 SLAM 算法 ORB-SLAM2 进行实验对比，为后续动态场景下视觉 SLAM 的研究奠定基础。

2.1 视觉 SLAM 基本框架

视觉 SLAM 主要由前端视觉里程计、回环检测、后端优化和建图组成。视觉 SLAM 中的前端视觉里程计通过传感器测算得到图像帧以及数据信息，并根据需求建立图像帧间的约束关系，通过对约束关系的计算与优化获取系统当前位置，同时利用计算后得到的数据信息完成所处环境中局部地图的构建。回环检测将当前相机捕捉的图像帧与系统中存储的图像帧进行对比，根据相似度判断是否经过相同场景，若判断为回环则将计算得到的位姿输入到后端优化模块中进行优化。视觉 SLAM 系统在计算位姿的过程中难以避免会出现误差，此时需要将经过前端视觉里程计得到的位姿输入到后端优化模块，并利用回环中的约束条件对位姿进行矫正。建图是根据视觉 SLAM 系统中各个图像帧的特征，结合系统存储的数据信息构建出真实场景地图。本节对前端视觉里程计中的特征点提取与匹配和前端视觉里程计中的相机位姿求解进行介绍，对后端优化、闭环检测与建图进行简单阐述。

2.1.1 特征点提取与匹配

特征点是图像中具有独特性质的局部像素区域，它可以帮助识别和匹配不同图像之间的相同物体或场景。常见的特征点提取算法有 Harris 角点检测、SIFT、SURF 和 ORB（Oriented FAST and Rotated BRIEF）。Harris 角点检测是一种基于图像灰度变化的角点检测方法，它通过计算图像窗口在各个方向上移动后的灰度变化来识别角点。虽然 Harris 角点检测对旋转和亮度变化具有一定的不变性，但对尺度变化敏感。SIFT 方法首先检测出图像中的关键点，然后计算出这些关键

点的方向、尺度和位置信息。在旋转、尺度缩放、亮度变化的情况下 SIFT 也能保持其特征的不变性。SURF 是基于 SIFT 改进的方法，它在保持 SIFT 特征旋转、尺度不变性和稳定性的同时，通过使用积分图像来提高计算速度。ORB 是一种快速的特征点检测和描述符提取算法，它结合了 FAST 角点检测和 BRIEF 描述子的优点，实现了旋转不变性，并且在性能上有显著的提升，其中 BRIEF 描述子是由 0 和 1 组成的向量，主要用于特征匹配。根据场景需要，本文采用 ORB 方法提取特征点。

FAST 算法检测图像中角点的具体步骤为:（1）在图像中寻找坐标 $\boldsymbol{p}=(u,v)$，并获取该位置的像素值 $\mathrm{I}(\boldsymbol{p})$；（2）以位置 $\boldsymbol{p}$ 为圆心画圆，得到该圆周上 16 个像素值，如图 2-1 所示；（3）若该圆周上有连续 N 个像素值都小于 $\mathrm{I}(\boldsymbol{p})-t_p$ 或者都大于 $\mathrm{I}(\boldsymbol{p})+t_p$，则将 $\boldsymbol{p}$ 位置认为是特征点的位置；其中 t_p 是设定的阈值，$\mathrm{I}(\bullet)$ 表示获取像素值函数。

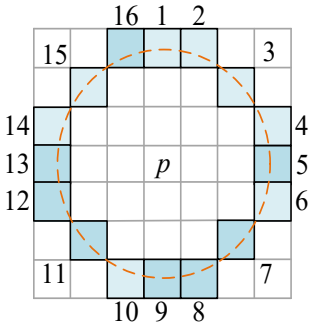


图 2-1 FAST 角点检测示意图

为获取帧与帧之间的位姿矩阵，在获取每个图像帧上的特征点集合后会进行特征匹配，从而得到匹配对，然后根据匹配对对应的约束关系进行估计得到位姿矩阵。特征匹配最简单的方法为暴力匹配法，在该方法中，对于当前图像帧 Γ_k 中的特征点集合对应的描述子集合 $\left\{\boldsymbol{x}_1^k,\boldsymbol{x}_2^k,...,\boldsymbol{x}_m^k\right\}$，假设下一帧 Γ_{k+1} 中的特征点集合对应的描述子集合为 $\left\{\boldsymbol{x}_1^{k+1},\boldsymbol{x}_2^{k+1},...,\boldsymbol{x}_l^{k+1}\right\}$，通过计算帧 Γ_k 中某点描述子 $\boldsymbol{x}_i^k(1\le i\le m)$ 与帧 Γ_{k+1} 中描述子集合中 $\left\{\boldsymbol{x}_1^{k+1},\boldsymbol{x}_2^{k+1},...,\boldsymbol{x}_l^{k+1}\right\}$ 所有元素的汉明距离 (Hamming distance)，其中汉明距离最小的则为匹配点。

在特征匹配的过程中，难以避免会出现误匹配[43]的情况，若不采取方法及时剔除错误匹配的匹配对，将会使得 SLAM 系统的误差变大，影响定位精度。常

见的剔除错误匹配方法为 RANSAC，该方法通过迭代及时排除不符合模型的数据，因此将特征匹配后的数据应用于该方法后能够减少错误数据参与位姿估计，提高 SLAM 系统运行的鲁棒性。

2.1.2 相机位姿求解

在获取帧间匹配对后，视觉 SLAM 系统会将匹配对中的特征点对进行数据关联获取位姿。在位姿求解过程中，视觉 SLAM 系统需要跟据传感器输出数据类别选择相应求解方法。例如相机可分为单目相机、双目相机和 RGB-D 相机，单目相机通常使用 2D-2D 对极几何[44]的方法对帧间的位姿进行求解，双目相机使用 3D-2D 点 PnP[45]方式估计求解帧与帧之间的位姿，ICP[46]主要针对使用 RGB-D 相机的情况，利用深度图像的深度值对特征点进行投影，然后对匹配对映射成的 3D 点对进行数据关联，从而获取帧间位姿。

根据场景需要，本文选取 RGB-D 相机对应的位姿求解方法。RGB-D 相机采集环境信息后会输出彩色图像和深度图像，其中的深度图像是帧间特征点转化为世界坐标点的重要依据。

在 ICP 算法中，假设帧 Γ_k 中特征点集合 $\left\{\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, ..., \boldsymbol{p}_m^k\right\}$ 与帧 Γ_{k+1} 中特征点集合 $\left\{\boldsymbol{p}_1^{k+1}, \boldsymbol{p}_2^{k+1}, ..., \boldsymbol{p}_m^{k+1}\right\}$ 相匹配。对于匹配对其中的特征点 $\boldsymbol{p}_i^k$ 和 $\boldsymbol{p}_i^{k+1}$，将该匹配对投影到世界坐标系中，计算过程如公式(2-1)所示：

$$\begin{cases} \boldsymbol{P}_i^k = d_i^k \Phi(\boldsymbol{p}_i^k) \\ \boldsymbol{P}_i^{k+1} = d_i^{k+1} \Phi(\boldsymbol{p}_i^{k+1}) \end{cases} \tag{2-1}$$

其中，$0 \le k \le m-1$，$\Phi(\bullet)$ 为投影函数，d_i^k 为第 k 帧中第 i 个匹配对中点 $\boldsymbol{p}_i^k$ 的深度值，d_i^{k+1} 为第 k+1 帧中第 i 个匹配点对中点 $\boldsymbol{p}_i^{k+1}$ 的深度值。

将得到的 3D 点集合 $\{\boldsymbol{P}_1^k, \boldsymbol{P}_2^k, ..., \boldsymbol{P}_m^k\}$ 和 $\{\boldsymbol{P}_1^{k+1}, \boldsymbol{P}_2^{k+1}, ..., \boldsymbol{P}_m^{k+1}\}$ 进行数据关联。对于集合中的点 $\boldsymbol{P}_i^k$ 和 $\boldsymbol{P}_i^{k+1}$，建立位姿约束，计算方式如公式(2-2)所示：

$$\boldsymbol{P}_i^{k+1} = \boldsymbol{R}\boldsymbol{P}_i^k + \boldsymbol{t} \tag{2-2}$$

其中，$\boldsymbol{R}$ 为相机相对旋转矩阵，$\boldsymbol{t}$ 为平移矩阵。

为得到更准确的位姿矩阵，需要将所有匹配对构成的约束关联起来，计算方

式如公式(2-3)所示：

$$\{\hat{\boldsymbol{R}},\hat{\boldsymbol{t}}\}=\min_{\boldsymbol{R},\boldsymbol{t}}\frac{1}{2}\sum_{i=1}^{m}\left\|\boldsymbol{P}_i^{k+1}-(\boldsymbol{R}\boldsymbol{P}_i^{k}+\boldsymbol{t})\right\|_2^2 \tag{2-3}$$

该方法利用多个匹配点进行位姿求解提高了相机位姿估计精度，但也存在一定的缺点。视觉 SLAM 系统对图像帧进行特征点提取的过程中会进行关键点的提取和描述子的计算，该过程相对耗时；由于特征点法的局限性，因此难以记录除特征点以外的信息；特征点法对于纹理不是很明显的图像帧容易造成特征点提取失败，从而造成跟踪失败。

直接法根据像素的亮度信息估计相机运动，无需计算描述子。因此与特征点法相比不仅减少了 SLAM 系统特征计算的时间，而且提高了动态低纹理场景下位姿计算的稳定性。

在直接法中，考虑一个世界坐标点 $\boldsymbol{P}_i^{w}$，点 $\boldsymbol{q}_i^{k}$ 和点 $\boldsymbol{q}_i^{k+1}$ 约束关系如公式(2-4)所示：

$$\begin{cases}\boldsymbol{q}_i^{k}=\dfrac{1}{Z}\boldsymbol{K}\boldsymbol{P}_i^{w}\\ \boldsymbol{q}_i^{k+1}=\dfrac{1}{Z_{k+1,k}}\boldsymbol{K}(\boldsymbol{R}\boldsymbol{P}_i^{w}+\boldsymbol{t})\end{cases} \tag{2-4}$$

其中，Z 为点 $\boldsymbol{q}_i^{k}$ 的深度值，$Z_{k+1,k}$ 为点 $\boldsymbol{P}_i^{w}$ 在第 k+1 帧中对应的深度值。

为得到位姿，需要找到与点 $\boldsymbol{q}_i^{k}$ 更相似的点 $\boldsymbol{q}_i^{k+1}$。通过构建光度误差 e_i 来描述 $\boldsymbol{P}_i^{w}$ 的在第 k 帧和第 k+1 帧中两个像素差值，如式(2-5)所示：

$$e_i=\mathrm{I}(\boldsymbol{q}_i^{k})-\mathrm{I}(\boldsymbol{q}_i^{k+1}) \tag{2-5}$$

其中，I(•) 为获取图像相应位置坐标像素值函数。

对于 m^{ref} 个 e_i 方程，则相机位姿求解问题如公式(2-6)所示：

$$\{\hat{\boldsymbol{R}},\hat{\boldsymbol{t}}\}=\frac{1}{2}\min_{\boldsymbol{R},\boldsymbol{t}}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} \tag{2-6}$$

其中，$\boldsymbol{e}=(e_1,e_2,...,e_{m^{\mathrm{ref}}})^{\mathrm{T}}$。

直接法求解位姿的速度比特征点法更快，且在低纹理场景中比特征点法的鲁棒性高，但该方法也存在一定缺点。直接法是基于灰度不变的条件，只有在相机位姿运动较小时直接法才能稳定生效。

2.1.3 后端优化

视觉 SLAM 系统在进行位姿求解过程中易出现因图像帧间特征点匹配错误造成位姿估计精度降低的问题，从而降低视觉 SLAM 系统定位精度。随着系统运行时间的增加，位姿误差会进一步累积。后端优化通过序列中关联的图像帧获取帧间关联数据，利用数据间的约束方程降低位姿估计环节中存在的误差。后端优化的方法主要有滤波器[47]方法和图优化(Graph Optimization)方法。

基于滤波器的后端优化方法主要通过贝叶斯滤波模型对状态量进行不断的迭代与更新，它将 SLAM 问题建模成运动方程和相应的观测方程，并将该两个方程融入到滤波器中进行状态估计，进而对位姿进行优化。然而，该优化方法的存储量和计算量会随着状态量的增加呈平方趋势增长。在特征点超过一定数量时，该滤波方法的效率会显著下降，因此基于滤波器的方法难以高效处理大型场景的数据。

图优化方法是视觉 SLAM 系统常见的优化方法，它通过构建一个由节点和边组成的图来表示整个系统的状态，并对其进行优化。常见的图优化方法为 g2o[48]，它主要采用非线性方法优化位姿。在图优化方法运行阶段，以机器人位姿作为节点，边为约束构建出机器人运动和观测约束模型。通过最小化图中节点和边之间的误差，计算得到优化后的位姿。图优化通过联合多帧图像信息和空间点进行位姿优化提高了系统的运行效率。

2.1.4 回环检测

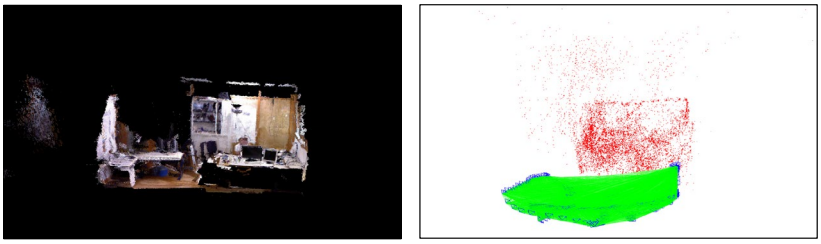
虽然视觉 SLAM 系统采用优化方法降低位姿计算的误差，但该方法难以完全消除累积误差。回环检测模块能够存储比相邻帧更久远的图像帧，若在相机运动过程中检测到相机之前出现的场景，则判断为产生回环[49]。若回环存在则建立与当前帧位姿关联的约束，后端对该约束关系进行优化，从而构建出精度更高的全局一致性地图。

为检测相机是否回到历史场景，回环检测采用词袋模型(Bag-of-Words，BOW)计算当前帧与存储在图像帧库中每帧的相似度，根据相似度判断是否发生回环。为保证回环检测的鲁棒性，系统会根据连续帧图像与当前帧计算出的相似度，若每个相似度都大于设定阈值则判定为回环。回环后将图像帧输入到后端，后端会

根据帧间约束对位姿进行优化，进而降低轨迹的累积误差。

2.1.5 建图

系统执行完运动估计、后端优化与闭环检测三个模块后会得到较为准确的3D 地图点和位姿数据，建图则是利用这些存储的数据以及之间存在的约束关系来建立环境地图[50]。定位、导航、避障、重建和交互是视觉 SLAM 系统建立地图主要作用。定位功能是移动机器人将当前场景建立的地图与系统存储的地图进行比较，根据是否经过相同的场景完成机器人的定位。导航功能为机器人根据场景路线建立地图，同时根据需求规划最优通行路径。避障功能是机器人会实时检测动态障碍物，并根据系统反馈及时避开障碍物所在位置，从而能够让机器人正常行驶。重建功能是机器人利用传感器传入的数据，建立所处环境的地图。交互功能是通过增加的高层次语义信息能让系统更高效完成任务，使系统对环境信息的理解更加准确，从而能高效完成多层次任务。半稠密点云地图和稀疏点云地图效果图如图 2-2 所示。



(a) 半稠密点云地图　　(b) 稀疏点云地图

图 2-2 SLAM 系统建图效果图

2.2 Transformer 基本模型

Transformer 是一种基于注意力机制的模型，自注意力（Self-Attention）机制能有效捕捉输入序列中的长距离依赖关系[51][52]。Transformer 在处理长序列方面与卷积神经网络（Convolutional Neural Network，CNN）相比具有更好的性能，尤其在物体分割方面，Transformer 模型与 CNN 相比能够保留更多的空间信息，因此能够更加高效地区分物体特征细节。Transformer 模型与 CNN 相比能够从不同的数据中学习更多信息，从而在图像分割任务中提高分割精度。本节将从 Transformer 基本框架出发，介绍一种基于 Transformer 模型的分割网络，同时比

较剔除动态物体后静态点分布效果。

2.2.1 Transformer 结构

Transformer 框架由编码层和解码层组成，如图 2-3 所示。编码层主要由多头自注意力(Multi-Self Attention，MSA)机制层和全连接前馈(Feed Forward)网络层组成，并使用了残差连接和层归一化的处理方法。解码层不仅拥有编码层中的多头自注意 MSA 机制层和全连接前馈(Feed Forward)网络层，而且还引入了掩码多头注意力层(Masked Multi-Head Attention)。掩码多头注意力层不仅能够增强了模型的注意力表达能力，而且能够避免在预测时依赖未来的信息，让模型预测更加合理。与编码层一样，解码层也使用了残差连接和层归一化的处理方法。由于自注意力机制本身不包含序列元素的位置信息，Transformer 通过位置编码向模型提供每个序列元素的相对或绝对位置信息。将 N 个编码层串联在一起，提高了模型学习更深层次信息的能力，从而更准确表达序列中的复杂信息。将 N 个解码层串联在一起，提高了模型提取更深层次信息能力，从而更加高效完成序列信息提取任务。因此，将基于 Transformer 模型用于图像目标分类能够更准确表达动态场景下潜在动态物体所在区域。

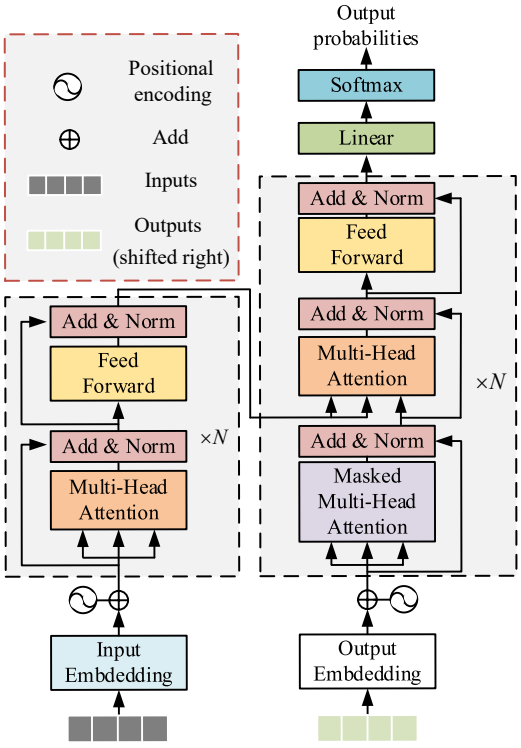


图 2-3 Transformer 基本框架图

2.2.2 物体分割算法

基于传统卷积神经网络的深度学习方法随着时间的发展取得一定的进步，但在物体分割方面与基于 Transformer 实例分割网络相比存在一定的劣势，这主要表现在以下方面。首先 Transformer 网络对中远距离特征信息有着较强的学习能力，它能够高效捕捉到图像各个元素中的全局依赖关系，进而能更加准确地学习图像中的物体特征。其次，Transformer 网络模型中的多头注意力机制可以增加图像特征表达的丰富度，通过适当叠加 Transformer 模型数量可提取图像更高层特征信息，并在自注意力机制不丧失空间信息的情况下能够对特征进行长距离的传播。最后，基于 Transformer 网络与卷积神经网络相比能够更高效进行并行运算，该方式提高了模型的训练速度。因此基于 Transformer 网络相较于传统卷积网络在图像目标检测与图像物体实例分割精度方面有较大优势。Swin Transformer[53] 是一种基于 Transformer 模型的图像分割网络，该网络的基本结构如图 2-4 所示。

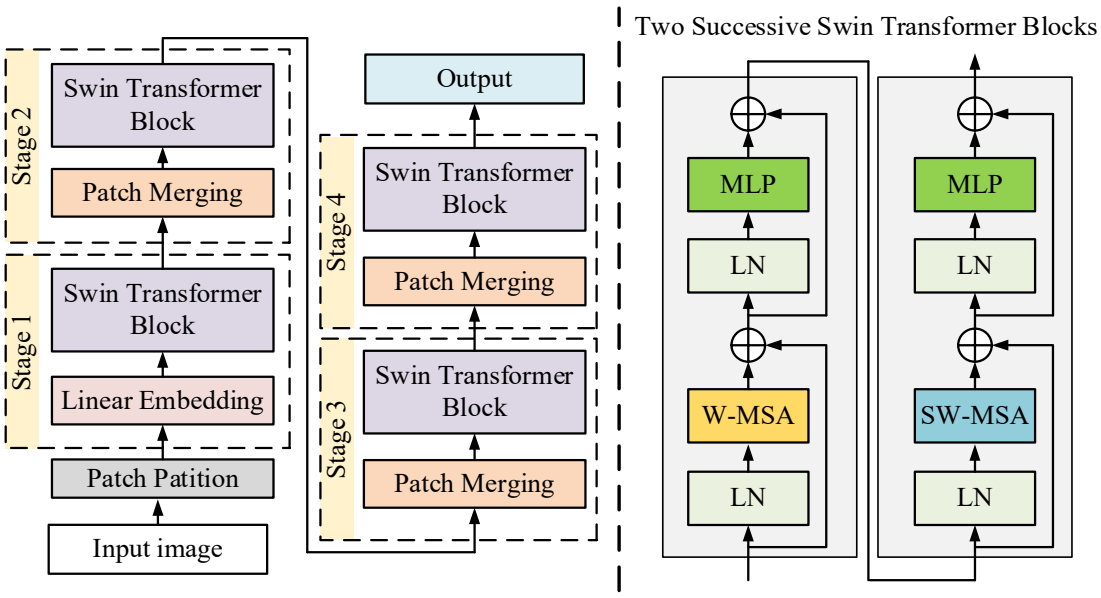


图 2-4 Swin Transformer 框架示意图

Swin Transformer 模型改进注意力机制如图 2-5 所示。在层 l 中，采用了规则注意力窗口(Window MSA，W-MSA)划分特征图的设计方案，模型在每个窗口使用自注意机制。在层 l+1 中，注意力窗口采用了可变式注意力窗口(Shifted Window MSA，SW-MSA)，该注意力窗口边界跨越了层 l 中注意力窗口的边界，从而为特征图之间建立了联系。

假设特征图大小为 $h \times w$，特征图通道数为 C，则 Transformer 中多头注意力

复杂度计算过程如公式(2-7)所示：

$$\Omega(\mathrm{MSA}) = 4hwC^2 + 2(hw)^2C \tag{2-7}$$

W-MSA 模块在 MSA 模块的基础上引入了注意力窗口，在处理上述同样大小的特征图时，假设注意力窗口尺寸大小为 $M \times M$，则 W-MSA 模块的复杂度计算方式如公式(2-8)所示：

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC \tag{2-8}$$

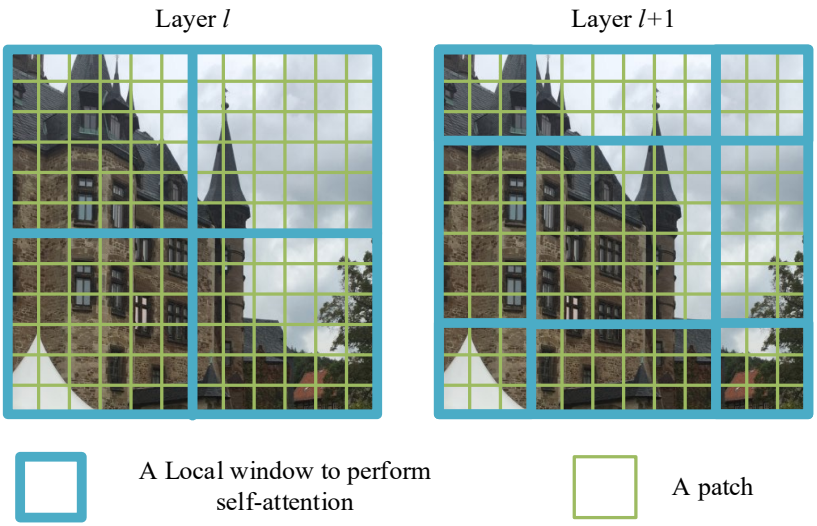


图 2-5 改进注意力窗口机制原理图

由公式(2-7)和公式(2-8)可知，MSA 复杂度与特征图大小成平方关系，W-MSA 复杂度与特征图大小成线性关系。当特征图尺寸增长到一定程度时，MSA 模块的复杂度增加的更快，因此 W-MSA 在处理大尺寸图片时比 MSA 更具有优势。Swin Transformer 算法通过变化注意力窗口引导多头自注意机制的计算，同时利用跨窗口策略建立联系，从而带来更高的效率，这为本文基于 Transformer 实例分割网络的设计提供了思路。

2.2.3 物体标记与动态物体剔除

传统视觉 SLAM 系统主要借助环境中静止不变的信息实现定位功能，若环境中动态物体的特征信息参与到定位线程中，将对视觉 SLAM 系统造成较大的系统误差，进而降低视觉 SLAM 系统的稳定性。因此，准确识别环境中的动态物体并剔除动态物体是视觉 SLAM 系统目前的主要任务。基于 Transformer 的实例分割网络与卷积网络分割效果相比有着一定的优势，这得益于 Transformer 的

注意力机制和相对位置编码方法。实例分割效果图如图 2-6 所示。

图 2-6 实例分割效果图

为减小动态物体对视觉 SLAM 系统的影响，需要对标记的潜在动态物体进行运动状态分析。首先对每个标记区域中的特征点进行运动判断，然后根据特征点运动状态分布剔除动态物体。剔除动态物体效果图如图 2-7 所示，其中剔除区域设置为白色。由图可知，是否能完整剔除动态物体取决于实例分割网络的分割精度。

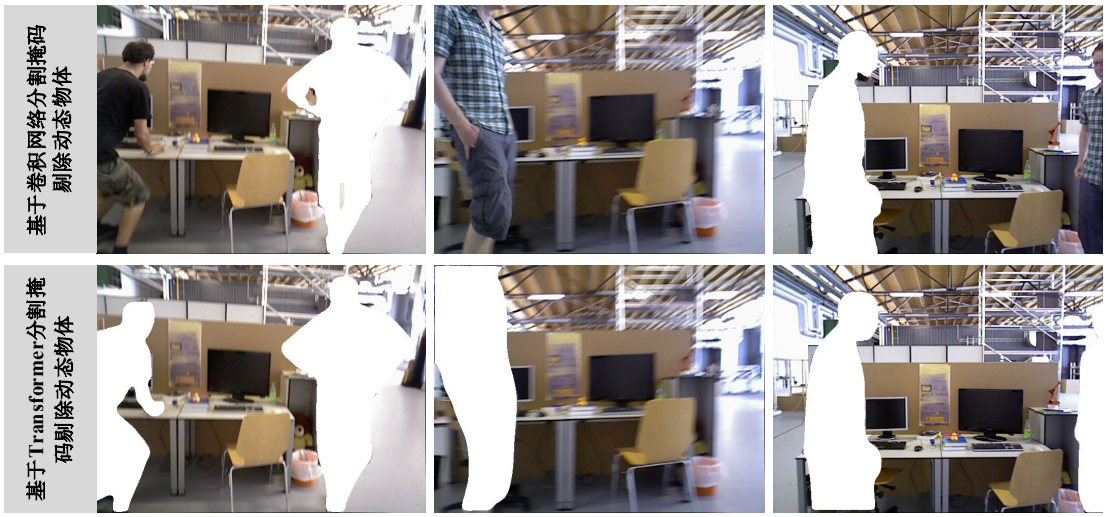


图 2-7 动态物体剔除效果图

ORB-SLAM2 算法和引入深度学习方法的视觉 SLAM 算法特征点分布效果图如图 2-8 所示。在引入深度学习网络后先标记潜在动态物体，然后根据判断剔除动态物体，从而剔除动态物体上所有的特征点，只将静态的特征点加入位姿计算中，降低动态点对 SLAM 系统定位的影响。

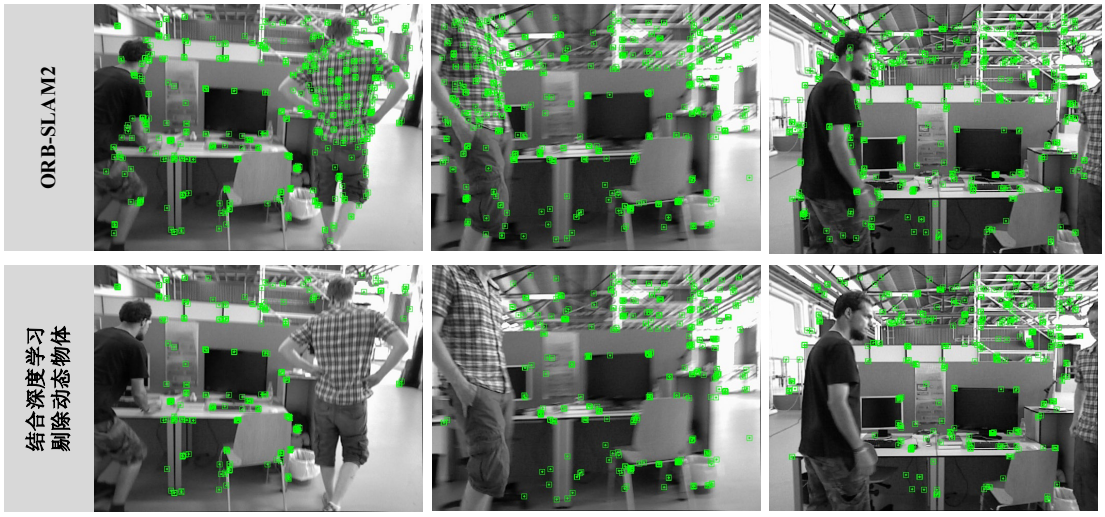


图 2-8 特征点分布效果图

Transformer 模型在图像分类任务中有着出色的表现，为接下来基于 Transformer 模型的物体分割方法提供了思路。将分割网络融入视觉 SLAM 前端，准确获取场景中存在的物体语义标签，语义标签与几何方法结合提高视觉 SLAM 系统在动态场景中运行的稳定性。但目前基于卷积的实例分割网络难以准确标记环境中的大物体和小物体，且传统视觉 SLAM 系统缺乏对标记物体运动的判断，因此视觉 SLAM 系统将难以构建出静态半稠密点云地图。解决以上问题是本文主要研究方向。

2.3 本章小结

本章介绍了视觉 SLAM 基础理论与 Transformer 基本框架。首先对视觉 SLAM2 的特征点提取与匹配、相机位姿求解、后端优化和回环检测进行简要分析，对 SLAM 系统中的地图构建进行了简单介绍，然后对 Transformer 基本框架进行简要阐述，并介绍了一种基于 Transformer 实例分割模型，最后将基于 Transformer 实例分割网络与基于卷积实例分割网络的分割效果进行对比，并比较这两种算法对动态物体剔除的效果，在动态场景下将基于 Transformer 实例分割的视觉 SLAM 算法与传统视觉 SLAM 算法 ORB-SLAM2 进行静态特征点分布对比，为动态场景中视觉 SLAM 的应用提供了理论基础。

第 3 章 基于 D-ORB 特征点提取的动态视觉 SLAM 算法

在动态场景中，由于移动机器人携带的 RGB-D 相机自身限制易出现深度图像部分深度值缺失的问题，使得系统在估计位姿时有效深度值特征点数量过少。部分深度值缺失的深度图像与特征点提取效果图如图 3-1 所示，其中蓝色特征点为无效深度值(深度值为零)特征点，绿色特征点为有效深度值(深度值大于零)特征点。在动态场景中传统视觉 SLAM 算法难以获取特征点的运动状态，难以高效剔除动态特征点，从而降低视觉 SLAM 系统定位和建图精度。

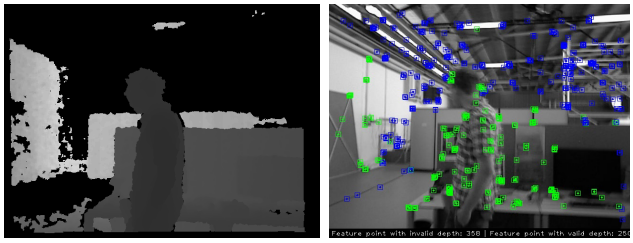


图 3-1 部分深度值缺失的深度图像与特征点提取效果图

针对移动机器人在动态场景中易出现有效深度值特征点数量过少和难以高效剔除动态特征点等问题，影响相机位姿估计精度。为此，提出一种基于 D-ORB 特征点提取的动态视觉 SLAM 算法。通过设计一种 D-ORB 特征点提取方法，及时划分出有效区域，同时对有效区域进行关联和 ORB 特征点提取，提高有效深度值特征点数量。为进一步减少动态特征点对 SLAM 系统定位精度的影响，联合位姿估计和偏转视角剔除动态点，提高视觉 SLAM 系统在动态场景中的位姿精度。本章技术路线框图如图 3-2 所示。

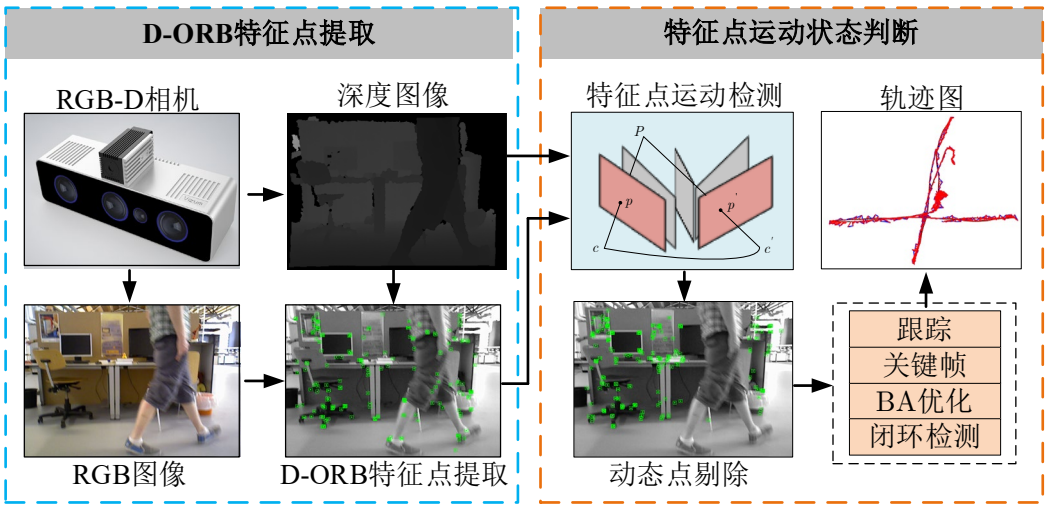


图 3-2 系统总框图

3.1 D-ORB 特征点提取方法

移动机器人使用 RGB-D 相机采集环境信息时，生成深度图像和对应的 RGB 图像帧。为获取移动机器人的当前位置，系统对 RGB 图像帧进行特征点提取，并将该帧上的特征点集合与上一帧特征点集合进行特征匹配得到匹配对，根据匹配对之间的约束关系计算位姿矩阵。假设第 k 帧特征点集合与第 k+1 帧特征点集合进行特征匹配后生成匹配对集合为 $\{\mathrm{Mch}_1, \mathrm{Mch}_2, \cdots, \mathrm{Mch}_n\}$，匹配对 $\mathrm{Mch}_i = \{\boldsymbol{p}_i^k, \boldsymbol{p}_i^{k+1}\}$。为获取位姿矩阵，系统依据对应的深度图像获取匹配对 Mch_i 中特征点 $\boldsymbol{p}_i^k$ 和 $\boldsymbol{p}_i^{k+1}$ 的深度值用于数据关联。当匹配对 Mch_i 中两点存在无效深度值的情况时，匹配对 Mch_i 中的特征点将无法进行数据关联，因此无法加入位姿计算环节。若在位姿计算过程中可利用的匹配对数量过少，将降低位姿估计精度，因此提高有效深度值特征点数量将提高匹配对集合中用于位姿估计的匹配对数量。为增加有效深度值特征点的数量，提高位姿估计精度，本章提出一种 D-ORB 特征点提取方法。该方法针对深度图像和对应的 RGB 图像建立金字塔模型，同时对金字塔中每层深度图像进行相同尺寸大小的区域划分。根据深度图像划分后的区域获取深度值分布，准确判断并保留有效区域。在特征点提取环节，将有效区域映射至对应的 RGB 图像中，对 RGB 图像中标定的有效区域进行 ORB 特征点提取，具体步骤如图 3-3 所示。

图 3-3 D-ORB 特征点提取方法

3.1.1 深度图像金字塔和图像区域划分

ORB 特征点提取方法在进行特征点提取时对输入的 RGB 图像构建金字塔，让系统获取更多图像特征信息，从而提高特征点提取的鲁棒性。为获取金字塔每层 RGB 图像的深度值分布，本章在 ORB 特征点提取方法的基础上构建深度图像金字塔。

将 RGB-D 相机采集到深度图像和对应的 RGB 图像分别构建金字塔模型，对于深度图像金字塔的第 L 层图像，都满足 DSA($\boldsymbol{\Gamma}_{org}$，$s^{L-1}$)。其中，DSA(•) 表示对图像进行降采样操作，$\boldsymbol{\Gamma}_{org}$ 表示输入系统的原始深度图像，s^{L-1} 表示将深度图像进行降采样的尺度因子。

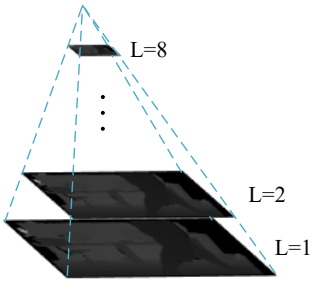


图 3-4 深度图像金字塔模型

本章遵循 ORB 特征点提取方法的金字塔构建方案，因此将 s 设置为 1.2，金字塔总层数设置为 8，深度图像金字塔的每层图像与 ORB 特征点提取方法中金字塔的每层 RGB 图像相对应。深度图像金字塔模型如图 3-4 所示。

ORB 特征点提取方法对金字塔每层 RGB 图像进行特征点提取之前，会对每层 RGB 图像进行区域划分，然后通过阈值响应检测并提取每个划分区域内的特征点。为准确获取每个划分区域内的深度值分布，本章对上述建立深度图像金字塔中的每层深度图像进行区域划分，划分方案与 ORB 特征点提取方法相同。区域划分示意图如图 3-5 所示。

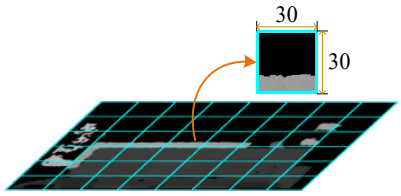


图 3-5 区域划分示意图

3.1.2 区域判定与有效区域 ORB 特征点提取

对深度图像金字塔每层深度图像进行区域划分后，获取每个划分区域的深度值分布，从而减少无效深度值特征点数量。对于深度图像金字塔中的第 L 层深度图像，得到划分区域的集合 $\boldsymbol{D}_L=\{\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_{Num}\}$，其中 Num 为划分区域总数量。对于第 i 个划分区域 $\boldsymbol{D}_i$，将通过 γ_i 反映子块中是否存在有效深度值，进而得到有效区域(γ_i=1)和无效区域(γ_i=0)中的某一种结果，具体计算方式如式(3-1)所

示：

$$\gamma_i=\begin{cases}1 & \text{if } \mathrm{DE}(\boldsymbol{D}_i)\neq 0\\ 0 & \text{if } \mathrm{DE}(\boldsymbol{D}_i)=0\end{cases} \tag{3-1}$$

其中，DE(•)是获取区域 $\boldsymbol{D}_i$ 有效深度值个数函数。

在对深度图像第 L 层图像进行区域判定后，将其中有效区域对应的位置映射到 RGB 图像金字塔的第 L 层中，对有效区域进行 ORB 特征点提取，从而提高了有效深度值特征点的数量，进而让更多匹配对加入位姿估计。区域判定与有效区域特征点提取示意图如图 3-6 所示。

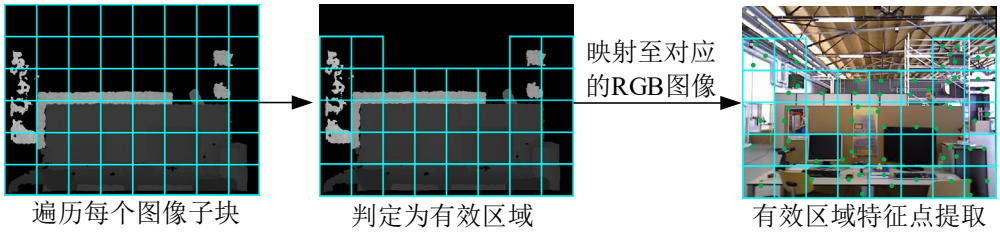


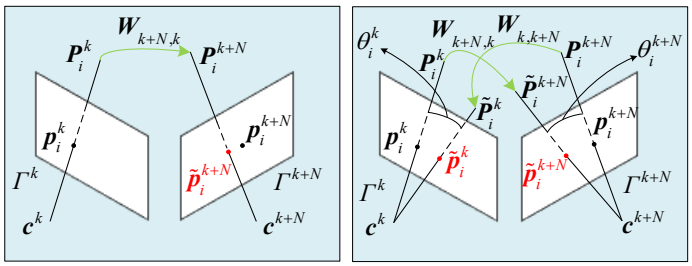
图 3-6 区域判定与有效区域特征点提取示意图

本章方法针对传统视觉 SLAM 算法在动态场景下存在深度图像部分深度值缺失，使得系统在估计位姿时有效深度值特征点数量过少的问题，提出一种 D-ORB 特征点提取的方法。该方法通过深度图像金字塔中每层深度图像划分区域的深度值分布判断否存在有效深度值，并将有效区域映射至对应 RGB 图像，完成有效区域 ORB 特征点提取。通过有效区域引导系统提取 ORB 特征点，提高了有效深度值特征点数量，这使得更多具有有效深度值的特征点加入到位姿估计中，从而提高了位姿估计的精度。

3.2 基于偏转视角的特征点运动状态判断

传统视觉 SLAM 算法由于缺乏特征点运动判断策略难以剔除动态点，从而降低视觉SLAM运行稳定性。为提高视觉SLAM系统在动态场景下的定位性能，需要进行位姿估计、特征点运动估计与判断来筛选并剔除动态点。

在位姿估计环节，由于运动幅度小的特征点在相邻帧之间的实际运动状态难以被准确观测，因此本章间隔一定帧数进行位姿估计，从而在特征点运动估计与判断环节能更准确地表征潜在动态特征点的运动状态。在特征点运动估计与判断环节，通过运动一致性约束获取潜在动态特征点的偏转误差，判断和剔除动态特征点。特征点运动判断如图 3-7 所示。



(a)位姿估计　　(b)特征点运动估计

图 3-7 特征点运动判断

3.2.1 位姿估计

视觉 SLAM 系统在实时运行的过程中，通过相机参数和特征点深度值可以将图像特征点的二维坐标映射为世界坐标。如图 3-7(a)所示，将帧 Γ^k 的第 i 个二维特征点 $\boldsymbol{p}_i^k=(u_i^k,v_i^k)$ 反投影到世界坐标系，得到三维点 $\boldsymbol{P}_i^k=(X_i^k,Y_i^k,Z_i^k)$，再将点 $\boldsymbol{P}_i^k$ 投影到帧 Γ^{k+N} 中得到投影点 $\tilde{\boldsymbol{p}}_i^{k+N}=(\tilde{u}_i^{k+N},\tilde{v}_i^{k+N})$，计算方式如式(3-2)所示：

$$\begin{cases}\boldsymbol{P}_i^k=\Pi^{-1}\left(\boldsymbol{l}_i^k\right)\\ \tilde{\boldsymbol{l}}_i^{k+N}=\Pi\left(\boldsymbol{W}_{k+N,k}\left(\boldsymbol{L}_i^k\right)^{\mathrm{T}}\right)\end{cases}\tag{3-2}$$

其中，$\boldsymbol{l}_i^k=(\boldsymbol{p}_i^k,1)$，$\tilde{\boldsymbol{l}}_i^{k+N}=(\tilde{\boldsymbol{p}}_i^{k+N},1)$，$\boldsymbol{L}_i^k=(\boldsymbol{P}_i^k,1)$，$\Pi^{-1}$ 和 Π 分别表示反投影函数和投影函数，$\boldsymbol{W}_{k+N,k}\in\mathrm{SE}(3)$ 表示从第 k 帧到第 $k+N$ 帧的相对位姿变换矩阵，N 表示间隔帧数。

为了准确估计相对位姿变换矩阵 $\boldsymbol{W}_{k+N,k}$，本章构建了重投影误差项，并联合最小二乘法和高斯-牛顿法优化求解得到最优解 $\tilde{\boldsymbol{w}}_{k+N,k}^{\vee}\in\mathbb{R}^6$，最后得到相对位姿变换矩阵的最优解 $\boldsymbol{W}_{k+N,k}$，计算方法如式(3-3)所示：

$$\begin{cases}\mathrm{e}_i^{\mathrm{t}}\left(\boldsymbol{W}_{k+N,k}\right)=\boldsymbol{p}_i^{k+N}-\tilde{\boldsymbol{p}}_i^{k+N}\\ \tilde{\boldsymbol{w}}_{k+N,k}^{\vee}=\underset{\tilde{\boldsymbol{w}}_{k+N,k}^{\vee}}{\operatorname{argmin}}\sum_{i=1}^{n}\rho_{\mathrm{Huber}}\left(\mathrm{e}_i^{\mathrm{t}}\left(\boldsymbol{w}_{k+N,k}\right)^{\mathrm{T}}\boldsymbol{\Omega}_p^{-1}\mathrm{e}_i^{\mathrm{t}}\left(\boldsymbol{w}_{k+N,k}\right)\right)\\ \boldsymbol{W}_{k+N,k}=\exp\left(\boldsymbol{w}_{k+N,k}\right)\end{cases}\tag{3-3}$$

其中，$\boldsymbol{w}_{k+N,k}\in se(3)$ 表示 k 帧到 $k+N$ 帧的相机位姿的相对位姿变换向量，$\boldsymbol{\Omega}_p$ 表示重投影误差的协方差矩阵，ρ_{Huber} 表示惩罚因子，n 表示在第 k 帧中与第 $k+N$ 帧所匹配特征点的数量，$\mathrm{e}_i^{\mathrm{t}}$ 为重投影误差因子。

3.2.2 特征点运动估计与判断

视觉 SLAM 系统运行过程中易受动态特征点的影响，若不能准确判断潜在动态特征点运动状态，将会降低 SLAM 系统鲁棒性。如图 3-7(b)所示，帧 Γ^{k+N} 上的二维点 $\boldsymbol{p}_i^{k+N}$ 在经过相对位姿变换矩阵变换后，将映射到帧 Γ^k 上得到投影点 $\tilde{\boldsymbol{p}}_i^k$。在世界坐标系中，从光心点 $\boldsymbol{c}^k$ 沿直线分别观测帧 Γ^k 上的点 $\tilde{\boldsymbol{p}}_i^k$ 和点 $\boldsymbol{p}_i^k$，在观测过程中得到从 $\tilde{\boldsymbol{p}}_i^k$ 到 $\boldsymbol{p}_i^k$ 的视角偏离误差。为了描述潜在动态点在不同帧间的视角偏离程度，本章在帧与帧之间观测潜在动态点的视角偏离误差，将潜在动态对象建模成一个带有偏转误差的实体 $\overline{\theta}_i$，其计算方法如式(3-4)所示：

$$\begin{cases} \theta_i^k = \Theta\left(\overrightarrow{\boldsymbol{c}^k\boldsymbol{m}_i^k}, \overrightarrow{\boldsymbol{c}^k\tilde{\boldsymbol{m}}_i^k}\right) \\ \overline{\theta}_i = \theta_i^k + \theta_i^{k+N} \end{cases} \tag{3-4}$$

其中，θ_i^k 表示第 i 组已匹配特征点在帧 Γ^k 上的视角偏离误差，Θ 表示获取向量之间夹角的夹角因子，$\boldsymbol{m}_i^k$ 和 $\tilde{\boldsymbol{m}}_i^k$ 分别表示匹配点 $\boldsymbol{p}_i^k$ 和投影点 $\tilde{\boldsymbol{p}}_i^k$ 在世界坐标系中的坐标。

在特征点运动判断阶段，将特征点运动估计环节得到的偏转误差与设定阈值 θ^{ref} 进行对比，若超过阈值则判定该特征点为动态点，反之为静态点。

3.3 实验结果与分析

实验分为三部分：有效深度值特征点的匹配性能测试，特征点运动判断实验，定位性能测试。并将本章算法与 ORB-SLAM2 在 TUM 数据集上进行性能对比。本章实验所用平台软硬件配置为：CPU 为 Inter i9-12900K 处理器，主频 3.2 GHZ，GPU 为 RTX3090 显卡，系统为 Ubuntu18.04。

3.3.1 有效深度值特征点的匹配性能测试

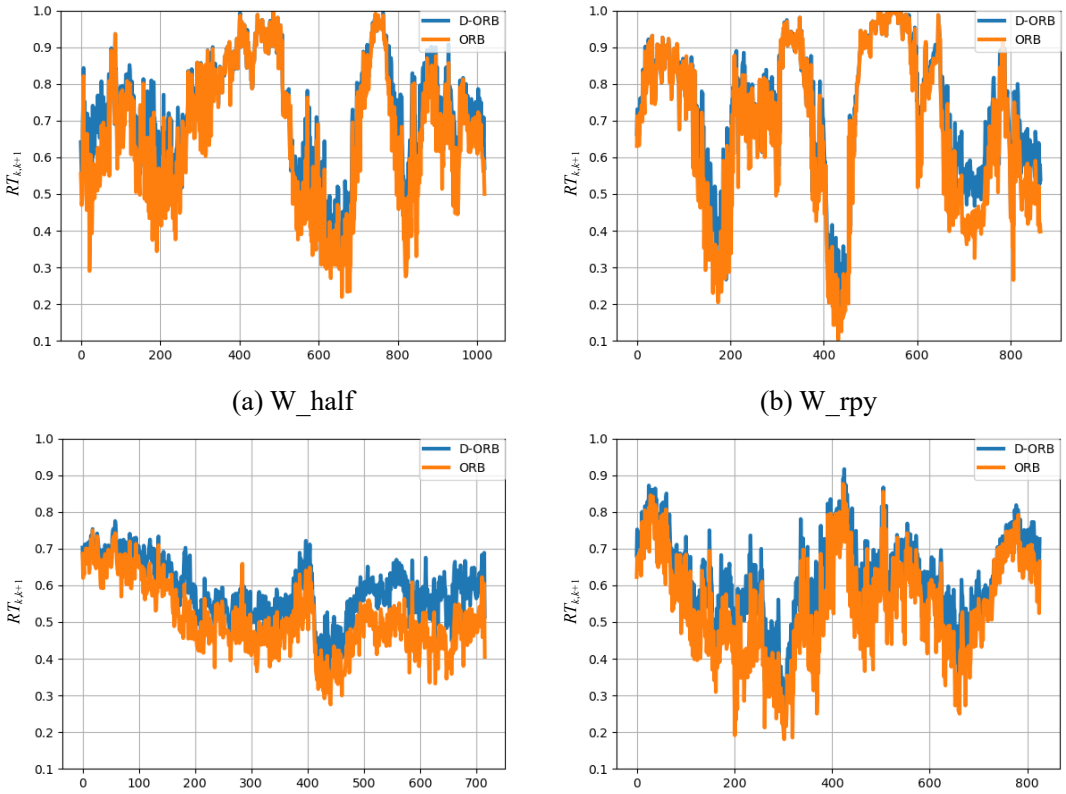
本章在 TUM 数据集中选取 W_half（freiburg3_walking_halfsphere）、W_static（freiburg3_walking_static）、W_rpy（freiburg3_walking_rpy）和 W_xyz（freiburg3_walking_xyz）四个序列对本章算法提取的特征点和传统方法提取的 ORB 特征点进行特征匹配，并对每个匹配对 Mch_i 中特征点的深度值进行分析。本节实

验以第 k 帧与第 k+1 帧间的比值 $RT_{k,k+1}$ 作为算法性能的评价指标， $RT_{k,k+1}$ 的计算方式如公式(3-5)所示：

$$RT_{k,k+1}=\frac{\sum_{i=1}^{N^m}\upsilon_i}{N^m} \tag{3-5}$$

其中，当第 i 个 Mch_i 中的点 $\boldsymbol{p}_i^k$ 和 $\boldsymbol{p}_i^{k+1}$ 的深度值都大于零时，对应的 υ_i 设置为 1，其它情况 υ_i 设置为 0， N^m 表示第 k 帧与第 k+1 帧经过特征匹配后总匹配对数量。

实验结果效果图如图 3-8 所示。由图可知，与使用 ORB 特征点提取方法相比，使用 D-ORB 特征点提取方法得到的比值 $RT_{k,k+1}$ 更高，尤其是在使用 ORB 特征点提取方法得到比值 $RT_{k,k+1}$ 较低的情况下提升更多，这是因为本章算法通过深度图像划分的区域获取深度值分布，准确判断并获取有效区域，将有效区域关联到对应 RGB 图像，引导系统对有效区域进行 ORB 特征点提取，增加有效深度值特征点数量，提高 $RT_{k,k+1}$ 比值，从而让更多有效深度值特征点加入到位姿估计中，进而提高 SLAM 系统定位精度。



(a) W_half　　(b) W_rpy

(c) W_static　　(d) W_xyz

图 3-8 使用 D-ORB 和 ORB 两种方法得到的比值 $RT_{k,k+1}$ 对比图

对上述某个测试序列中所有的比值 $RT_{k,k+1}$ 进行求和，并计算得到平均值 $\overline{RT}$，计算方式如公式(3-6)所示：

$$\overline{RT}=\frac{1}{N_F-1}\sum_{i=1}^{N_F-1}RT_{i,i+1} \tag{3-6}$$

其中，N_F 为数据集总帧数。

如表 3-1 所示为两种算法运行四个序列得到的 $\overline{RT}$ 值对比表格。由表中数据可知，使用本章算法提出的 D-ORB 特征点提取方法得到的比值 $\overline{RT}$ 要高于使用 ORB 特征点提取方法的比值。本章提出的 D-ORB 特征点提取方法之所以要优于 ORB 特征点提取方法，是因为本章在使用 RGB-D 相机时，考虑到一些特征明显的区域深度值是否有效，通过判断及时获取有效区域，从而提高有效深度值特征点加入特征匹配环节的数量，进而让系统在利用特征点以及对应深度值进行位姿估计时，能够输入更多有有效深度值的特征点。本章算法不仅提高了特征点的利用效率，还增加了有效深度值的特征点数量，提高了位姿估计的精度，验证了 D-ORB 特征点提取方法对提高有效深度值特征点数量的有效性。

表 3-1 两种算法运行四个序列得到的 $\overline{RT}$ 值对比

序列	ORB	D-ORB	性能提升
W_xyz	0.54	0.62	14.81%
W_half	0.66	0.72	9.09%
W_static	0.51	0.58	13.72%
W_rpy	0.68	0.72	5.88%

3.3.2 特征点运动判断实验

为验证本章特征点运动状态评估策略的有效性，选取与 3.3.1 节实验相同的数据集对 ORB-SLAM2 和本章算法进行特征点运动判断实验。ORB-SLAM2 和本章算法特征点运动判断效果图如图 3-9 所示，其中，红色框为标记辅助框，算法判定为动态点设置为红色，判定为静态点设置为绿色。图 3-9(a)(d)(g)(j)为所选序列中的某一帧原始彩色图像。由图 3-9(b)(e)(h)(k)可知 ORB-SLAM2 无法剔除特征点，这是因为该算法没有特征点运动判断线程，因此在动态场景中难以准确获取特征点运动状态，从而无法及时筛选出动态点。图 3-9(c)(f)(i)(l)为本章算法（用 Ours 标注）特征点运动状态判断效果图，由图可知本章算法在引入特征点

运动判断后能有效区分出大多数动态特征点，将这些动态点进行剔除，减少动态点的数量，从而提高视觉 SLAM 系统定位精度。本章提出的算法减少了动态点对视觉 SLAM 系统的影响，验证了本章基于偏转视角的特征点运动状态判断策略的有效性。

(a) W_half 图像帧　(b) ORB-SLAM2　(c)Ours

(d) W_rpy 图像帧　(e) ORB-SLAM2　(f)Ours

(g) W_static 图像帧　(h) ORB-SLAM2　(i)Ours

(j) W_xyz 图像帧　(k) ORB-SLAM2　(l)Ours

图 3-9 ORB-SLAM2 和本章算法判断特征点运动判断效果图

3.3.3 定位性能测试

在本节实验中，使用了与 3.3.1 节相同数据集测试本章算法和 ORB-SLAM2 算法的定位性能，并将本章算法轨迹构建图像效果与 ORB-SLAM2 轨迹构建图

像效果进行对比。ORB-SLAM2 和本章算法在 TUM 数据集下的绝对轨迹均方根误差（Root Mean Squared Error，RMSE）数据和相对位姿误差(Relative Pose Error，RPE)数据如表 3-2 和表 3-3 所示，从表中数据可以看出，本章提出的算法在轨迹精度和位姿精度方面较 ORB-SLAM2 算法有着较大提升。本章算法的绝对轨迹均方根误差与 ORB-SLAM2 算法相比平均降低了 70.18%，相对位姿误差与 ORB-SLAM2 算法相比平均降低了 80.27%。实验结果表明，在动态场景下，本章算法较传统算法 ORB-SLAM2 在轨迹构建方面优势明显。

表 3-2 两种算法绝对轨迹均方根误差对比

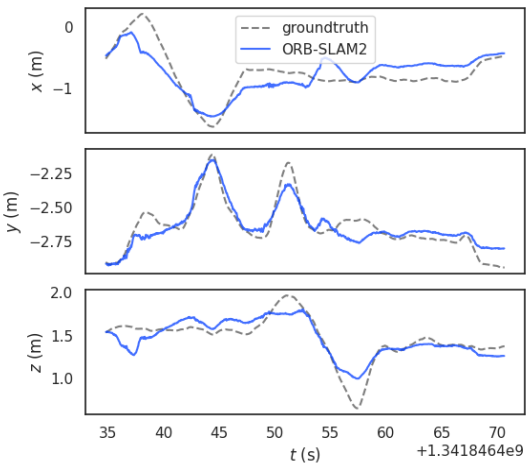
序列	ORB-SLAM2/m	Ours/m	性能提升
W_half	0.351	0.101	71.23%
W_xyz	0.459	0.098	78.65%
W_static	0.090	0.028	68.89%
W_rpy	0.662	0.252	61.93%

表 3-3 两种算法相对位姿误差对比

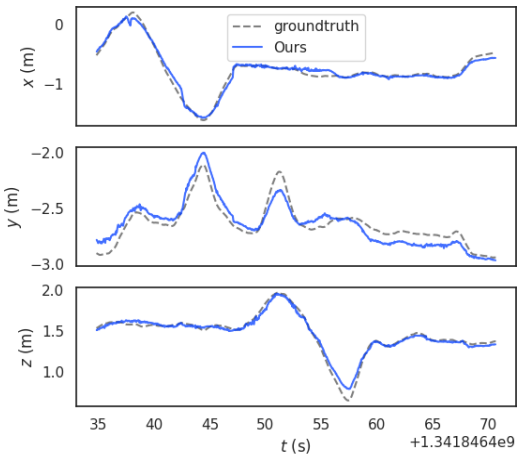
序列	ORB-SLAM2/m	Ours/m	性能提升
W_half	0.482	0.117	75.73%
W_xyz	0.864	0.102	88.19%
W_static	0.201	0.036	82.09%
W_rpy	0.814	0.203	75.06%

根据算法运行的数据绘制相应轨迹图，ORB-SLAM2 算法与本章算法分别在 x 轴、y 轴和 z 轴方向上的轨迹对比效果图如图 3-10 所示。其中，虚线线条为序列真实轨迹，实线线条为算法估计轨迹。图 3-10(a)(c)(e)(g)可知 ORB-SLAM2 算法轨迹估计与真实轨迹偏移误差较大，这是因为 ORB-SLAM2 算法缺乏特征点运动状态判断策略，当 ORB-SLAM2 算法运行于动态场景中，由于动态特征点缺乏约束条件，因此动态点会被引入位姿估计线程中，从而使系统定位精度降低。由图 3-10(b)(d)(f)(h)可知本章算法轨迹误差比 ORB-SLAM2 小，本章算法轨迹精度之所以优于 ORB-SLAM2，是因为本章算法引入了 D-ORB 特征点提取策略，通过 D-ORB 特征点提取方法提高有效深度值特征点的数量，使得系统在进行位姿计算环节能够获取更多有效深度值特征点，为降低动态点对视觉 SLAM 系统的影响，通过引入一种基于偏转视角的特征点运动状态判断策略，利用偏转视角评价特征点角度变化，将角度变化较大的特征点设置为动态点，同时及时筛选出

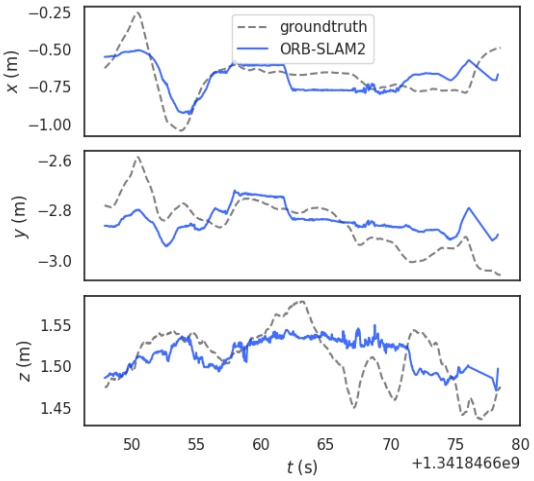
动态特征点，减少动态点的数量，从而降低动态点对 SLAM 系统定位的影响。本章算法轨迹估计精度与 ORB-SLAM2 算法相比有较大的提升，因此本章提出的基于偏转视角的特征点运动状态判断策略能有效提高轨迹构建精度。



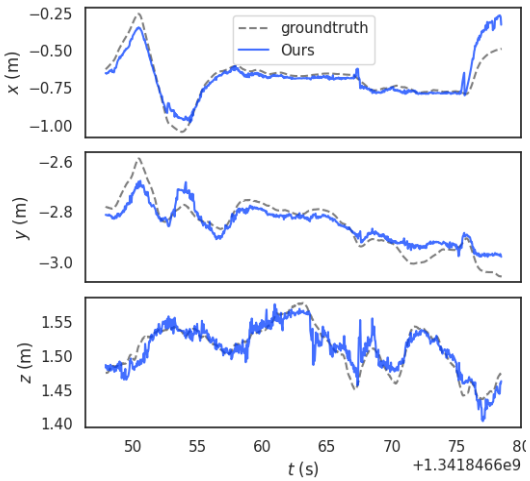
(a) ORB-SLAM2 W_half



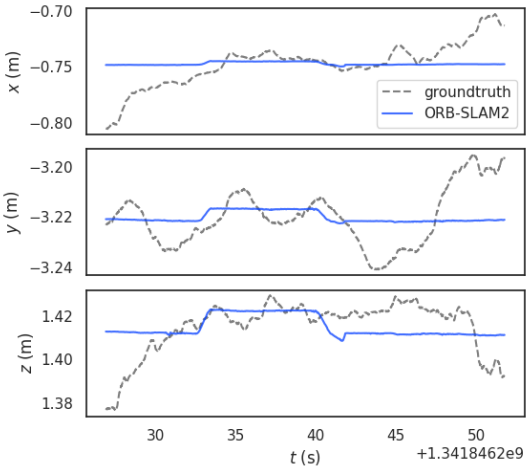
(b) Ours W_half



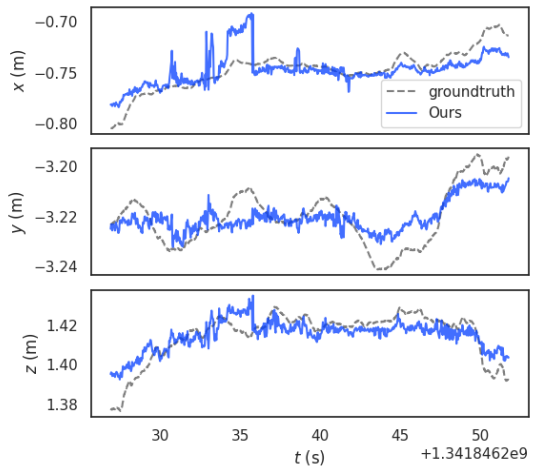
(c) ORB-SLAM2 W_rpy



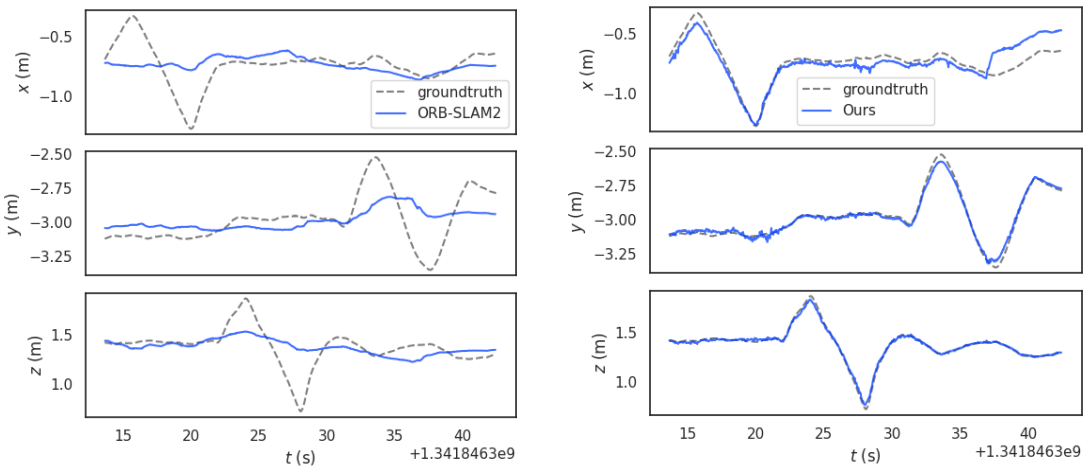
(d) Ours W_rpy



(e) ORB-SLAM2 W_static



(f) Ours W_static



(g) ORB-SLAM2 W_xyz　　(h) Ours W_xyz

图 3-10 ORB-SLAM2 算法与本章算法轨迹对比效果图

3.4 真实场景实验

为验证本章所提算法有效性，设置真实场景进行实验，真实场景布局如图 3-11(c)所示。将 Intel RealSense D435i 传感器(图 3-11(b))安装在 Husky 机器人(图 3-11(a))上方，机器人软硬件配置为：CPU 为 i7-10875H 处理器，GPU 为 GTX1080，操作系统为 Ubuntu 18.04。移动机器人沿着箭头方向从 RB1 到 RB2 行进，移动机器人行进期间行人在 PN1 点与 PN2 点之间往复行走，直到移动机器人到达终点 RB2。真实场景实验包括有效深度值特征点的匹配性能测试、动态点剔除以及轨迹构建。

(a) 轮式移动机器人　(b) Intel RealSense D435i 传感器　(c) 真实场景布局

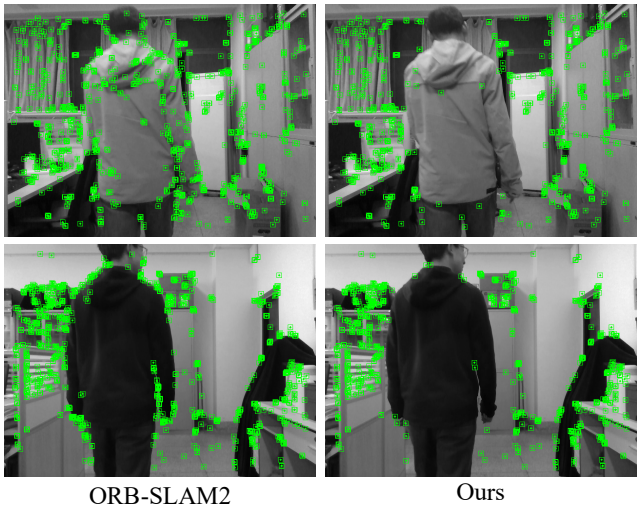
图 3-11 真实场景设置

3.4.1 有效深度值特征点的匹配性能测试

将真实场景中得到的特征点进行特征匹配，并记录真实场景中所有相邻帧间的比值 $RT_{k,k+1}$，计算得到平均值 $\overline{RT}$。根据系统输出的平均值 $\overline{RT}$ 结果，本章算法和 ORB-SLAM2 算法分别为 0.71 和 0.66。本章算法之所以优于 ORB-SLAM2 算法，因为本章算法在筛选出无效区域后，使每个有效区域提取到特征点数量增加，这些特征点大部分具有有效深度值，因此在进行完特征匹配得到的匹配对集合中，加入到位姿估计的匹配对数量增多，从而提高位姿估计的精度。

3.4.2 动态点剔除实验

ORB-SLAM2 算法和本章算法在真实场景实验中剔除动态点效果图如图 3-12 所示，由图可知本章算法能剔除大部分动态点，而 ORB-SLAM2 算法难以获取特征点运动状态，难以剔除动态点。因为本章算法根据深度图像选取了有效区域，对有效区域进行关联后提取到更多有效深度值特征点，提高了位姿估计的精度。为判断环境中特征点运动状态，将得到的位姿矩阵加入约束中得到偏转误差，通过帧间偏转误差间约束获取偏转视角，根据设定阈值评估特征点运动状态，并及时区分出检测到的动态点。



ORB-SLAM2　　Ours

图 3-12 动态点剔除效果图

3.4.3 轨迹构建实验

本章算法和 ORB-SLAM2 算法轨迹构建效果对比图如图 3-13 所示。在动态

场景中，由于 ORB-SLAM2 算法缺乏特征点运动状态判断策略，使得系统在轨迹构建线程中难以区分出动态点，从而降低了轨迹构建精度。本章算法根据深度图像划分的区域，利用其中深度值分布及时得到有效区域和无效区域，并只对有效区域进行特征点提取，提高了有效深度值特征点数量。在特征点运动状态判断环节，联合位姿估计和偏转视角计算特征点在动态场景下的运动状态，并剔除掉动态点，进而降低动态点对轨迹构建的影响。

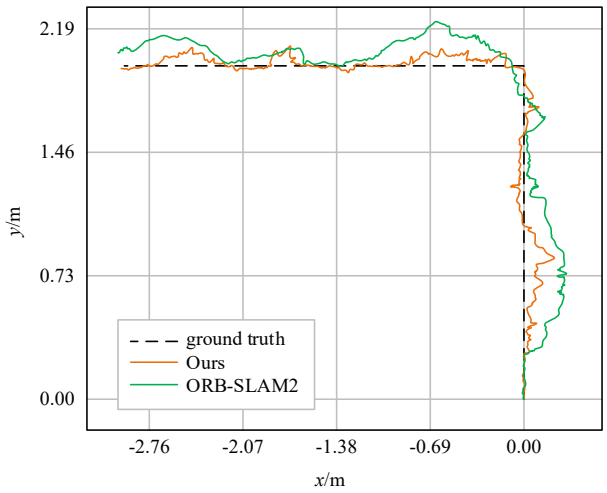


图 3-13 轨迹对比图

3.5 本章小结

本章节提出一种 D-ORB 特征点提取的动态视觉 SLAM 算法。在 D-ORB 特征点提取环节中，首先构建深度图像金字塔，对金字塔中每层深度图像进行区域划分，然后对这些区域进行有效深度值判断并去除无效区域，这使得更多具有有效深度值的特征点加入到利用深度值和对应的特征点进行位姿估计的策略中。在特征点运动状态估计环节，通过位姿估计和偏转视角预测特征点在动态场景下的运动状态，并及时剔除动态点，从而提高 SLAM 系统在动态场景下运行的鲁棒性。本章算法在 TUM 数据集中和真实场景中进行验证，在 TUM 数据集中，本章算法的绝对轨迹均方根误差与 ORB-SLAM2 算法相比平均降低了 70.18%，相对位姿误差与 ORB-SLAM2 算法相比平均降低了 80.27%，在真实场景中，本文算法与 ORB-SLAM2 相比，本章提出的算法表现出更小的轨迹误差。

第 4 章 基于实例分割的动态视觉 SLAM 算法

由于第 3 章算法无法完全剔除动态点，难以完全消除动态点对 SLAM 系统的影响。为此，本章在第 3 章算法的基础上，提出一种基于实例分割的动态视觉 SLAM 算法。目前传统的视觉 SLAM 算法在动态场景下易出现难以完整标记潜在动态物体和无法准确判断物体运动状态等问题，为此，提出一种基于实例分割的动态视觉 SLAM 算法。在潜在动态物体实例分割环节，通过设计一种 ReT-encoder 模块，引导模型关注并提取图像局部特征信息和全局特征信息，同时设计一种混合权重 FPN 模块，通过权重系数融合学习大物体特征和小物体特征，精确标记潜在动态物体区域。在潜在动态物体运动状态判断环节，联合第 3 章算法剔除动态点后估计的位姿与本章实例分割网络输出的掩码构建物体投影偏差，从而及时剔除动态物体。本章算法框图如图 4-1 所示。

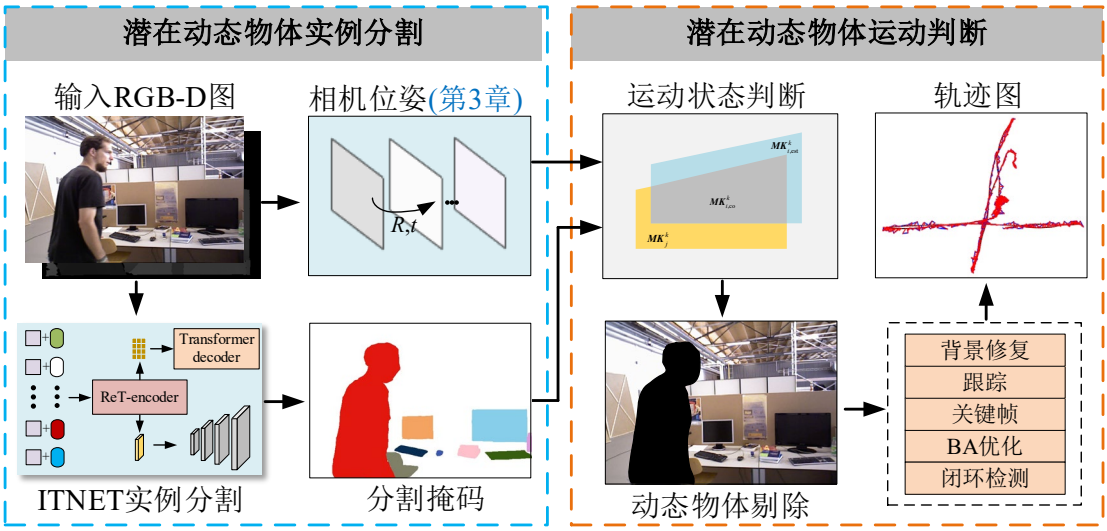


图 4-1 系统总框图

4.1 基于改进 Transformer 实例分割网络

传统实例分割算法在分割大物体和小物体时，易出现分割失效或者分割效果较差等问题。为此，本章改进一种 Transformer 实例分割网络 ITNET（Improved Transformer encoder NETwork for instance segmentation）。该网络包含提取图像特征的主干网络模块和提高语义信息融合准确率的分割模块。主干网络模块的 ReT-encoder 层使用区域多头注意力机制和多头注意力机制关注图像中潜在动态物体的全局特征信息和局部特征信息，从而生成更具有判别能力和表达能力的特征图，

并将得到的具有高维丰富信息的特征图输入分割模块。分割模块中的混合权重FPN(M-FPN)模块通过权重系数融合并学习大物体和小物体的特征信息，提高标记潜在动态物体所在区域的准确率。如图 4-2 所示为 ITNET 网络结构图。

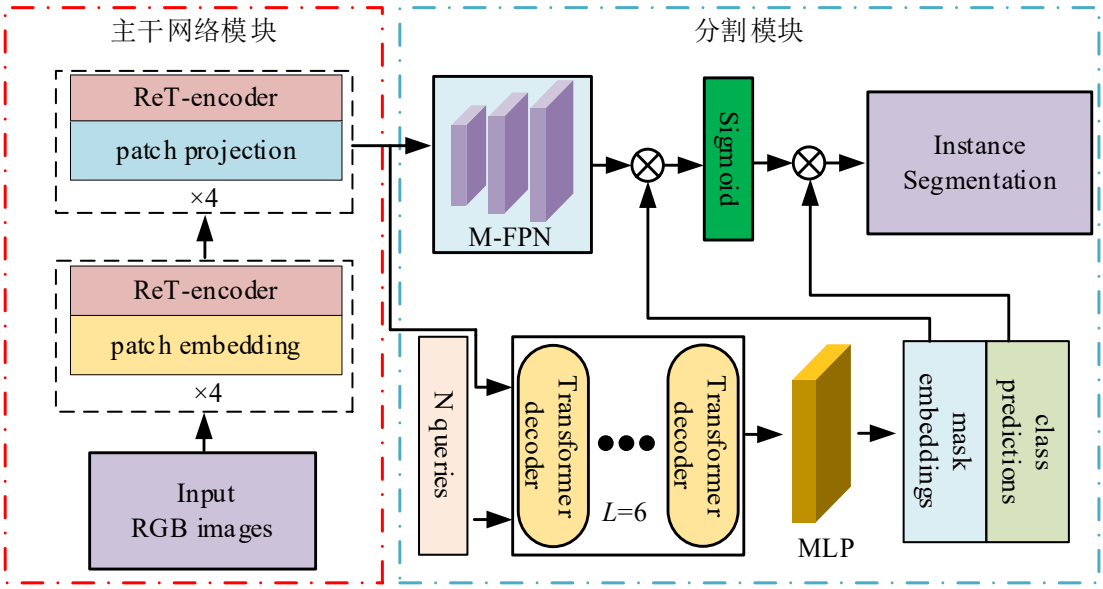


图 4-2 ITNET 网络结构图

4.1.1 主干网络模块

传统 Transformer 模块的注意力机制在关注中、大型物体特征信息方面有着出色的表现，但在关注小物体特征信息方面准确率较低，难以生成具有判别能力的小物体特征信息。为了更准确地提取潜在动态物体特征信息，本章通过改进 Transformer 中的注意力机制，设计了一种 ReT-encoder 模块，该模块是由两个 Transformer encoder 层和一个改进的 Transformer encoder 层组成。将特征图 $\boldsymbol{F}_{n\text{-}1}$ 输入 ReT-encoder 模块经过 Transformer encoder 层得到具有全局特征信息的特征图 $\boldsymbol{F}_1$，并将 $\boldsymbol{F}_1$ 输入改进的 Transformer encoder 层得到具有丰富局部特征信息的特征图 $\boldsymbol{F}_2$，从而生成更具有判别能力的小物体特征信息。通过 Transformer encoder 层建立特征图 $\boldsymbol{F}_2$ 中各个局部特征信息的全局联系，最后得到具有丰富局部特征信息和全局特征信息的特征图 $\boldsymbol{F}_n$。ReT-encoder 模块不仅能充分获取大物体特征信息，还能联合区域多头注意力（Region Multi-head Self Attention，RMSA）机制和多头注意力（Multi-head Self Attention，MSA）机制更准确地关注并学习小物体特征信息，从而更准确地分割潜在动态物体。ReT-encoder 结构图如图 4-3 所示。

特征图 $\boldsymbol{F}_{n\text{-}1}$ 在进行 Norm 归一化处理后进入 MSA 多注意力机制模块中，MSA 模块通过权重分配加强对特征的学习，联合多头特性增加特征图特征的丰富性，进一步加强特征的表达，并通过 MLP 引导学习特征的分类，特征图 $\boldsymbol{F}_{n\text{-}1}$ 转化为特征图 $\boldsymbol{F}_1$ 的过程如公式(4-1)所示：

$$\begin{cases} \boldsymbol{F}_{\text{pre},1} = \text{MSA}(\text{Norm}(\boldsymbol{F}_{n-1}))+\boldsymbol{F}_{n-1} \\ \boldsymbol{F}_1 = \text{MLP}(\text{Norm}(\boldsymbol{F}_{\text{pre},1}))+\boldsymbol{F}_{\text{pre},1} \end{cases} \tag{4-1}$$

通过改进多注意力机制中的注意力窗口得到 RMSA 模块，重新对小物体特征进行权重分配，进而增加对小物体特征表达的丰富性，提高对小物体特征的学习能力，特征图 $\boldsymbol{F}_1$ 转化为特征图 $\boldsymbol{F}_2$ 的过程如公式(4-2)所示：

$$\begin{cases} \boldsymbol{F}_{\text{pre},2} = \text{RMSA}(\text{Norm}(\boldsymbol{F}_1))+\boldsymbol{F}_1 \\ \boldsymbol{F}_2 = \text{MLP}(\text{Norm}(\boldsymbol{F}_{\text{pre},2}))+\boldsymbol{F}_{\text{pre},2} \end{cases} \tag{4-2}$$

特征图 $\boldsymbol{F}_2$ 此时具有丰富的大物体特征信息和丰富的小物体特征信息。为得到更为丰富特征的特征图，将特征图 $\boldsymbol{F}_2$ 重新进行全局的权重分配以增强大物体和小物体的特征表达，进而得到丰富特征的特征图 $\boldsymbol{F}_n$。特征图 $\boldsymbol{F}_n$ 能更准确地学习大物体特征和小物体特征，从而提高大物体和小物体区域标记的精度，$\boldsymbol{F}_n$ 由公式(4-3)得到：

$$\begin{cases} \boldsymbol{F}_{\text{pre},3} = \text{MSA}(\text{Norm}(\boldsymbol{F}_2))+\boldsymbol{F}_2 \\ \boldsymbol{F}_n = \text{MLP}(\text{Norm}(\boldsymbol{F}_{\text{pre},3}))+\boldsymbol{F}_{\text{pre},3} \end{cases} \tag{4-3}$$

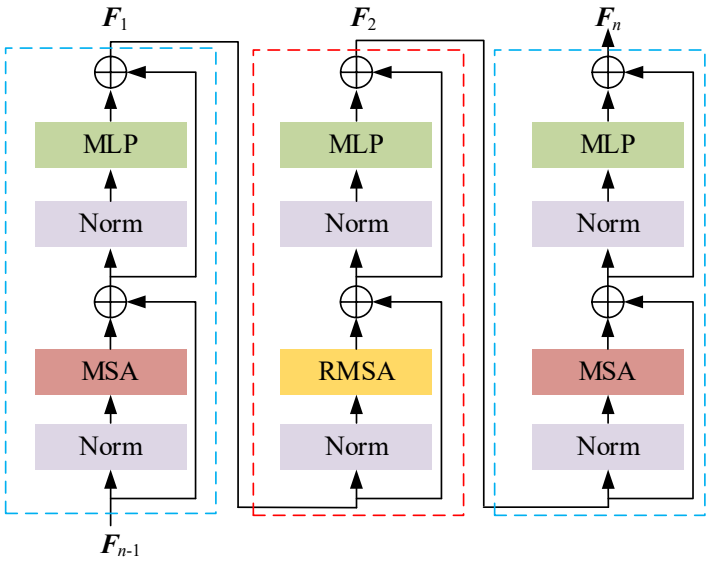


图 4-3 ReT-encoder 结构图

如图 4-4 所示为 RMSA 机制原理图。区域多头注意力机制的作用为将特征图划分为四个相同大小的子区域，引导模型关注各个子区域的全局特征信息，进

而更准确地学习潜在动态物体的特征。将尺寸大小为 $H\times W$ 的特征图 $\boldsymbol{F}$ 分为四个尺寸大小相同的子特征图，并将得到的子特征图输入多头注意力机制中，引导模型关注子特征图区域，从而得到子特征图内部的全局联系信息。经过注意力机制关注的子特征图进行 Reshape 融合后得到具有丰富局部特征信息的特征图 $\boldsymbol{F}^r$，更好地对潜在动态物体特征进行学习。

特征图 $\boldsymbol{F}$ 经过 Unfold 操作以缩小注意力窗口，并将得到的四个子特征图输入 MSA 模块以关注学习小物体特征。经过 MSA 模块的子特征图最后经过 Reshape 融合将得到特征图 $\boldsymbol{F}^r$。特征图 $\boldsymbol{F}^r$ 将更高效地学习小物体特征，从而提高小物体分割的准确率。特征图 $\boldsymbol{F}$ 经过 RMSA 模块后得到特征图 $\boldsymbol{F}^r$ 的过程如式(4-4)所示：

$$\boldsymbol{F}^r = \text{Reshape}(\text{MSA}(\text{Unfold}(\boldsymbol{F}))) \tag{4-4}$$

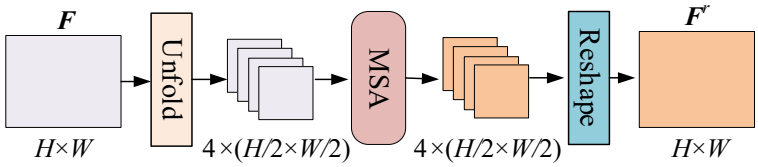


图 4-4 RMSA 机制原理图

4.1.2 基于混合权重 FPN 的分割模块

为了更准确地标记潜在动态物体所在区域，本章设计一种混合权重的 FPN 模块，该模块由融合小物体特征的 SW-FPN 模块和融合大物体特征的 LW-FPN 组成。将特征图 $\boldsymbol{F}$ 输入 LW-FPN 模块和 SW-FPN 模块得到特征图 $\boldsymbol{F}_L$ 和 $\boldsymbol{F}_S$，并将得到的特征图进行 concat 融合和 1×1 卷积得到特征图 $\boldsymbol{F}_m$，从而得到更具有判别能力的特征表达。混合权重 FPN 模块不仅有效融合了小物体的特征信息，还更准确表达了大物体所在区域的特征，充分突出潜在动态物体所在区域的有效信息。混合权重 FPN 模块结构如图 4-5 所示。

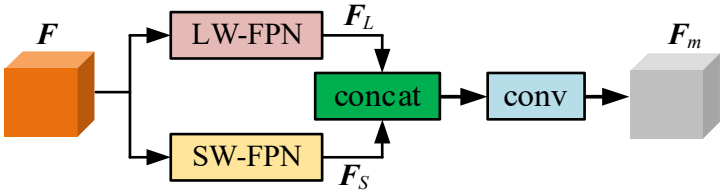


图 4-5 混合权重 FPN 模块结构

SW-FPN 模块和 LW-FPN 模块可通过设置融合权重 FPN 模块的权重系数而

得到，融合权重 FPN 模块的结构图如图 4-6 所示。融合权重 FPN 模块的作用为通过分配不同的权重系数，进而在上采样过程中融合不同类型的特征，更准确地学习到潜在动态物体的特征。

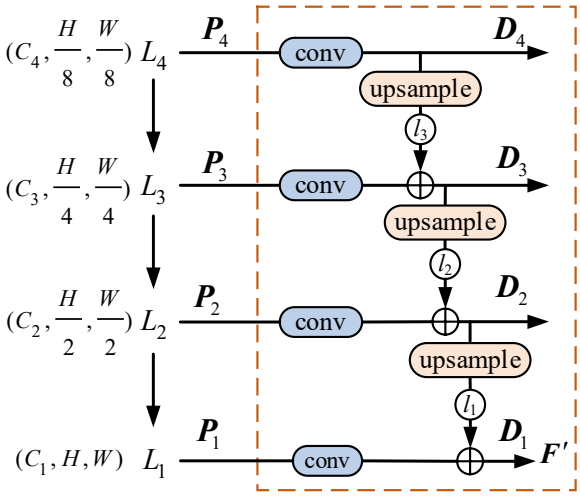


图 4-6 融合权重 FPN 结构图

如图 4-6 所示，将来自主干网络模块的特征图 $\boldsymbol{F}$ 输入到融合权重 FPN 模块中的 L_4 层，特征图 $\boldsymbol{F}$ 依次经过 L_4、L_3、L_2 和 L_1 卷积层卷积后，得到特征图 $\boldsymbol{P}_4$、$\boldsymbol{P}_3$、$\boldsymbol{P}_2$ 和 $\boldsymbol{P}_1$，对应的通道数为 C_4、C_3、C_2 和 C_1。将特征图 $\boldsymbol{P}_1$、$\boldsymbol{P}_2$、$\boldsymbol{P}_3$ 和 $\boldsymbol{P}_4$ 进行权重分配和特征融合后得到特征图 $\boldsymbol{D}_1$、$\boldsymbol{D}_2$、$\boldsymbol{D}_3$ 和 $\boldsymbol{D}_4$，并输出其中具有最多特征信息的特征图 $\boldsymbol{D}_1$，从而得到了具有丰富特征的特征图 $\boldsymbol{F}'$，更好地对潜在动态物体特征进行学习。经融合权重 FPN 模块获得特征图的计算方法如式(4-5)所示：

$$\begin{cases} \boldsymbol{D}_i = l_i \times f_i^{\text{upsample}}\left(\boldsymbol{D}_{i+1}\right) + f_i^{\text{conv}}\left(\boldsymbol{P}_i\right) & 1 \leqslant i \leqslant 3 \\ \boldsymbol{F}' = \boldsymbol{D}_1 \end{cases} \tag{4-5}$$

其中，l_i 表示当前层 i 的权重系数，f_i^{upsample} 表示当前层 i 的上采样因子，f_i^{conv} 表示当前层 i 的 1×1 卷积因子。

在融合权重 FPN 网络中，其深层位置的特征图 $\boldsymbol{P}_3$ 和 $\boldsymbol{P}_4$ 分别含有较为丰富的大物体特征信息和丰富的大物体特征信息，而浅层位置的特征图 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$ 分别含有较为丰富的小物体特征信息和丰富的小物体特征信息。本章通过网络层数关系，将权重系数建模成一个与当前层位置 i 相关的实体，因此不同的权重分配方

式将分别得到融合大物体的 LW-FPN 模块权重系数和融合小物体的 SW-FPN 模块权重系数，计算方法如式(4-6)所示：

$$l_i = \begin{cases} (i+1)^{-1} & \text{if 融合小物体特征} \\ i \times (M+1)^{-1} & \text{if 融合大物体特征} \end{cases} \tag{4-6}$$

其中，M 表示网络总层数。

4.2 潜在动态物体运动判断

视觉 SLAM 系统运行过程中易受动态物体的影响，若不能准确判断潜在动态物体运动状态，将会降低视觉 SLAM 系统鲁棒性。为降低动态物体对视觉 SLAM 系统的影响，本节提出了一种动态物体检测与剔除的方法。通过帧与帧之间的位姿矩阵将当前帧的潜在动态物体标记区域映射到下一帧，利用物体投影偏差计算并判断潜在动态物体运动状态。

将第 k-1 帧图像输入到 ITNET 实例分割网络会得到分割掩码集合{$\boldsymbol{MK}_1^{k-1}$，$\boldsymbol{MK}_2^{k-1}$，…，$\boldsymbol{MK}_{m1}^{k-1}$}和目标边界框集合{$\boldsymbol{BX}_1^{k-1}$，$\boldsymbol{BX}_2^{k-1}$，…，$\boldsymbol{BX}_{m1}^{k-1}$}，计算出目标边界框集合中每个目标边界框的中心点。在分割掩码映射环节，利用第 3 章剔除动态点后估计出的位姿矩阵将第 k-1 帧中第 i 个目标边界框 $\boldsymbol{BX}_i^{k-1}$（$1 \le i \le m1$）中心坐标映射到 k 帧，进而得到 $\boldsymbol{BX}_i^{k-1}$ 和 $\boldsymbol{MK}_i^{k-1}$ 在 k 帧中的位置。

为更准确获取物体运动状态，本节构建出一种物体投影偏差的方法。将目标边界框 $\boldsymbol{BX}_i^{k-1}$ 映射到 k 帧后，与对应的掩码 $\boldsymbol{MK}_i^{k-1}$ 进行位置对齐得到 $\boldsymbol{MK}_{i,\text{est}}^{k}$。在获取第 k 帧图像的分割掩码集合{$\boldsymbol{MK}_1^{k}$，$\boldsymbol{MK}_2^{k}$，…，$\boldsymbol{MK}_{m2}^{k}$}后，得到与第 k-1 帧掩码 $\boldsymbol{MK}_i^{k-1}$ 对应物体的分割掩码 $\boldsymbol{MK}_j^{k}$（$1 \le j \le m2$），并根据 $\boldsymbol{MK}_j^{k}$ 和 $\boldsymbol{MK}_{i,\text{est}}^{k}$ 标记区域之间的联系得到标记区域 $\boldsymbol{MK}_{i,\text{co}}^{k}$，$\boldsymbol{MK}_j^{k}$、$\boldsymbol{MK}_{i,\text{est}}^{k}$ 和 $\boldsymbol{MK}_{i,\text{co}}^{k}$ 三者关系如图 4-7 所示。最后利用 $\boldsymbol{MK}_j^{k}$、$\boldsymbol{MK}_{i,\text{est}}^{k}$ 和 $\boldsymbol{MK}_{i,\text{co}}^{k}$ 构建出物体投影偏差 e_i^{pro}，e_i^{pro} 计算方式如公式(4-7)所示：

$$e_i^{\text{pro}} = 1 - \frac{\text{A}(\boldsymbol{MK}_{i,\text{co}}^{k})}{\text{A}(\boldsymbol{MK}_j^{k}) + \text{A}(\boldsymbol{MK}_{i,\text{est}}^{k}) - \text{A}(\boldsymbol{MK}_{i,\text{co}}^{k})} \tag{4-7}$$

其中，A(•) 表示获取标记区域像素数量函数，$\boldsymbol{MK}_{i,\mathrm{co}}^{k} = \boldsymbol{MK}_{i,\mathrm{est}}^{k} \cap \boldsymbol{MK}_{j}^{k}$。

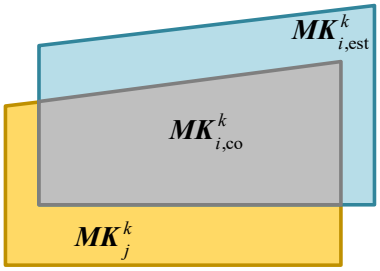


图 4-7 掩码标记区域关系示意图

e_i^{pro} 值计算反映了物体运动的可能性，e_i^{pro} 越大表示 $\boldsymbol{MK}_{j}^{k}$ 和 $\boldsymbol{MK}_{i,\mathrm{est}}^{k}$ 重合部分越少，表示是动态物体的可能性越大，反之则重合部分越多，表示动态物体可能性越小。本节构建出物体投影偏差将所有判定为动态物体的区域进行标记和剔除，减少动态物体对系统定位的影响。

4.3 实验结果与分析

实验分为四个部分: 各模块组合算法 AP（Average Precision）值对比、潜在动态物体分割算法、动态物体剔除和 SLAM 系统评估。实验所用平台软硬件配置与第 3 章 3.3 节相同，深度学习框架由开源的 PyTorch 实现。

4.3.1 各模块组合算法 AP 值对比

本章选择 MS-COCO 数据集对各个模块组合的算法进行实验。MS-COCO（2017）数据集包含了 11.8 万张图片的训练数据集（train2017）、0.5 万张图片的验证数据集（val2017）以及 4.1 万张图片的测试数据集，常用于目标检测以及实例分割等场景。在本章所述各模块组合算法中，主干网络模块分别采用了 ResNet-101-FPN、ReT-encoder（MSA）和 ReT-encoder，分割模块分别使用了 FPN 和 M-FPN。其中，ReT-encoder（MSA）表示将 ReT-encoder 层中的 RMSA 模块替换为 MSA 模块。在训练过程中，考虑到服务器配置，本章选取了训练数据集中的 4 万张图片进行训练，训练模型的批处理尺寸（batch size）设置为 16，epoch 设置为 64，学习率（learning rate）采用了 Adam 优化训练网络。主干网络模块为 ResNet-101-FPN 的初始学习率设置为 2×10^{-4}，主干网络模块为 ReT-encoder（MSA）和 ReT-encoder 的初始学习率设置为 1×10^{-3}。

本章选取测试数据集中的 1 万张图片，对 AP（Average Precision）、AP_S（AP for Small Objects）和 AP_L（AP for Large Objects）进行评估以验证本章 ITNET 实例分割网络的分割精度。不同组合算法 AP 值折线图如图 4-8 所示。其中，R101F+F 表示主干网络为 ResNet-101-FPN 且分割模块使用 FPN，R101F+MF 表示主干网络为 ResNet-101-FPN 且分割模块使用 M-FPN，RETM+F 表示主干网络为 ReT-encoder（MSA）且分割模块使用 FPN，RETM+MF 表示主干网络为 ReT-encoder（MSA）且分割模块为 M-FPN，RET+F 表示主干网络为 ReT-encoder 且分割模块使用 FPN，RET+MF 表示主干网络为 ReT-encoder 且分割模块使用 M-FPN。从图中数据可知，ReT-encoder 层和 M-FPN 模块能有效提高物体分割的 AP 值，因此本章设计的 ITNET 网络能更准确地分割潜在动态物体。

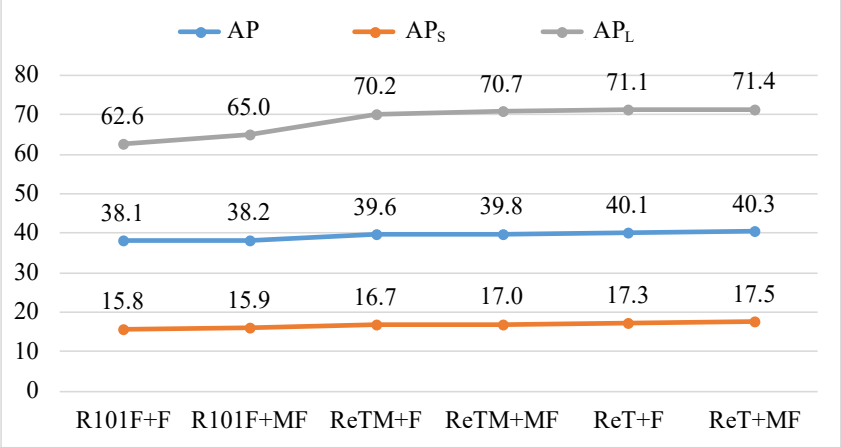


图 4-8 不同组合算法的 AP 值数据对比

4.3.2 潜在动态物体分割算法

本章在 TUM 数据集中选择所需的动态场景序列，用来验证 SOLO[54]、Yolact[55]和 ITNET 三种分割网络在动态场景下的实例分割性能，选取的序列与第 3 章 3.3 节使用的序列相同。三种算法实例分割效果图如图 4-9 所示，其中红色框为辅助标记框。图 4-9(a)(b)(c)为场景中存在大物体时的实例分割效果图，从图中可以看出：Yolact 算法和 SOLO 算法由于缺乏对视野占据面积较大区域的特征联系机制，因此对潜在动态物体中的大物体进行实例分割时无法完整分割大物体；ITNET 算法与前两种算法相比，其主干网络模块由于采用了多头注意力机制，加强了大物体的全局特征联系，并结合混合权重 FPN 模块得到大物体特征所在区域，更准确地引导模型标记出潜在动态物体的区域，从而能够正确完成实

例分割。图 4-9(d)(e)(f)为小物体的分割效果图，由于小物体在视野中占比面积小且特征相对较弱，SOLO 算法和 Yolact 算法没有加强对小物体的特征学习，难以高效学习小物体特征，因此无法完整分割小物体；ITNET 算法在主干模块中采用了区域注意力机制关注局部特征信息，提高了提取小物体特征信息的完整性和准确性，同时混合权重 FPN 模块加强了对小物体的区域特征的学习，从而提高了小物体实例分割的准确率。如图 4-9(f)所示，其小物体实例分割效果明显优于前两种算法，验证了 ITNET 实例分割网络在大物体分割和小物体分割方面有突出效果。

(a) Yolact 大物体分割　(b) SOLO 大物体分割　(c) ITNET 大物体分割

(d) Yolact 小物体分割　(e) SOLO 小物体分割　(f) ITNET 小物体分割

图 4-9 三种算法实例分割效果图

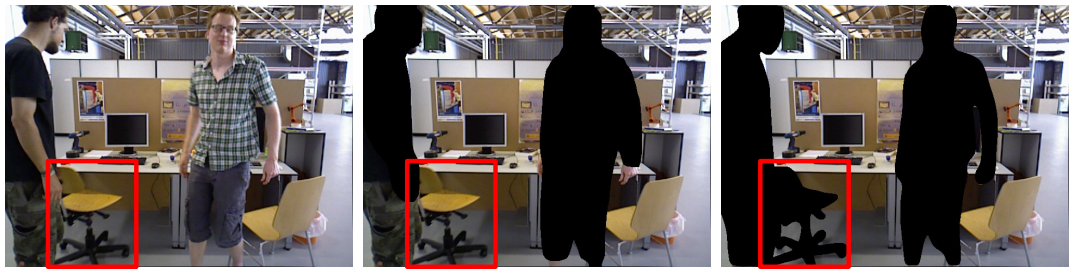
Yolact、SOLO 和本章 ITNET 三种实例分割算法的各类 AP 值对比如表 4-1 所示。从表中可知，ITNET 算法各类 AP 的平均值与 Yolact 算法相比提高了 43.22%，与 SOLO 算法相比提高了 17.15%。

表 4-1 三种实例分割算法 AP 值数据对比

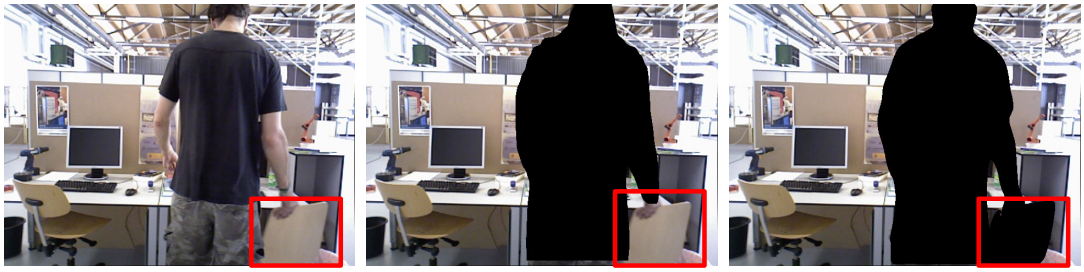
算法	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Yolact	31.5	50.4	32.3	12.0	33.4	47.0
SOLO	37.8	59.5	40.1	16.4	40.5	54.2
ITNET	40.3	61.7	42.3	17.5	60.2	71.4

4.3.3 动态物体剔除

本章在 TUM 数据集中选择所需的动态场景序列，用来验证比较算法动态物体剔除性能，选取的序列与第 3 章 3.3 节使用的序列相同。ORB-SLAM2、DynaSLAM 和本章算法在上述四个数据集中某一帧动态物体剔除效果对比图如图 4-10 所示，其中红色框为本章辅助标记框，剔除区域设置为黑色。由图 4-10 (a)(d)(g)(j)可知 ORB-SLAM2 算法显然无法准确剔除动态物体，这是因为 ORB-

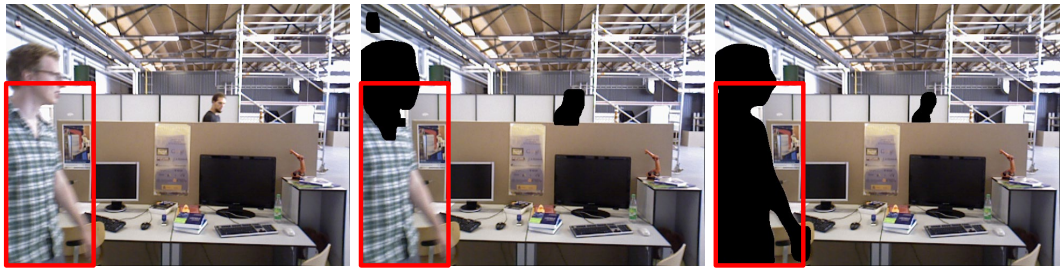


(a) W_static ORB-SLAM2　(b) W_static DynaSLAM　(c) W_static Ours



(d) W_xyz ORB-SLAM2　(e) W_xyz DynaSLAM　(f) W_xyz Ours

(g) W_half ORB-SLAM2　(h) W_half DynaSLAM　(i) W_half Ours



(j) W_rpy ORB-SLAM2　(k) W_rpy DynaSLAM　(l) W_rpy Ours

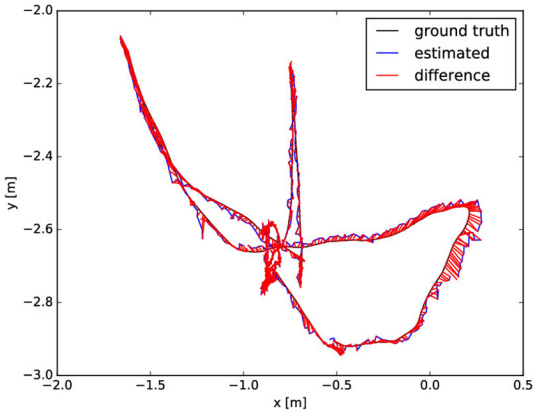
图 4-10 动态物体剔除效果图

SLAM2 算法中没有引入物体检测网络，因此难以约束动态物体对算法性能的影响。由图 4-10 (b)(e)(h)(k)可知，DynaSLAM 不能完整剔除动态物体，这是由于该算法使用的实例分割网络难以准确标记环境中潜在运动的大物体和小物体区域，在物体运动判断过程中难以准确判断例如图中椅子的运动状态，因此 DynaSLAM 难以有效降低动态物体对轨迹精度的影响。图4-10 (c)(f)(i)(l)为本章算法动态物体剔除后效果图，本章算法通过 ITNET 分割网络精确标记环境中潜在运动的大物体和小物体，在潜在动态物体运动判断过程中，通过引入物体投影偏差策略判断场景中潜在动态物体的运动状态，并及时剔除动态物体，从而降低动态物体对 SLAM 系统定位的影响。

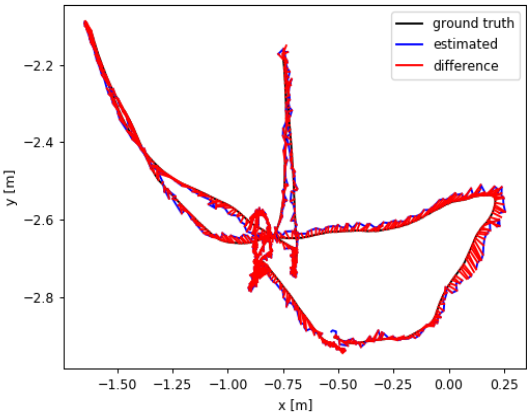
4.3.4 SLAM 系统评估

本章在 TUM 数据集中选择所需的动态场景序列，用来验证 DynaSLAM 和本章算法的构建轨迹性能，选取的序列与第 3 章 3.3 节使用的序列相同。DynaSLAM 和本章算法构建的轨迹图如图 4-11 所示。其中，图中的黑色线和蓝色线分别表示真实轨迹和算法运行出的轨迹，红色线表示算法运行的轨迹与真实轨迹的差异。本章算法与 DynaSLAM 算法相比轨迹构建误差更小，这是因为 DynaSLAM 算法实例分割网络效果较差，无法完整识别并标记动态场景中潜在运动的小物体和大物体，在动态物体判断环节，DynaSLAM 算法在相邻帧间进行位姿判断，难以判断移动位置较小的物体，因此难以准确剔除动态物体，降低了 DynaSLAM 算法定位精度。而本章算法由于引入了 ITNET 实例分割网络，能够识别并标记动态场景下潜在移动的大物体和小物体。在潜在动态物体运动判断环节，通过对物体投影偏差评估策略的构建准确计算出场景中潜在动态物体的运动状态，及时剔除动态物体，因此有效降低了动态物体对 SLAM 系统定位的影响。本章算法剔除了环境中的动态物体，减少动态物体的特征加入位姿估计策略中，因此本章算法构建的轨迹图更接近真实轨迹。

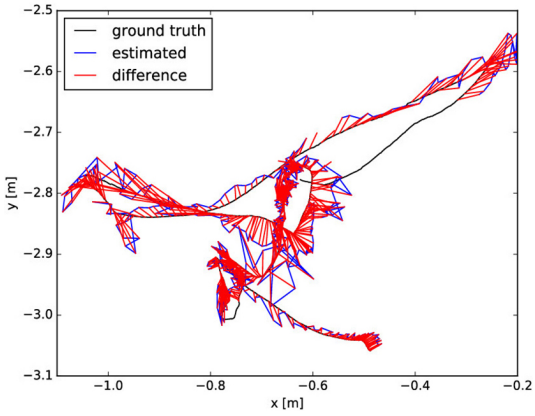
DynaSLAM 和本章算法的绝对轨迹均方根误差如图 4-12 所示，本章算法与 DynaSLAM 相比，绝对轨迹均方根误差平均降低了 15.88%，这是因为本章的 ITNET 分割精度较高，利用物体投影偏差判断并及时剔除动态物体，因此本章算法有效提高了动态场景下的定位精度。



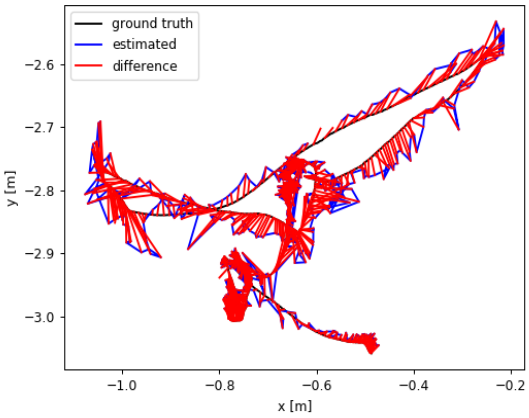
(a) DynaSLAM W_half



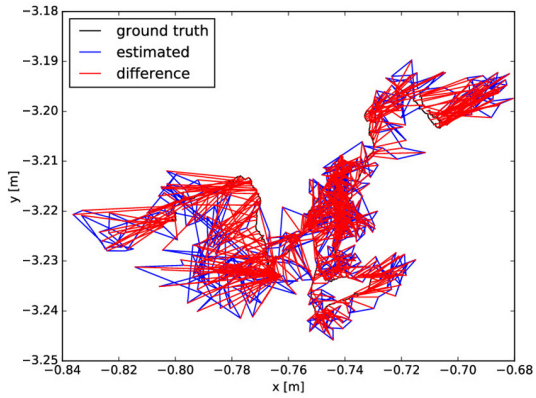
(b) Ours W_half



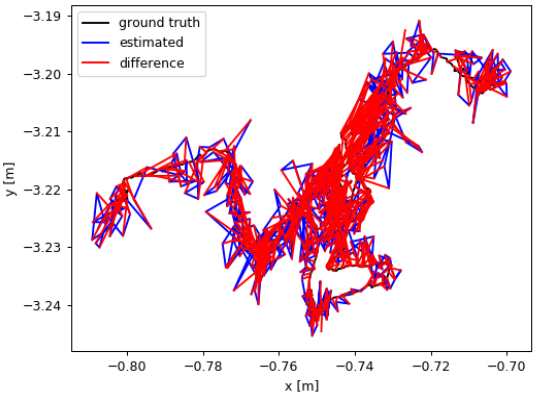
(c) DynaSLAM W_rpy



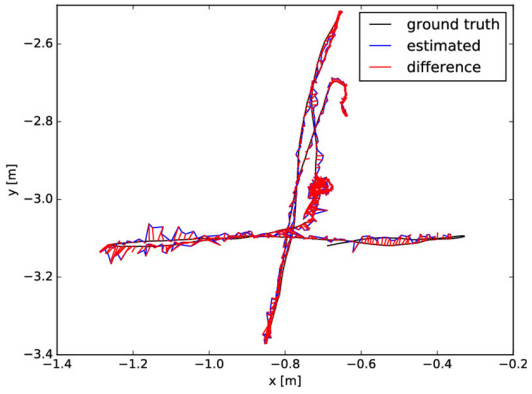
(d)Our W_rpy



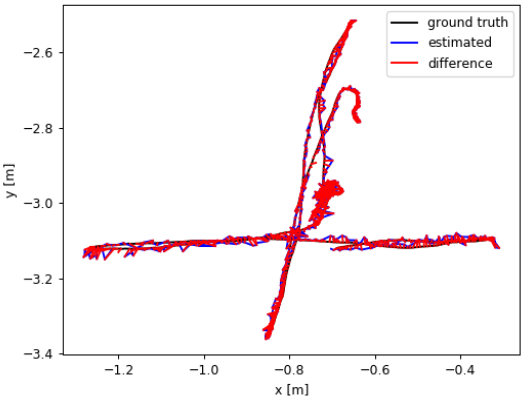
(e) DynaSLAM W_static



(f)Ours W_static



(g) DynaSLAM W_xyz



(h)Ours W_xyz

图 4-11 两种算法轨迹比较图

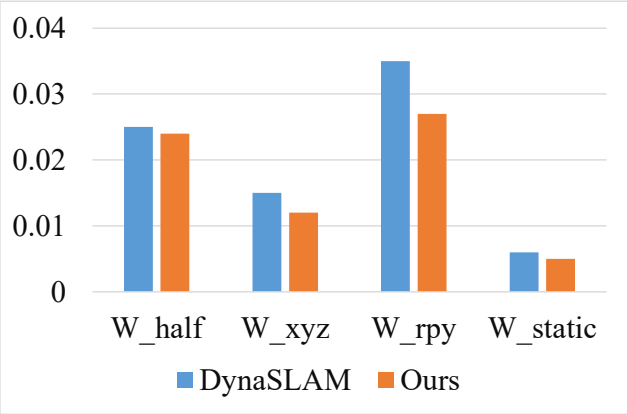


图 4-12 两种算法对轨迹均方根误差对比图

4.4 真实场景实验

在真实场景实验中，潜在动态物体的实例分割环节作为系统输入是由 PC 端离线生成。实验平台为 Husky 轮式移动机器人，外观如图 4-13(a)所示,其硬件配置和第 3 章 3.4 节真实场景使用的硬件配置相同。图 4-13(b)为真实实验环境。图 4-13(c)为真实场景的平面布局，其中浅蓝色部分为工作台，浅绿色部分为课桌，机器人从 A 开始，沿着标记方向的路线行进到达终点 B，C~D 段为行人往返线路。

(a) 轮式移动机器人　　(b) 真实实验环境　　(c) 真实场景布局

图 4-13 实验平台及实验真实环境

4.4.1 真实场景运行效果

本章实例分割算法对真实场景潜在物体分割效果如图 4-14 所示为，其中红色框为辅助标记框。由图中分割掩码对比可知本章提出的 ITNET 实例分割网络

在真实场景实验中能更准确地分割出动态场景下的小物体鼠标以及大物体行人。

图 4-14 潜在动态物体分割效果图

物体投影偏差评估策略剔除动态物体后的效果图如图 4-15 所示，其中红色框为本章所作辅助标记框，剔除区域设置为黑色。本章通过位姿矩阵映射掩码区域，利用掩码间的联系构建出物体投影偏差评估策略，及时筛选出真实场景中的动态物体。因此本章算法有效降低了动态物体对 SLAM 系统的影响，验证了本章算法在真实动态场景中的有效性。

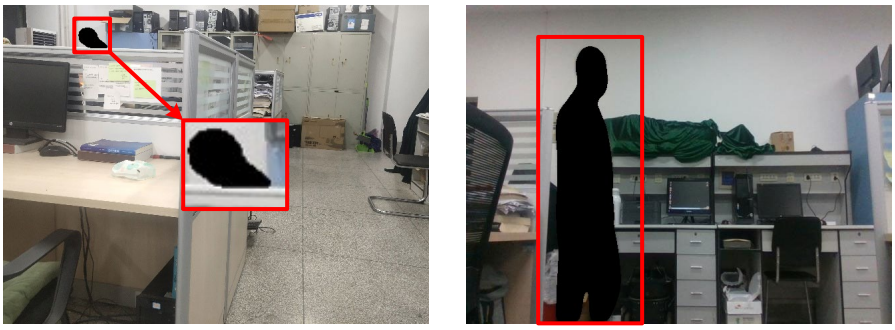


图 4-15 动态物体剔除效果图

4.4.2 真实场景运行轨迹图

本章算法和 ORB-SLAM2 算法运行于真实场景得到的轨迹如图 4-16 所示。在动态环境下，与 ORB-SLAM2 算法相比，本章算法能更接近真实轨迹，这是由于 ITNET 实例分割网络可以标记出动态场景中潜在的动态物体，并通过物体投影偏差评估策略判断物体运动状态，从而更精确地剔除动态物体，大幅减少了动态物体特征参与轨迹图的构建，有效提高了位姿估计的准确性，因此本章算法在动态环境下具有较高的稳定性。

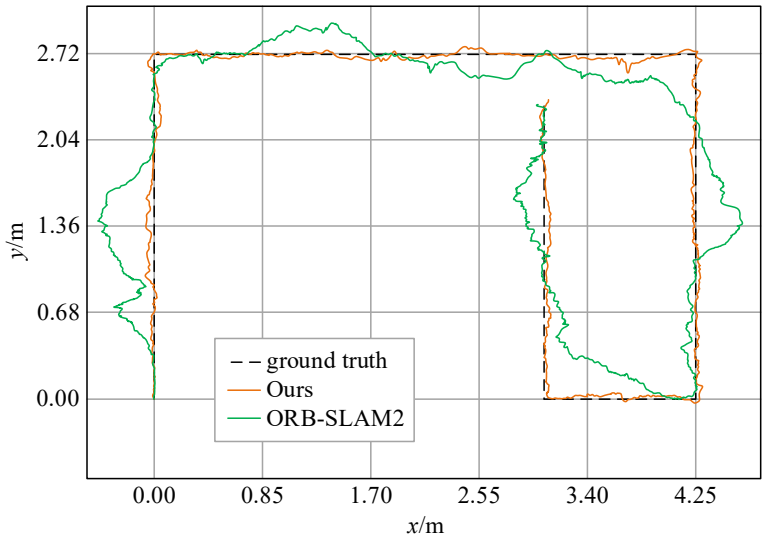


图 4-16 两种算法与真实轨迹比较图

4.5 本章小结

为了提高移动机器人在动态场景中运行的鲁棒性，本章提出了一种基于实例分割的动态视觉 SLAM 算法。在潜在动态物体实例分割环节，通过设计一种 ReT-encoder 模块，引导模型关注并提取图像局部特征信息和全局特征信息，同时设计一种混合权重 FPN 模块，通过权重系数融合学习大物体特征和小物体特征，精确标记潜在动态物体区域。在潜在动态物体运动状态环节，联合第 3 章算法剔除动态点后估计的位姿与本章实例分割网络输出的掩码构建物体投影偏差，从而及时剔除动态物体，降低动态物体对视觉 SLAM 系统的影响。本章所提算法改善了动态环境对视觉 SLAM 系统的影响，使得移动机器人能够在动态环境中进行同时定位与建图。通过引入区域多头注意力机制和混合权重 FPN 模块，改善了潜在动态大物体和小物体分割效果差的问题，提高了潜在动态物体分割的准确率。在潜在动态物体运动状态判断环节，采用物体投影偏差评估策略剔除动态物体，有效提高了视觉 SLAM 系统定位精度。本章算法使用公开数据集 TUM 和真实场景进行了验证，在 TUM 数据集中，本章算法与 DynaSLAM 相比，绝对轨迹均方根误差平均降低了 15.88%，在真实场景中，本章算法在位姿估计和定位方面有着较大优势。

第 5 章 全文总结与展望

5.1 全文总结

本文从目前视觉 SLAM 算法在动态场景下存在的缺陷出发，分析针对动态场景视觉 SLAM 算法的研究现状，对视觉 SLAM 的各个模块的基础工作原理进行介绍，依据基于 Transformer 分割网络物体分割精度，设计出一套动态场景下基于改进 Transformer 实例分割的视觉 SLAM 算法。在公开数据集和真实场景中进行算法性能验证，结果表明该方法有效改善了视觉 SLAM 系统在动态场景中的问题，提高了视觉 SLAM 定位和建图的精度。

论文主要研究内容与创新点概括如下：

（1）本文首先对视觉 SLAM 中的模块展开了研究，然后对 Transformer 基本框架进行简单介绍，并简要分析了一种基于 Transformer 实例分割网络。最后依据动态场景的需要建立模型用于提高视觉 SLAM 系统运行的稳定性，在动态场景下将基于 Transformer 实例分割的视觉 SLAM 算法与传统视觉 SLAM 算法 ORB-SLAM2 进行实验对比，实验结果表明，将基于 Transformer 实例分割模型与视觉 SLAM 系统相结合能有效减少动态点的数量。

（2）针对移动机器人在动态场景中易出现有效深度值特征点数量过少和难以高效剔除动态特征点等问题，影响相机位姿估计精度。为此，本文提出一种基于 D-ORB 特征点提取的动态视觉 SLAM 算法。通过设计一种 D-ORB 特征点提取方法，及时划分出有效区域，同时对有效区域进行关联和 ORB 特征点提取，提高有效深度值特征点数量。为进一步减少动态特征点对 SLAM 系统定位精度的影响，联合位姿估计和偏转视角剔除动态点，提高视觉 SLAM 系统在动态场景中的位姿精度。在公开数据集进行对比实验，结果表明本文算法的绝对轨迹均方根误差与 ORB-SLAM2 算法相比平均降低了 70.18%，相对位姿误差与 ORB-SLAM2 算法相比平均降低了 80.27%。在动态场景下，本文算法较传统算法 ORB-SLAM2 在轨迹构建方面优势明显。

（3）目前传统的视觉 SLAM 算法在动态场景下易出现难以完整标记潜在动态物体和无法准确判断物体运动状态等问题，为此，本文提出一种基于实例分割

的动态视觉 SLAM 算法。在潜在动态物体实例分割环节，通过设计一种 ReT-encoder 模块，引导模型关注并提取图像局部特征信息和全局特征信息，同时设计一种混合权重 FPN 模块，通过权重系数融合学习大物体特征和小物体特征，精确标记潜在动态物体区域。在潜在动态物体运动状态判断环节，联合第 3 章算法剔除动态点后估计的位姿与本文实例分割网络输出的掩码构建物体投影偏差模型获取并及时剔除动态物体。在公开数据集进行实例分割实验、动态物体剔除实验与系统性能评估实验，实验表明提出的 ITNET 算法各类 AP 的平均值与 Yolact 算法相比提高了 43.22%，与 SOLO 算法相比提高了 17.15%，绝对轨迹均方根误差与 DynaSLAM 相比平均降低了 15.88%。

5.2 研究展望

本文以动态场景下基于改进 Transformer 实例分割的视觉 SLAM 算法为主题进行研究，改进传统视觉 SLAM 算法得到基于 D-ORB 特征点提取的动态视觉 SLAM 算法和基于实例分割的动态视觉 SLAM 算法，通过公开数据集和真实场景验证本文算法，实验表明本文所提算法在动态场景中实现了较高的定位精度。但仍存在一些问题需要进一步探索。

（1）本文提出的算法有效提高了视觉 SLAM 系统在动态场景中的定位精度，但未对回环检测和建图进行改进，如何高效地将语义标签用于提高视觉 SLAM 系统在动态场景中回环检测准确率以及静态稠密点云地图构建精度是以后要进行的重要工作。

（2）本文采用基于 Transformer 实例分割网络对图像进行潜在动态物体分割，提高了物体分割精度，但该过程计算量复杂且消耗时间多，因此下一步将对算法进行改进，使得算法能够实时运行。

参考文献

[1] 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.

[2] Cebollada S, Payá L, Flores M, et al. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data[J]. Expert Systems with Applications, 2021, 167: 114195.

[3] 陈孟元, 陈何宝, 刘金辉等.动态场景下基于双重特征约束的视觉惯性 SLAM 算法[J].中国惯性技术学报, 2023, 31(04):381-389+400.

[4] 秦晓辉, 周洪, 廖毅霏等.动态环境下基于时序滑动窗口的鲁棒激光 SLAM 系统[J].湖南大学学报(自然科学版), 2023, 50(12):49-58.

[5] Zhu X, Yi J, Cheng J, et al. Adapted error map based mobile robot UWB indoor positioning[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69(9): 6336-6350.

[6] Al Khatib E I, Jaradat M A K, Abdel-Hafez M F. Low-cost reduced navigation system for mobile robot in indoor/outdoor environments[J]. IEEE Access, 2020, 8(3) : 25014-25026.

[7] Chou C C, Chou C F. Efficient and accurate tightly-coupled visual-lidar slam[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(9): 14509-14523.

[8] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]. IEEE International Conference on Robotics and Automation (ICRA 2016). IEEE, 2016: 1271-1278.

[9] Droeschel D, Behnke S. Efficient continuous-time SLAM for 3D lidar-based online map[C]. IEEE International Conference on Robotics and Automation (ICRA 2018). IEEE, 2018: 5000-5007.

[10] Macario Barros A, Michel M, Moline Y, et al. A comprehensive survey of visual slam algorithms[J]. Robotics, 2022, 11(1): 24-50.

[11] Fan Y, Zhang Q, Tang Y, et al. Blitz-SLAM: A semantic SLAM in dynamic environments[J]. Pattern Recognition, 2022, 121: 108225.

[12] 冯洲, 续欣莹, 郑宇轩等.动态场景下基于实例分割和三维重建的多物体单目 SLAM[J].仪器仪表学报, 2023, 44(08):51-62.

[13] Liu Y, Miura J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods[J]. Ieee Access, 2021, 9: 23772-23785.

[14] Ai Y, Rui T, Yang X, et al. Visual SLAM in dynamic environments based on object detection[J]. Defence Technology, 2021, 17(5): 1712-1721.

[15] 潘小鹃, 刘浩敏, 方铭等.基于语义概率预测的动态场景单目视觉 SLAM[J]. 中国图象图形学报, 2023, 28(07):2151-2166.

[16] 于雅楠, 卫红, 陈静.基于局部熵的 SLAM 视觉里程计优化算法[J].自动化学报, 2021, 47(06):1460-1466.

[17] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]. 2011 International Conference on Computer Vision (ICCV 2011). Barcelona, Spain, IEEE, 2011: 2320-2327.

[18] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]. Proceedings of the European Conference on Computer Vision (ECCV 2014). Berlin, Springer, IEEE, 2014: 834-849.

[19] Engel J, Koltun V, Cremers D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(3): 611-625.

[20] 贾嫣晗, 邹风山, 徐方等.完全在线的双目直接法视觉 SLAM 算法[J].控制与决策, 2023, 38(11):3093-3102.

[21] 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.

[22] Klein G, Murray D. Parallel Tracking and Mapping for small AR workspaces[C]. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007). Nara, Japan, IEEE, 2007: 225-234.

[23] 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.

[24] 付梦印, 吕宪伟, 刘彤等.基于 RGB-D 数据的实时 SLAM 算法[J].机器人, 2015, 37(06):683-692.

[25] 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.

[26] 张涛, 陈浩, 闫捷等.基于双向重投影的双目视觉里程计[J].中国惯性技术学

报, 2018, 26(06):752-759.

[27] Campos, Carlos, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics 37.6 (2021): 1874-1890.

[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(01):45-52.

[29] Zou D, Tan P. Coslam: Collaborative visual slam in dynamic environments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35(2): 354-366.

[30] Chen C S, Hung Y P, Cheng J B. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 21(11): 1229-1234.

[31] Dai W, Zhang Y, Li P, et al. Rgb-d slam in dynamic environments using point correlations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(1): 373-389.

[32] Kaneko M, Iwami K, Ogawa T, et al. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018). Piscataway, USA, IEEE, 2018: 371-378.

[33] Zhong F, Wang S, Zhang Z, et al. Detect-SLAM: Making object detection and SLAM mutually beneficial[C]. IEEE Winter Conference on Applications of Computer Vision (WACV 2018). Nevada, USA , IEEE, 2018: 1001-1010.

[34] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]. European Conference on Computer Vision (ECCV 2016). Amsterdam, Netherlands IEEE, 2016: 21-37.

[35] 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): 1-16.

[36] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018). Madrid, Spain, IEEE, 2018: 1168-1174.

[37] Zhou G, Liu W, Zhu Q, et al. ECA-mobilenetv3 (large)+ SegNet model for binary

sugarcane classification of remotely sensed images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-15.

[38] 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.

[39] Hameed K, Chai D, Rassau A. Score-based mask edge improvement of Mask-RCNN for segmentation of fruit and vegetables[J]. Expert Systems with Applications, 2022, 190: 116205.

[40] 姚二亮, 张合新, 宋海涛等.基于语义信息和边缘一致性的鲁棒 SLAM 算法[J].机器人, 2019, 41(06):751-760.

[41] Xu H, Guo M, Nedjah N, et al. Vehicle and pedestrian detection algorithm based on lightweight YOLOv3-promote and semi-precision acceleration[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(10): 19760-19771.

[42] 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: 1-16.

[43] Ma J, Jiang X, Jiang J, et al. LMR: Learning a two-class classifier for mismatch removal[J]. IEEE Transactions on Image Processing, 2019, 28(8): 4045-4059.

[44] Cvišić I, Marković I, Petrović I. Soft2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric[J]. IEEE Transactions on Robotics, 2022, 39(1): 273-288.

[45] Yin J, Luo D, Yan F, et al. A novel lidar-assisted monocular visual SLAM framework for mobile robots in outdoor environments[J]. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 1-11.

[46] Kuçak R A, Erol S, Erol B. The strip adjustment of mobile LiDAR point clouds using iterative closest point (ICP) algorithm[J]. Arabian Journal of Geosciences, 2022, 15(11): 1017-1018.

[47] Chauchat P, Barrau A, Bonnabel S. Factor graph-based smoothing without matrix inversion for highly precise localization[J]. IEEE Transactions on Control Systems Technology, 2020, 29(3): 1219-1232.

[48] Tanaka T, Sasagawa Y, Okatani T. Learning to bundle-adjust: A graph network approach to faster optimization of bundle adjustment for vehicular slam[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV

2021). Montreal, Canada, IEEE, 2021: 6250-6259.

[49] Memon A R, Wang H, Hussain A. Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems[J]. Robotics and Autonomous Systems, 2020, 126: 103470.

[50] He G, Yuan X, Zhuang Y, et al. An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 70: 1-9.

[51] Han K, Xiao A, Wu E, et al. Transformer in transformer[J]. Advances in Neural Information Processing Systems, 2021, 34: 15908-15919.

[52] Zhao H, Jiang L, Jia J, et al. Point transformer[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision(ICCV 2021). Montreal, Canada, IEEE, 2021: 16259-16268.

[53] 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(ICCV 2021). Montreal, Canada, IEEE, 2021: 10012-10022.

[54] Wang X, Kong T, Shen C, et al. Solo: Segmenting objects by locations[C]. European Conference on Computer Vision (ECCV). UK, August 23-28, 2020: 649-665.

[55] Bolya D, Zhou C, Xiao F, et al. Yolact: Real-time instance segmentation[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019). Seoul, South Korea, IEEE, 2019: 9157-9166.

攻读学位期间发表的学术论文和科研成果

一、发表学术论文

【1】陈孟元*,**钱润邦**,郭行荣,等.动态场景下基于实例分割与运动一致性约束的VSLAM 算法[J].中国惯性技术学报,2023,31(10):986-995.

【2】**Runbang Qian***，Hangrong Guo，Mengyuan Chen et al. A Visual SLAM Algorithm Based on Instance Segmentation and Background Inpainting in Dynamic Scenes[C]. 38th Youth Academic Annual Conference of Chinese Association of Automation (YAC 2023). IEEE, 2023: 132-136.

二、参与项目

【1】安徽省重点研究与开发计划项目（项目号：202304a05020073）（参与）。

【2】安徽省高校协同创新项目（项目号：GXXT-2021-050）(参与)。

【3】安徽省高校杰出青年科研项目（项目号：2022AH020065）（参与）。

致谢

在毕业论文完成之际，三年的研究生生活也即将迎来了尾声，在三年的学习和生活中收获了很多，在此，我要感谢在三年里给予我生活和学习巨大帮助的老师，同学，家人。

我要感谢我的导师陈孟元教授，感谢陈老师在三年来对我的帮助。在科研方面，陈老师以身作则与我们一起探讨论文创新点，同时指导我们实验验证创新点的可行性，在论文撰写过程中陈老师对我们有极大的耐心，教导我们论文写作的注意事项，更改论文中的错误，为我们解决了一个又一个难题。老师对待工作与学术有着严谨的态度，对待学习有着刻苦钻研的精神，这些为我们树立了榜样。

我要感谢我的同门郭行荣、龚光强、程浩和姚满宇，是他们在我学习困难的时候给予我帮助。

我要感谢我的家人，是你们对我无私的奉献才让我能够全身心投入到学习中，感谢你们默默无闻的付出。

最后感谢抽出宝贵时间在百忙之中评审论文的各位老师。

尚德教学 唯实惟新