DOI:10. 13382/j. jemi. B2508337

动态遮挡场景下一种改进 Oneformer 分割网络的 VSLAM 算法*

陈孟元[1,2] 陈 俊[1,2] 唐 哲[1,2] 范帅龙[1,2] 张坦坦[1,2] 冯峥嵘[1,2]
(1. 安徽工程大学电气工程学院 芜湖 241000;2. 高端装备先进感知与智能控制教育部重点实验室 芜湖 241000)

摘 要:针对传统同步定位与建图(SLAM)算法在动态遮挡场景中难以有效标记被遮挡物体、无法准确判断潜在物体的运动状态以及在动态物体剔除后造成特征点数量减少的问题,提出了一种改进 Oneformer 分割网络的视觉 SLAM 算法。该算法通过设计特征增强卷积、特征增强模块和遮挡关注模块,来增加被遮挡区域的关注度,并优化相对位置编码以提升被遮挡物体边界的语义准确性,从而实现对潜在动态物体的精确标记;使用相机位姿估计初步确定相机位置,再进行物体运动估计的方法进行物体的运动判断;采用最优近邻像素匹配策略,利用相邻帧中的静态信息来完成对动态区域的修复,进而提取修复后的特征点用于位姿估计。在公开数据集 TUM 及真实场景中进行了验证,与 DS-SLAM 和 DynaSLAM 算法相比,绝对轨迹误差的均方根误差均值分别降低了 84. 08%、22. 29%,表现出良好的轨迹精度。

关键词: VSLAM;动态遮挡场景;分割网络;运动估计;背景修复
中图分类号: TP242. 6;TN209 **文献标识码**: A **国家标准学科分类代码**: 510. 4050

Improved VSLAM algorithm for Oneformer segmentation networks in dynamic occlusion scenarios

Chen Mengyuan[1,2] Chen Jun[1,2] Tang Zhe[1,2] Fan Shuailong[1,2] Zhang Tantan[1,2] Feng Zhengrong[1,2]
(1. School of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China; 2. Key Laboratory of Advanced Perception and Intelligent Control for High-end Equipment, Ministry of Education, Wuhu 241000, China)

Abstract: To address the challenges faced by traditional simultaneous localization and mapping (SLAM) algorithms in dynamic occlusion scenarios—namely, the inability to effectively label occluded objects, accurately determine the motion state of potential objects, and the reduction in feature point count after dynamic object removal—this paper proposes an improved visual SLAM (VSLAM) algorithm based on the Oneformer segmentation network. This algorithm enhances attention to occluded regions by designing feature-enhancing convolutions, feature enhancement modules, and occlusion attention modules. It optimizes relative position encoding to improve semantic accuracy of occluded object boundaries, enabling precise marking of potential dynamic objects. Object motion is assessed by first determining the camera position via camera pose estimation, followed by object motion estimation. An optimal nearest-neighbor pixel matching strategy is employed to repair dynamic regions using static information from adjacent frames, enabling the extraction of repaired feature points for pose estimation. Validation on the TUM public dataset and real-world scenarios demonstrated superior trajectory accuracy. Compared to DS-SLAM and DynaSLAM algorithms, the mean root mean square error of absolute trajectory error decreased by 84. 08% and 22. 29%, demonstrated excellent trajectory accuracy.

Keywords: VSLAM; dynamic occlusion scene; segmentation network; motion estimation; background repair

收稿日期: 2025-04-25 Received Date: 2025-04-25
*基金项目:安徽省重点研究与开发计划项目(高新领域)(202304a05020073)、安徽省学术和技术带头人后备人选科研活动经费择优资助(2022H292)资助

0 引　言

同步定位与建图(simultaneous localization and mapping, SLAM)技术专注于使移动机器人在未知环境中实现自主导航与探索,行进过程中,该技术能够精确地构建环境的模型与地图[1-3]。当采用相机作为传感器时,这一技术便被称为视觉SLAM(visual SLAM, VSLAM)。若搭载的是激光传感器,则被称为激光SLAM,然而,激光传感器在信息量获取、体积等方面存在局限性,一定程度上制约了激光SLAM技术的广泛应用。相比之下,视觉SLAM凭借其小巧的体积、较低的成本以及更丰富的信息获取能力,逐渐成为SLAM领域的研究热点[4-6]。ORB-SLAM算法是目前主流的视觉SLAM算法,自诞生以来便经历了多次迭代与优化[7]。ORB-SLAM1最初是专为单目相机设计的,随后,Mur-artal等[8]推出了ORB-SLAM2算法,该版本在原有基础上实现了对双目立体视觉和RGB-D相机的兼容,通过特征点信息来精确计算物体的运动姿态。然而,在动态环境中,ORB-SLAM2难以避免地会受到动态物体产生的动态点的影响,这极易导致特征点的误匹配,进而造成相机位姿估计的偏差[9-10],影响后续的地图构建与定位精度。为了进一步提升系统的性能与鲁棒性,Campos等[11]推出了最新的ORB-SLAM3,该版本在ORB-SLAM2的基础上进行了诸多改进。尽管如此,ORB-SLAM3仍然面临着动态场景下由动态特征点引发的问题,尚无法完全解决这一挑战。为了应对以上问题,Su等[12]则开发了RTD-SLAM,这是一种基于YOLOv5s的系统,该系统结合语义信息和光流技术来识别与剔除动态特征点。Bescos等[13]研发了DynaSLAM算法,该算法以ORB-SLAM2为基础框架,并融入了多视图几何技术来辨识动态特征点,通过应用深度学习模型Mask R-CNN,DynaSLAM能够实现对动态目标的精准分割,有效地减轻了动态特征点对系统的影响。Kaneko等[14]提出的Mask-SLAM系统,利用DeepLabV2算法对图像进行分割,通过解析这些语义信息,能够剔除已知的动态物体。Zhong等[15]开发了Detect-SLAM系统,该系统基于目标检测网络SSD,能有效剔除动态特征点。冯一博等[16]提出了一种应用于动态场景的VSLAM算法,该算法基于YOLOv3s[17]模型。其工作原理是通过移除检测框内全部特征点来减轻动态物体对系统性能的干扰。Yu等[18]提出了DS-SLAM系统,该系统融合了SegNet[19]实时语义分割网络与运动一致性验证技术,旨在打造更加稳健的视觉SLAM系统。

综上所述,针对传统SLAM算法在动态遮挡场景中受动态物体的影响,造成特征误匹配、且难以对被遮挡的潜在动态物体完整地标记或准确分割,造成特征点误判断,以及在动态特征点去除后,造成整体的特征信息减少的问题,本文提出了一种在动态遮挡场景下基于改进Oneformer分割网络[20]的VSLAM算法。首先对Oneformer进行改进,通过设计的多方向特征增强卷积、双池化增强模块和遮挡关注模块对其架构进行改进,实现对潜在动态物体精确标记与分割。然后利用改进后的分割网络分割出的掩膜信息与物体运动估计得到的信息结合后,能准确识别并剔除场景中的动态物体。随后,使用一种改进的最优近邻像素匹配的背景修复算法进行动态区域的修复。最后,从修复后的区域中提取特征点,用于后续的位姿估计。在公开的TUM数据集以及实际场景中,本文对改进后的算法进行了验证,并与关算法进行了对比分析,展现出良好的位姿估计和建图能力。

1 系统总体框架

本文针对传统SLAM算法在动态遮挡场景下,被遮挡的潜在动态物体由于像素权重少,边界语义信息弱而难以被标记或无法准确分割,和其运动状态难以被准确判断,以及在其剔除动态特征点后,动态区域无特征点,造成整体的特征点减少,影响后续的位姿估计与建图。据此,本文提出了一种改进Oneformer分割网络的VSLAM算法。首先使用改进的分割网络实现对被遮挡潜在动态物体的精确分割,再结合相机位姿估计与物体运动估计的方法,准确地判断与剔除动态物体,最后针对动态特征点剔除后留下的空白区域,采用了一种改进的最优近邻像素匹配的背景修复算法,对动态区域进行修复。在动态区域修复完成后,从修复后的图像中提取特征点,并利用这些特征点进行后续的位姿估计与建图。改进的VSLAM算法的总体框架图如图1所示。

2 潜在动态物体的识别与分割

传统的分割算法对物体分割时,由于物体部分被遮挡,致使物体的权重降低,边界语义信息弱,从而影响分割的准确性和完整性,因此本文提出了一种基于Oneformer网络改进的分割网络,具体如下:

本文的算法通过设计多方向特征增强卷积,旨在解决动态物体在运动中可能呈现多种姿势状态或不规则部分被遮挡的情况;双池化增强模块的设计旨在提升网络对弱语义特征的捕捉能力,通过结合最大池化和平均池化等不同的池化策略,网络能够获取到更丰富的特征表达;遮挡关注模块的设计是为了更好地关注被遮挡区域,网络能够自适应地调整对不同区域的关注度,从而更准确地识别出被遮挡的物体。网络结构为:在主干构建特征金字塔网络的基础上,引入了残差结构以增强主干网

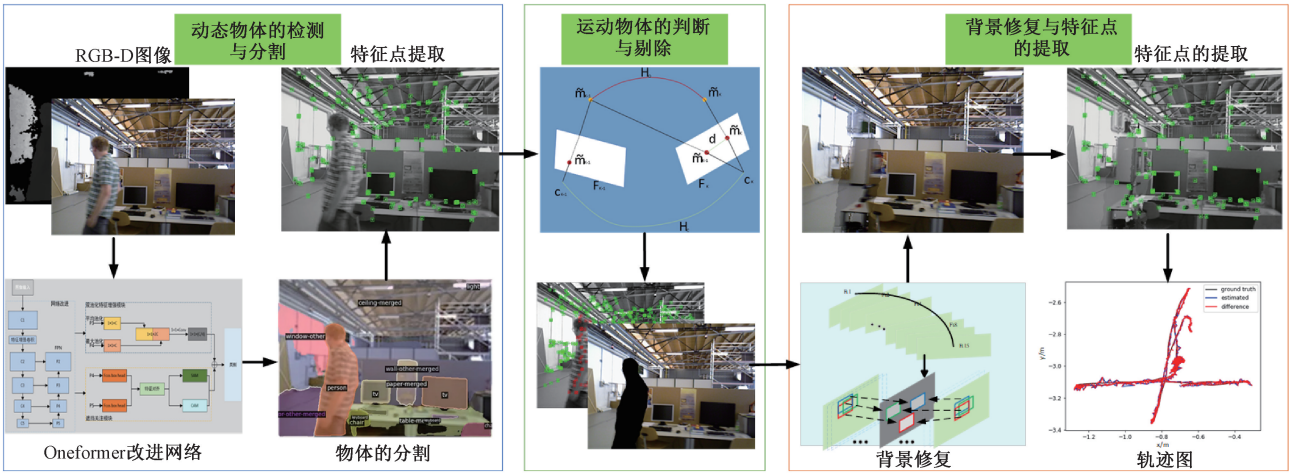


图 1　系统总体框架图

Fig. 1　Overall system framework diagram

络的性能,形成残差-特征金字塔网络。在主干网络中融入特征增强卷积,在网络的 Neck 部分对输入的 P3 和 P4 特征图层实施特征增强处理,旨在强化这些特征图中的弱语义信息。随后,将高维特征图 P4、P5 特征图送入遮挡关注模块,该模块能够增强动态物体区域的像素权重,从而提升对动态物体的识别准确性。分割网络结构图如图 2 所示。

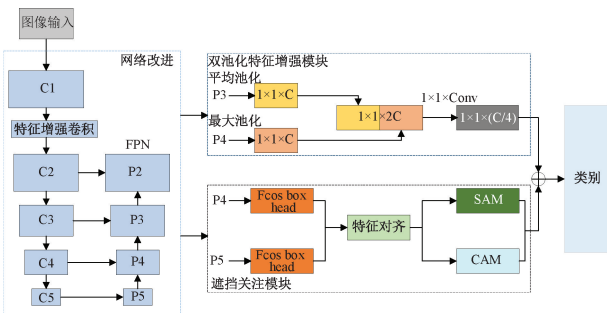


图 2　Oneformer 网络改进模块

Fig. 2　Oneformer network improvement module

2.1　多方向特征增强卷积

本文设计的多方向特征增强卷积是由 4 条不同方向的条形卷积搭配空洞卷积,所设计的多方向条形卷积如图 3 所示。方法步骤为:首先,采用 3×3 的深度卷积对输入数据进行初步处理,以精确捕捉其局部特征。这一步骤为后续的多方向条形卷积提供了必要的局部信息基础。其次,并行地应用 4 个方向(包括水平、垂直、反对角线及主对角线)的条形卷积,每个方向均搭配相应的空洞卷积。这一并行处理的方式使得网络能够同时从多个尺度和不同层级上获取丰富的特征映射,从而更加全面地理解输入数据的特征结构。最后,将来自这 4 个不同方向的条形卷积与空洞卷积的特征进行融合。这一融合过程不仅保留了各个方向上的关键特征信息,还生成了更为丰富和全面的多尺度特征表示。这种融合策略使得网络在关注局部细节的同时,也能够兼顾全局特征,从而提升了整体的特征提取能力。计算方式如式(1)和(2)所示。

$$M_{dfec} = f_{3\times3}^{dwc}(M_{in}) \oplus \sum_{i=1}^{4} Direction \tag{1}$$

$$M_{out} = M_{in} \oplus MLP(Norm(M_{dfec})) \tag{2}$$

其中,M_{in} 和 M_{out} 分别表示特征增强卷积模块的输入与输出,$f_{3\times3}^{dwc}$ 表示深度卷积,3×3 表示卷积核的大小,*Direction* 表示多方向卷积在不同方向上的卷积操作,*Norm* 表示对输入特征进行批归一化,*MLP* 则表示多层感知分类器,$\oplus$ 表示逐元素相加。

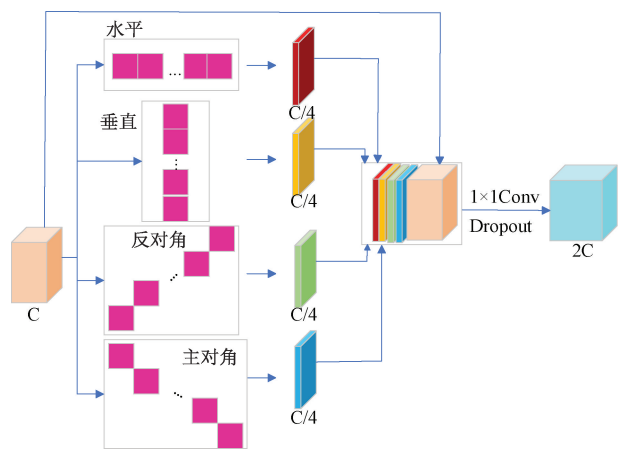


图 3　多方向条形卷积

Fig. 3　Multi-directional bar convolution

2.2　双池化增强模块

本文采用了多种尺度的平均池化和最大池化技术来

提取图像特征,随后,将这些多尺度特征进行融合。该模块能够全方位地捕捉并利用图像的局部细节与全局上下文信息。步骤为:当特征图 $\boldsymbol{D}$ 输入时,双池化增强模块首先会运用多种尺寸的池化核来捕捉图像的多尺度特征。针对每种尺寸的池化操作,设计了两个并行分支:一个分支执行平均池化,用于融合池化区域的信息;另一个分支则执行最大池化,以精准捕捉池化区域内的显著特征。接下来,将这两个分支产生的特征图在通道维度上进行合并,并依次通过批归一化 *BN* 层和 ReLU 激活函数,以增强特征的稳定性和非线性表达能力。随后,应用一个 1×1 卷积、*BN* 层和 ReLU 激活函数,对合并后的特征图进行进一步的特征提取和降维处理。之后,使用双线性插值方法将这些特征图上采样至与模块输入特征图相同的尺寸。再将模块输入特征图经过一个 1×1 卷积、*BN* 层和 ReLU 激活函数的处理,其通道数被压缩至原始输入通道数的四分之一。然后,将这个处理后的特征图与前面通过多尺度池化得到的特征图在通道维度上进行合并。最后,再次应用一个 1×1 卷积、*BN* 层和 ReLU 激活函数,以进一步融合这些多尺度特征,从而生成双池化增强模块的输出特征图。共采用了 3 种尺寸的池化操作:全局池化、4×4 核大小且步长为 4 的池化,以及 2×2 核大小且步长为 2 的池化,这些不同尺寸的池化操作能够捕捉不同尺度的特征信息。计算方式如式(3)~(5)所示,模块示意图如图 4 所示。

$$\boldsymbol{D}' = Up(f(\delta(BN([GAP(\boldsymbol{D});GMP(\boldsymbol{D})])))) \tag{3}$$

$$\boldsymbol{D}'_{(k,s)} = Up(f(\delta(BN([GAP_{(k,s)}(\boldsymbol{D});GMP_{(k,s)}(\boldsymbol{D})])))) \tag{4}$$

$$\boldsymbol{W} = f([f(\boldsymbol{D});\boldsymbol{D}';\boldsymbol{D}'_{(4,4)};\boldsymbol{D}'_{(2,2)}]) \tag{5}$$

其中,$GAP_{(k,s)}$ 和 $GMP_{(k,s)}$ 中的 k 表示池化核的大小,s 表示步长大小,*GAP* 和 *GMP* 分别表示平均池化和最大池化,*Up* 表示上采样,δ 表示 ReLU 激活函数,f 表示 1×1 卷积、*BN* 层和 ReLU 激活函数的处理操作,$\boldsymbol{D}'$ 和 $\boldsymbol{D}'_{(k,s)}$ 表示不同池化的结果,$\boldsymbol{W}$ 表示双池化增强模块输出的结果。

2.3　遮挡关注模块

通道注意力机制的功能在于为特征图 $\boldsymbol{Q}$ 中的各个通道分配相应的权重,它通过对特征图执行平均池化和最大池化操作,从而获取每个通道的特征信息,分别得到两组特征 $\boldsymbol{F}_{avg}$ 和 $\boldsymbol{F}_{MAX}$。随后,将这两组特征经过全连接层 FC 模块,以此增强通道间的关联性,生成富含特征信息的 $\boldsymbol{fv}$。最后用 $\boldsymbol{fv}$ 对输入特征图 $\boldsymbol{Q}$ 进行逐层通道加权得到 $\boldsymbol{Q}_v$。

空间注意力机制则旨在提升特征图中别遮挡区域像素值的权重。当特征图被输入至空间注意力机制时,会分别进行平均池化和最大池化,然后将这两组池化后的特征进行 concat 融合,形成新的特征图 $\boldsymbol{fc}$。接着,该特征图会通过一个 3×3×1 的卷积层以及 sigmoid 函数,从而得到权重分配空间注意图 $\boldsymbol{fu}$。这张特征图能够引导网络更加关注动态物体所在的区域。将 $\boldsymbol{fu}$ 与原始特征图 $\boldsymbol{Q}$ 连接得到经空间注意力加权后的特征图 $\boldsymbol{Q}_u$。经过此多注意力机制而获得的通道输出 $\boldsymbol{fv}$ 和空间输出 $\boldsymbol{fu}$ 计算方式分别如式(6)和(7)所示。示意图如图 5 所示。

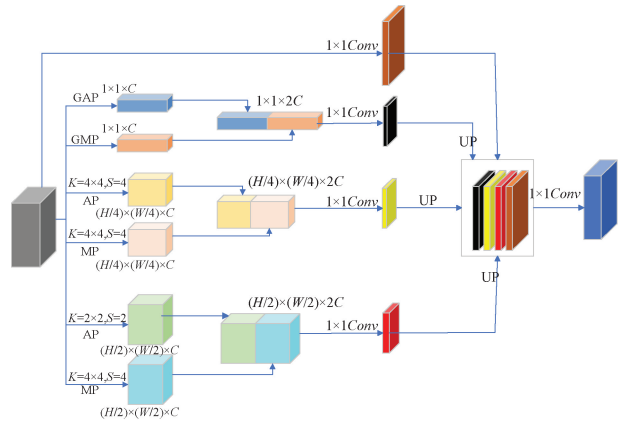


图 4　双池化增强模块

Fig. 4　Dual pooling enhancement module

$$\boldsymbol{fv} = sigmoid(P\eta(\boldsymbol{F}_{avg} + \boldsymbol{F}_{MAX})) \tag{6}$$

$$\boldsymbol{fu} = sigmoid(c(\boldsymbol{fc})) \tag{7}$$

其中,η 表示 ReLU 函数,P 为全连接层的参数,c 为 3×3×1 卷积网络。

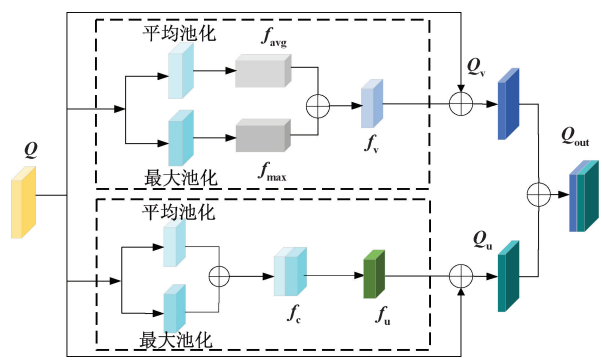


图 5　遮挡关注模块

Fig. 5　Masking attention module

3　潜在动态物体运动判断

在潜在动态物体的分割阶段,尽管能够初步识别出可能的动态元素,但缺乏有效的验证机制来确认这些候选物体是否真正发生了位移。针对这一问题,本文提出了一种结合相机位姿估计与物体运动状态判断的动态判定流程,旨在精确评估潜在动态物体的实时运动状态。

已知条件包括:一个空间静态点 p,以及当前帧 Y,该帧为关键帧 Y_{10} 之后的第 10 帧,则对于点 p 在关键帧

Y_{10} 和当前帧 Y 之间的重投影误差 $\boldsymbol{e}_p$ 可以表示为：

$$\boldsymbol{e}_p = \tau - \pi(\boldsymbol{I}_{10}, \boldsymbol{K}, \boldsymbol{L}(\boldsymbol{R} + \boldsymbol{n}_R, \boldsymbol{t} + \boldsymbol{n}_t)) \tag{8}$$

其中，τ 是特征点 p 投影到关键帧 Y_{10} 中 2D 像素坐标，π 是投影函数，$\boldsymbol{I}_{10}$ 是特征点 p 在关键帧 Y_{10} 上的 3D 坐标，$\boldsymbol{L}(\boldsymbol{R},\boldsymbol{t})$ 是从世界坐标系到相机坐标系的相对变换矩阵，$\boldsymbol{R}$ 是旋转矩阵，$\boldsymbol{t}$ 是平移矩阵，$\boldsymbol{K}$ 为相机内参矩阵，$\boldsymbol{n}_R$ 和 $\boldsymbol{n}_t$ 分别是旋转和平移的噪声。

为了确保重投影误差尽可能小，进而进行优化求解最小化重投影误差。定义一个代价函数 J，不仅考虑到所有特征点的重投影误差，还引入了正则化项 $\boldsymbol{R}(\boldsymbol{L})$ 来约束相机姿态 $\boldsymbol{L}$ 的变化，从而避免过拟合。同时，使用加权残差 $\boldsymbol{w}_i$ 来反映不同特征点的重要性，引入一个误差协方差矩阵 $\boldsymbol{\Sigma p}$ 来表示特征点位置的不确定性。假设存在先验信息 P_0 关于初始姿态的分布，可以采用贝叶斯框架下的最大后验概率估计来优化 $\boldsymbol{R}$ 和 $\boldsymbol{t}$。

$$J(\boldsymbol{R},\boldsymbol{t}) = argmin\left(\sum_{i=1}^{A} \boldsymbol{w}_i \boldsymbol{e}_p^{\mathrm{T}} \boldsymbol{\Sigma p}^{-1} \boldsymbol{e}_p + \lambda \boldsymbol{R}(\boldsymbol{L}) - logP_0(\boldsymbol{L}) \right) \tag{9}$$

其中，A 表示特征点的集合，λ 是正则化参数，用来平衡数据项和正则化项的影响。

为了更精确地判断一个特征点是否属于动态物体，除了视角偏差外，还可以考虑时间维度上的连续性，即特征点在多个连续帧间的运动一致性。对于每个特征点 p，它在一系列连续帧间的平均视角偏差 $\bar{\varphi}_p$ 的计算为：

$$\bar{\varphi}_p = \frac{1}{N} \sum_{j=1}^{N} \omega(\boldsymbol{k}\boldsymbol{m}_j, \boldsymbol{k}\ddot{\boldsymbol{m}}_j) + \frac{1}{N} \sum_{j=1}^{N} \omega(\boldsymbol{k}\ddot{\boldsymbol{m}}_{j+10}, \boldsymbol{k}\boldsymbol{m}_{j+10}) \tag{10}$$

其中，$\bar{\varphi}_p$ 表示特征点 p 在多帧间视角偏差的平均值，N 表示参与计算的特征点数量，ω 是衡量夹角的函数，$\boldsymbol{m}_j$ 和 $\ddot{\boldsymbol{m}}_j$ 分别是特征点及其投影点在第 j 帧世界坐标系统中的位置，$\boldsymbol{k}$ 代表了从世界坐标系到相机坐标系的变化矩阵，而下标 j+10 表示第 j+10 帧，$\boldsymbol{m}_{j+10}$ 和 $\ddot{\boldsymbol{m}}_{j+10}$ 分别表示特征点 p 及其投影点在第 j+10 帧中的位置。

在平均视角偏差 $\bar{\varphi}_p$ 求解后，与相应阈值 θ 比较，引入速度向量 $\boldsymbol{v}_p$ 和加速度向量 $\boldsymbol{a}_p$ 来表征特征点的运动状态。这样，可以得到一个决策规则：

$$f_p = \alpha \cdot s(\bar{\varphi}_p > \theta) + \beta \cdot s(|\boldsymbol{v}_p| > v_{th}) + \gamma \cdot s(|\boldsymbol{a}_p| > a_{th}) \tag{11}$$

其中，α,β,γ 是权重系数，s 是指示函数，v_{th} 和 a_{th} 分别是速度和加速度的阈值，f_{th} 是最终决策的阈值，如果 $f_p > f_{th}$，则判定为动态点，否则为静态点。

4　背景修复与特征优化

传统的动态 SLAM 算法在剔除动态物体后，面临特征信息减少的问题，这导致特征点提取数量不足、位姿估计不准确，进而影响回环检测和静态地图的构建。为解决此问题，本文使用了一种基于近似最近邻匹配改进的最优近邻像素匹配背景修复算法，具体分为以下两步。

步骤 1）信息估计

本文采取的做法是以目标帧为基准，沿箭头方向移动一个包含 15 帧的图像帧作为修复窗口。在该时间窗口内，对每一帧图像执行 ORB 特征提取与计数操作，记数为 $\sum Wi$。随后，将每个候选参考帧与目标帧进行特征点匹配，记录匹配数量为 $\sum Zi$。若 $\sum Wi$ 和 $\sum Zi$ 满足式（12）的条件，则判定该候选参考帧包含高效且丰富的图像信息。

$$\left| \sum W_i - \sum Z_i \right| < 60 \tag{12}$$

位姿是相邻帧之间联系的重要表达指标之一，采用对极几何来估计邻帧之间的联系。$\boldsymbol{P} = [x,y,z]^{\mathrm{T}}$ 为空间中的一点，在参考帧和目标帧中的投影分别为 $\boldsymbol{p}_1$ 和 $\boldsymbol{p}_2$，由针孔相机模型可得像素点 $\boldsymbol{p}_1$ 和 $\boldsymbol{p}_2$ 的像素位置为：

$$s_1\boldsymbol{p}_1 = \boldsymbol{V}\boldsymbol{P} \tag{13}$$

$$s_2\boldsymbol{p}_2 = \boldsymbol{V}(\boldsymbol{R}_{21}\boldsymbol{P} + \boldsymbol{t}_{21}) \tag{14}$$

其中，s_1 和 s_2 为尺度因子，$\boldsymbol{V}$ 为相机内参矩阵，$\boldsymbol{R}_{21}$ 和 $\boldsymbol{t}_{21}$ 分别为目标图相对于参考图的旋转和平移矩阵，再将三维向量投影到二维平面，继而得出任意两帧之间的位移变化量 Δp 和旋转变化量 $\Delta\theta$。

步骤 2）参考帧选取

具体步骤为：第一步取正逆向 15 帧的时间窗口作为一组，从该窗口内提取特征并进行匹配，以筛选出满足特定条件的候选参考帧。这些条件包括 Δp 超过阈值 τ 以及 $\Delta\theta$ 大于阈值 γ 时，将此帧作为候选参考帧。接下来，进一步筛选这些候选参考帧，若满足步骤 1）的条件，则将满足的候选参考帧加入参考帧库。参考帧选取方法的数学表达式如式（15）所示。

$$f = \left\{ fi \mid \begin{pmatrix} [\Delta p > \tau] \cup [\Delta\theta > \gamma] \\ \cup [|Wi - Zi| < 100] \end{pmatrix} \right\} \tag{15}$$

在初始化阶段，将待修复的目标图像设为 A 图，参考帧设为 B 图。随后，在 A 图中随机选取一个 3×3 像素块作为匹配块，并为其随机赋予一个偏移量，在 B 图中找到一个与 A 图中匹配块相对应的匹配块。在传播阶段，计算 A 图中匹配块与 B 图中匹配块的偏移量差异，并找出偏移量最小的值。在搜索阶段，针对 B 图中的每一个像素点，在当前匹配块为中心的同心圆内寻找一个更加匹配的偏移量，以替换当前的偏移量，搜索的初始半径设为图片的尺寸，然后以一半的速率逐渐减小半径，直至搜索结束。最后，将这些偏移量所对应的像素值赋予 A 图中对应的像素块。偏移量的具体计算公式如式（16）所示。

$$\boldsymbol{R} = \frac{\sum (\boldsymbol{A} - \boldsymbol{B})^2 + \sum (\bar{\boldsymbol{A}} - \bar{\boldsymbol{B}})^2}{n^2} \tag{16}$$

其中，$\boldsymbol{A}$、$\boldsymbol{B}$、$\bar{\boldsymbol{A}}$、$\bar{\boldsymbol{B}}$、n^2 分别为 A 图中的原始矩阵块，B

图中的原始矩阵块,A 图中的偏移矩阵块,B 图中的偏移矩阵块,图像的大小。经实验得知通常参考帧库由 2~4 帧图像构成,一般迭代次数为 3 次。

5　实验结果与分析

本文实验是在 Ubuntu 18.04 操作系统,CPU 为 Intel 13th i9-13900HX 2.20 GHz,GPU 为 8 G 显存的 NVIDIA GeForce RTX 4070 Laptop 上进行实验。在公开数据集 TUM 中进行验证实验,选取了多种算法和本文算法进行对比,其中 TUM 数据集是德国慕尼黑大学提出,是 SLAM 领域最常用的基准数据集之一,尤其适用于评估 SLAM 算法在室内环境下的定位精度与建图质量,其含有 39 种不同的测试序列,分为多种类别,以下使用的测试序列为其中的动态序列。

5.1　改进的 Oneformer 分割效果实验

为了评估本文改进的算法性能,实验在相同训练条件下,以原始 OneFormer 网络为基本模型,逐步引入特征增强卷积模块、双池化增强模块和遮挡关注模块进行消融实验,使用像素准确率(pixel accuary,PA)和交并比(intersection over union,IOU)来衡量效果,其中 PA 为正确归类的累计像素数量与测试集像素总数量比值。IOU 为该类预测覆盖区域与该类真实覆盖区域的重叠区域与合并区域的像素量比值,用来衡量预测与标签的重合度,IOU 值越大越好,mIOU 则为各类别 IOU 的平均。其结果如表 1 所示,其中改进算法在 w/xyz 动态数据集上测试的 PA 值达到了 83.7%,mIOU 值达到了 55.9%。

表 1　消融实验

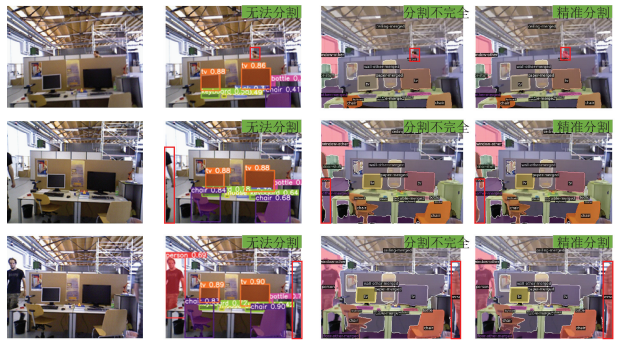
Table 1　Ablation experiments

特征增强卷积	双池化增强模块	遮挡关注模块	PA/%	mIOU/%
			80.6	53.1
√			81.3	54.5
	√		82.1	54.3
		√	81.7	53.5
√	√	√	83.7	55.9

为了验证本文改进的分割网络算法的分割效果,在 TUM 数据集的动态子集上,本文对 3 种不同情况上的遮挡进行了全面的验证,其中图中红色方框作为辅助标记。由图 6 中的分割效果对比可以得出,YOLOv8 与未改进的 Oneformer 分割精度不足,致使遮挡物体无法被精准分割,而本文所改进的算法对像素权重少和边界语义信息弱的潜在被遮挡物体的分割有着很好的效果。

5.2　潜在动态物体运动判断

本文选取 TUM 数据集中 3 种动态数据集进行物体



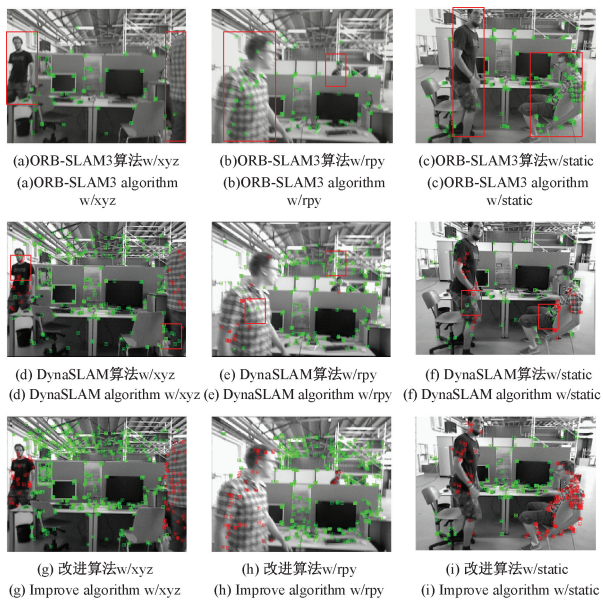
(a) 原图　(b) YOLOv8模型　(c) Oneformer模型　(d) 改进算法

(a) Original images　(b) YOLOv8 model　(c) Oneformer model　(d) Improve algorithm

图 6　算法分割效果验证

Fig. 6　Validation of algorithmic segmentation effects

的特征点的判断评估,如图 7 所示。可以看出 ORB-SLAM3 算法将动态物体和静态物体上的特征点都被判断为静态特征点(人物为动态目标,框内的点为动态物体上的静态特征点),DynaSLAM 的特征点的判断效果明显优于前者,但是也会存在误判断,而本文改进算法相比于 ORB-SLAM3 和 DynaSLAM 算法,其判断特征点的准确性更加精确。



(a)ORB-SLAM3算法w/xyz　(b)ORB-SLAM3算法w/rpy　(c)ORB-SLAM3算法w/static

(a)ORB-SLAM3 algorithm w/xyz　(b)ORB-SLAM3 algorithm w/rpy　(c)ORB-SLAM3 algorithm w/static

(d) DynaSLAM算法w/xyz　(e) DynaSLAM算法w/rpy　(f) DynaSLAM算法w/static

(d) DynaSLAM algorithm w/xyz　(e) DynaSLAM algorithm w/rpy　(f) DynaSLAM algorithm w/static

(g) 改进算法w/xyz　(h) 改进算法w/rpy　(i) 改进算法w/static

(g) Improve algorithm w/xyz　(h) Improve algorithm w/rpy　(i) Improve algorithm w/static

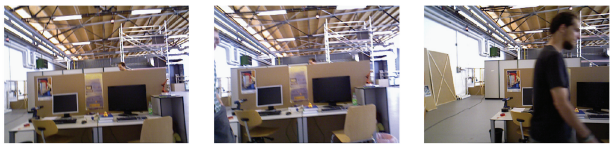
图 7　动态物体判断实验效果图

Fig. 7　Dynamic object judgment experiment effect diagram

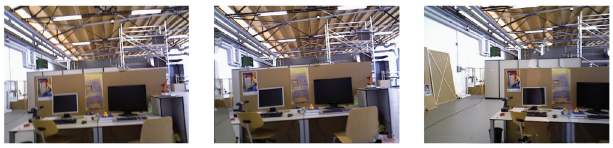
5.3　动态区域背景修复与特征提取

本文使用了一种基于近似最近邻匹配的最优近邻像素匹配背景修复算法,修复效果如图 8 所示。其中图 8(a)~(c)为包含动态人物的 RGB 原始图,图 8(d)~(f)为修复后的 RGB 图像,可以看出修复后的图像只包含场景中原始静态背景。选用 ORB-SLAM3 和

DynaSLAM 算法与本文算法进行特征点提取对比,其中 ORB-SLAM3 由于无法判断动态与静态物体,致使动态物体上的特征点无法去除。DynaSLAM 算法在剔除动态物体上的特征点后,导致用于位姿估计和地图构建的静态特征点数量减少。而本文改进算法在剔除动态区域,对剔除后的区域进行了背景修复,提取到更为丰富的静态特征点,进而能够构建出更准确的轨迹图。特征点提取效果如图 9 所示。



(a) 原始图像帧1 (a) Original image frame 1　(b) 原始图像帧2 (b) Original image frame 2　(c) 原始图像帧3 (c) Original image frame 3



(d) 图像修复帧1 (d) Image repair frame 1　(e) 图像修复帧2 (e) Image repair frame 2　(f) 图像修复帧3 (f) Image repair frame 3

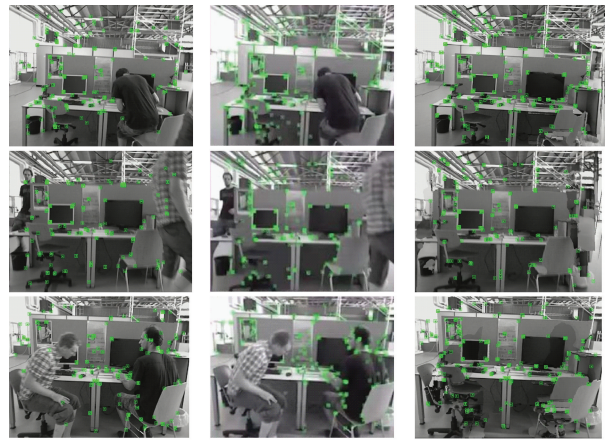
图 8　背景修复效果图

Fig. 8　Background restoration effect

5.4　SLAM 系统评估

本文进行了相对轨迹误差和绝对轨迹误差进行效果比较,选取 TUM 数据集中的 w/xyz、w/rpy、w/static 和 w/half 高动态子集以及 s/static、s/xyz 低动态子集进行本文算法效果验证,并使用 ORB-SLAM2 算法、ORB-SLAM3 算法、DS-SLAM 算法和 DynaSLAM 算法和本文算法进行对比。

本文使用均方根误差(root mean square error,RMSE)和标准差(standard deviation,S. D.)来衡量相对轨迹误差和绝对轨迹误差,从而验证本文算法的鲁棒性,其中数值越低则代表系统鲁棒性越高。根据表格数据,由于 ORB-SLAM2 和 ORB-SLAM3 算法无法判断动态与静态物体,



(a) ORB-SLAM3 特征提取 (a) ORB-SLAM3 feature extraction　(b) DynaSLAM 特征提取 (b) DynaSLAM feature extraction　(c) 改进算法特征提取 (c) Improve algorithm feature extraction

图 9　3 种不同算法的特征点提取

Fig. 9　Feature point extraction with three different algorithms

其鲁棒性效果很差。DynaSLAM 对潜在的动态物体上的特征点存在误判断,且提取的特征信息较少,进而影响了算法的定位精度。本文改进算法利用改进的 Oneformer 分割网络、相机位姿估计与物体运动状态判断结合的动态判断方式和背景修复,不仅更好的解决动态特征点带来的影响,而且对动态区域进行了修复,使得用于位姿估计的特征点数量也得到了提升。由表 2 和 3 数据可以得出本文算法相比于 ORB-SLAM2、ORB-SLAM3、DS-SLAM 和 DynaSLAM 算法,以 RMSE 值和 S. D. 值为一组,相对位姿误差分别平均减少了 92. 39% 和 94. 60%、89. 45% 和 92. 00%、20. 64% 和 45. 42%、58. 04% 和 69. 84%。绝对轨迹误差分别平均减少了 97. 88% 和 96. 76%、93. 03% 和 96. 85%、84. 08% 和 82. 77%、22. 29% 和 20. 39%。并选取 4 种高动态子集进行更加直观的轨迹对比实验,由生成的轨迹图可以看出本文改进算法的轨迹误差更小,更加贴近真实轨迹,如图 10 所示。

表 2　TUM 数据集下 5 种不同算法相对位姿误差对比结果(单位:m)

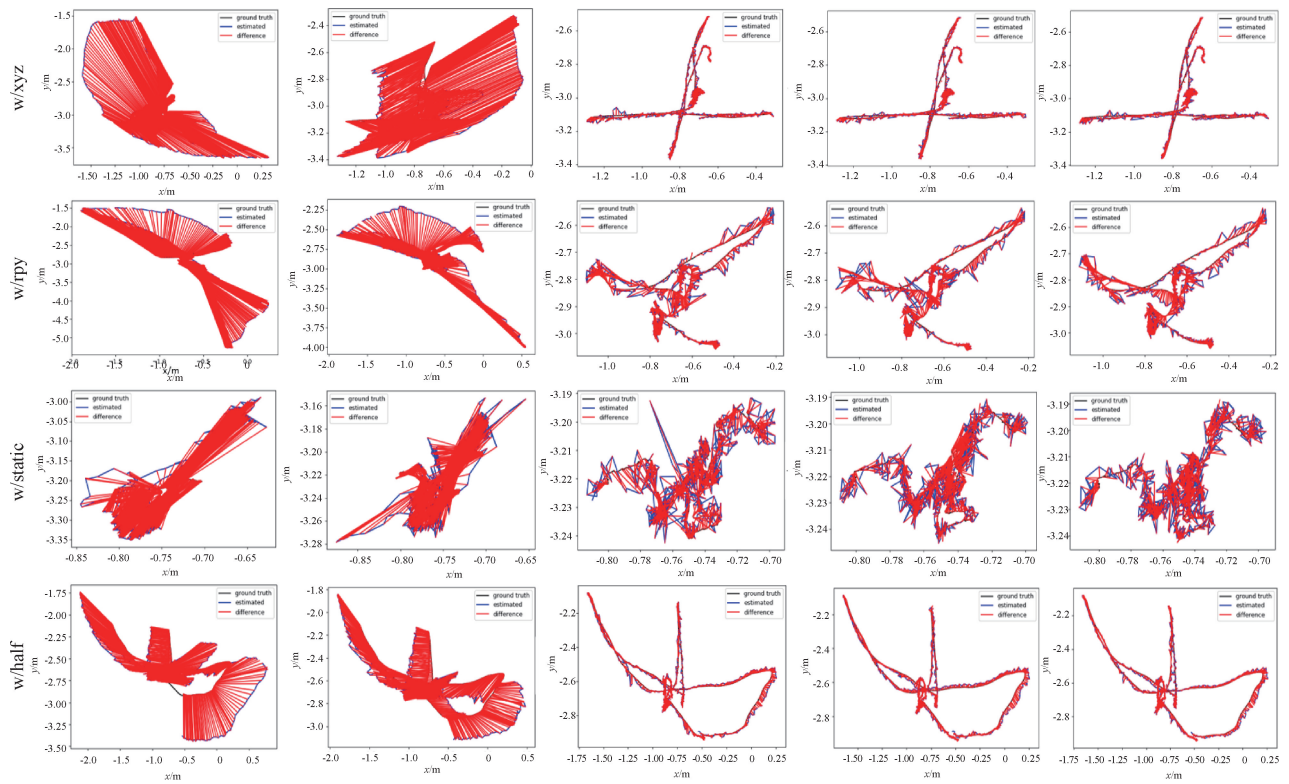
Table 2　Comparison results of relative position error of five different algorithms under TUM dataset (Unit: m)

Seq	ORB-SLAM2		ORB-SLAM3		DS-SLAM		DynaSLAM		本文算法	
	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.
w/xyz	0. 742 6	0. 450 2	0. 619 1	0. 323 9	0. 037 5	0. 023 3	0. 078 2	0. 045 7	0. 019 4	0. 027 7
w/rpy	0. 748 1	0. 428 9	0. 722 8	0. 523 3	0. 182 0	0. 118 1	0. 241 3	0. 115 6	0. 140 3	0. 032 5
w/static	0. 630 4	0. 398 1	0. 465 2	0. 210 5	0. 008 1	0. 004 6	0. 056 6	0. 031 3	0. 023 0	0. 010 3
w/half	0. 675 5	0. 433 0	0. 301 4	0. 096 2	0. 037 7	0. 014 8	0. 139 7	0. 107 2	0. 027 7	0. 014 3
s/static	0. 135 2	0. 008 1	0. 009 0	0. 002 6	0. 010 6	0. 004 2	0. 008 1	0. 005 1	0. 008 8	0. 004 5
s/xyz	0. 007 2	0. 003 8	0. 004 5	0. 004 1	0. 005 8	0. 005 5	0. 009 3	0. 003 2	0. 004 3	0. 003 9
average	**0. 489 8**	**0. 287 0**	**0. 353 7**	**0. 1934**	**0. 047 0**	**0. 028 4**	**0. 088 9**	**0. 051 4**	**0. 037 3**	**0. 015 5**

表3　TUM数据集下5种不同算法绝对轨迹误差对比结果(单位:m)

Table 3　Comparison results of absolute trajectory errors of five different algorithms under the TUM dataset (Unit:m)

Seq	ORB-SLAM2		ORB-SLAM3		DS-SLAM		DynaSLAM		本文算法	
	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.
w/xyz	0.923 3	0.515 7	0.418 4	0.142 3	0.025 2	0.015 0	0.017 5	0.008 4	0.013 5	0.007 1
w/rpy	1.892 0	0.468 5	0.569 2	0.250 4	0.434 1	0.235 4	0.036 8	0.028 6	0.026 0	0.016 3
w/static	0.367 2	0.135 5	0.125 9	1.041 9	0.006 3	0.004 7	0.006 9	0.003 1	0.007 5	0.003 3
w/half	0.654 0	0.388 1	0.035 7	0.112 5	0.025 4	0.017 1	0.024 9	0.013 3	0.020 3	0.010 4
s/static	0.009 1	0.005 6	0.012 5	0.006 4	0.008 5	0.004 5	0.007 9	0.003 7	0.008 0	0.006 1
s/xyz	0.008 8	0.004 7	0.009 5	0.008 9	0.012 6	0.009 0	0.011 2	0.004 6	0.006 3	0.005 9
average	**0.642 4**	**0.253 0**	**0.195 2**	**0.260 4**	**0.085 4**	**0.047 6**	**0.017 5**	**0.010 3**	**0.013 6**	**0.008 2**



(a) ORB-SLAM2算法　(b) ORB-SLAM3算法　(c) DS-SLAM算法　(d) DynaSLAM算法　(e) 改进算法

(a) ORB-SLAM2 algorithm　(b) ORB-SLAM3 algorithm　(c) DS-SLAM algorithm　(d) DynaSLAM algorithm　(e) Improve algorithm

图10　5种不同算法的轨迹对比

Fig. 10　Trajectory comparison of five different algorithms

5.5　真实场景中算法运行评估

为了验证本文算法的有效性和适用性,使用Husky轮式机器人,此平台相关配置为CPU是i7-10875 H处理器,内存为8 GB;GPU为GTX1080,操作系统为Ubuntu18.04,在真实场景中录取数据集。

本文选取系统生成的真实数据集中高遮挡与低遮挡俩组场景进行效果验证,实验效果图如图11所示。由图11(b)可以看出本文改进的分割网络可以对遮挡物体(运动的人)进行精准的识别与分割,由图11(c)可以看出本文算法的运动判断与背景修复及特征点提取有着良好的效果。轨迹对比图如图12所示,可以看出相比于ORB-SLAM3的2号轨迹,本文算法的1号轨迹更加贴近真实的3号轨迹,因此本文算法在动态遮挡环境下具有较高的鲁棒性。

6　结　论

为提高移动机器人在动态遮挡场景中鲁棒性,本文提出了一种基于Oneformer分割网络改进的VSLAM算法,该算法具有以下优点:1)针对现有实例分割网络在具

(a) 原始图
(a) Original images

(b) 潜在动态物体分割
(b) Potential dynamic objectsegmentation

(c) 动态背景修复与特征点提取
(c) Dynamic background restorationand feature point extraction

图 11　真实场景验证

Fig. 11　Real scenario validation

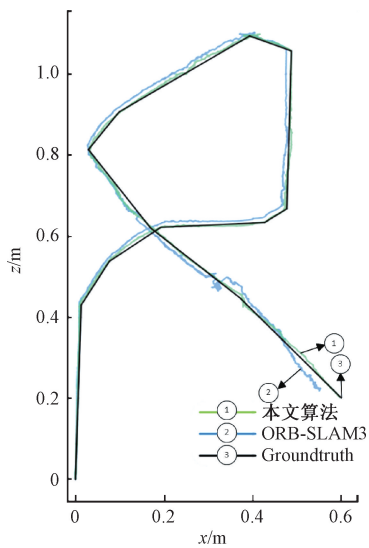


图 12　真实场景轨迹图对比

Fig. 12　Comparison of real scene trajectory map

有遮挡场景下难以分割或分割不完全被遮挡物体,提出一种融合特征增强卷积、双池化增强模块和遮挡关注模块的 Oneformer 分割网络,提高被遮挡动态物体的分割能力。2)引入相机位姿估计与物体运动状态判断结合的动态判断方式解决了潜在动态物体运动状态判断问题。3)采用一种基于近似最近邻匹配的最优近邻像素匹配背景修复算法进行动态背景修复。使用公开数据集 TUM 和自制数据集对本文所提算法进行了验证,并与多种算法进行对比,在定位精度方面有较大优势,并体现出了良好的构图能力。由于视觉 SLAM 的适用范围偏向室内,且精度易受光照强度影响,所以在未来,将会利用多传感器融合来解决此类问题,提高在不同复杂场景下的算法精度和鲁棒性。

参考文献

[1] 刘丰宇,程向红,曹毅. 基于深度学习与特征点速度约束的室内动态 SLAM 方法[J]. 中国惯性技术学报, 2023, 31(5):438-443.
LIU F Y, CHENG X H, CAO Y. An indoor dynamic SLAM method based on deep learning and feature point velocity constraints [J]. Journal of Chinese Inertial Technology, 2023, 31(5):438-443.

[2] 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, 2022, 50(3):608-618.
CHEN M Y, DING L M, ZHANG Y K. Fast PL-SLAM algorithm based on improved keyframe extraction strategy[J]. Acta Electronica Sinica, 2022, 50(3):608-618.

[3] 高兴波,史旭华,葛群峰,等. 面向动态物体场景的视觉 SLAM 综述[J]. 机器人,2021, 43(6):733-750.
GAO X B, SHI X H, GE Q F, et al. A survey of visual SLAM for scenes with dynamic objects [J]. Robot, 2021, 43(6):733-750.

[4] 冉宁,范晨锋,张少康,等. 一种改进 ORB 特征点提取与匹配的图像处理算法[J]. 电子测量与仪器学报, 2025,39(4):213-224.
RAN N, FAN CH F, ZHANG SH K, et al. An image processing algorithm for improved ORB feature point extraction and matching [J]. Journal of Electronic Measurement and Instrumentation, 2025, 39(4): 213-224.

[5] 栾添添,吕奉坤,班喜程,等. 高动态环境下的傅里叶梅林变换视觉 SLAM 算法[J]. 仪器仪表学报,2023, 44(7):242-251.
LUAN T T, LYU F K, BAN X CH, et al. Fourier-merlin transform visual SLAM algorithm for highly dynamic environments [J]. Chinese Journal of Scientific Instrument,2023,44(7):242-251.

[6] 马哲伟,周福强,王少红. 昏暗环境下自适应 ORB-SLAM2 算法研究[J]. 电子测量技术,2024,47(6): 94-99.
MA ZH W, ZHOU F Q, WANG SH H. Research on adaptive ORB-SLAM2 algorithm in dim environment[J]. Electronic Measurement Technology, 2024, 47(6): 94-99.

[7] 王柯赛,姚锡凡,黄宇,等. 动态环境下的视觉 S-LAM 研究评述[J]. 机器人, 2021,43(6):715-732.
WANG K S, YAO X F, HUANG Y, et al. A review of visual SLAM research in dynamic environments [J]. Robot, 2021,43(6):715-732.

[8] MUR-ARTAL R, TARDÓS J D. ORB-SLAM2: An opensource SLAM system for monocular, stereo, and RGB-D cameras[J]. IEEE Transactions on Robotics, 2017,33(5):1255-1262.

[9] RAN T, YUAN L, ZHANG J, et al. RS-SLAM: A robust semantic SLAM in dynamic environments based on RGB-D sensor[J]. IEEE Sensors Journal, 2021, 21(18): 20657-20664.

[10] 刘剑锋,孙力帆,普杰信,等. 基于刚性约束的双移动机器人协同定位[J]. 电子学报, 2020, 48(9): 1777-1785.
LIU J F, SUN L F, PU J X, et al. Cooperative localization in a team of two mobile robots based on rigid constraints [J]. Acta Electronica Sinica, 2020,48(9): 1777-1785.

[11] 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.

[12] SU P, LUO S, HUANG X. Real-time dynamic SLAM algorithm based on deep learning[J]. IEEE Access, 2022, 10: 87754-87766.

[13] 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.

[14] 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 Workshops, 2018: 258-266.

[15] ZHONG F, WANG S, ZHANG Z, et al. Detect-S-LAM: Making object detection and SLAM mutually beneficial[C]. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018: 1001-1010.

[16] 冯一博,张小俊,王金刚. 适用于室内动态场景的视觉SLAM算法研究[J]. 燕山大学学报, 2022, 46(4): 319-326.
FENG Y B, ZHANG X J, WANG J G. Research on visual SLAM algorithm for indoor dynamic scenes[J]. Journal of Yanshan University, 2022,46(4): 319-326.

[17] CHUN L Z, DIAN L, ZHI J Y, et al. YOLOv3: Face detection in complex environments [J]. International Journal of Computational Intelligence Systems, 2020, 13(1): 1153-1160.

[18] YU C, LIU Z X, LIU X J, et al. DS-SLAM: A se-mantic visual SLAM towards dynamic environments[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1168-1174.

[19] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[20] JAIN J, LI J, CHIU M T, et al. Oneformer: One transformer to rule universal image segmentation[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 2989-2998.

作者简介

陈孟元(通信作者),2007 年于安徽工程大学获得学士学位,2011 年于安徽工程大学获得硕士学位,2019 年于中国科学技术大学获得博士学位,现为安徽工程大学教授,硕士生导师,主要研究方向为移动机器人 SLAM。

E-mail: mychen@ahpu.edu.cn

Chen Mengyuan (Corresponding author) received his B. Sc. degree from Anhui University of Technology in 2007, his M. Sc. degree from the same institution in 2011, and his Ph. D. from the University of Science and Technology of China in 2019. He is now a professor and master's advisor at Anhui University of Technology. His main research interest includes SLAM for mobile robots.

陈俊,2022 年于合肥师范学院获得学士学位,现为安徽工程大学硕士研究生,主要研究 SLAM 视觉方向。

E-mail: 1446167891@qq.com

Chen Jun received a B. Sc. degree from Hefei Normal University in 2022. He is now a M. Sc. candidate at Anhui University of Technology. His main research interest includes visual SLAM research.