

DOI:10.19651/j.cnki.emt.2416549

针对室内高低动态环境的视觉 SLAM 算法研究^{*}符强^{1,2} 曾凡治^{1,2} 纪元法^{1,2,3} 任风华^{1,2}

(1. 桂林电子科技大学广西精密导航技术与应用重点实验室 桂林 541004; 2. 桂林电子科技大学信息与通信学院 桂林 541004; 3. 时空信息与智能位置服务国际联合实验室 桂林 541004)

摘要: 针对大多数经典视觉 SLAM 在室内的动态环境下鲁棒性不足问题,在基于 ORB-SLAM3 算法框架之下,提出了一种可区分室内高低动态环境的视觉 SLAM。首先提出一种根据连续多帧之间位姿变换做重投影误差来区分室内环境中的先验动态对象处于高动态还是低动态的算法。然后根据环境的高低动态决定是否结合 YOLOv8-Seg 实例分割网络对动态环境中的动态特征进行剔除,保证 SLAM 系统的跟踪精度。最后针对动态特征引起地图中出现重复性的地图点,在局部地图跟踪加入一种重复地图点消除算法,对动态环境中出现的重复地图点进行删除,进一步保证系统的稳定跟踪。在公开数据集 TUM RGB-D 上实验结果表明,改进后的算法相对于 ORB-SLAM3 算法在定位精度上均有提升,低动态环境下最大提升 60.41%,高动态环境下最大提升 94.65%。与其他动态特征去除算法相比,在大部分序列上实现了更高的定位精度,且在实时性上也更具优势。在所提算法有效解决 SLAM 应对室内动态环境的问题,提升了 SLAM 的定位精度。

关键词: 同时定位与建图;ORB-SLAM3;YOLOv8-Seg;高低动态环境;重复地图点消除
中图分类号: TP391.9;TN98 **文献标识码:** A **国家标准学科分类代码:** 510.99

Research on visual SLAM algorithm for indoor high and low dynamic environments

Fu Qiang^{1,2} Zeng Fanzhi^{1,2} Ji Yuanfa^{1,2,3} Ren Fenghua^{1,2}

(1. Guangxi Key Laboratory of Precision Navigation Technology and Application, Guilin University of Electronic Technology, Guilin 541004, China; 2. School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China; 3. International Joint Laboratory of Spatiotemporal Information and Intelligent Location Services, Guilin 541004, China)

Abstract: Aiming at the problem that most classic visual SLAMs are not robust enough in indoor dynamic environments, a visual SLAM that can distinguish between high and low dynamic environments is proposed based on the ORB-SLAM3 algorithm framework. First, an algorithm is proposed to distinguish whether the prior dynamic objects in indoor environments are in high or low dynamics based on the reprojection error of the pose transformation between multiple consecutive frames. Then, according to the high and low dynamics of the environment, it is decided whether to combine the YOLOv8-Seg instance segmentation network to remove the dynamic features in the dynamic environment to ensure the tracking accuracy of the SLAM system. Finally, in order to deal with the repeated map points in the map caused by dynamic features, a repeated map point elimination algorithm is added to the local map tracking to delete the repeated map points in the dynamic environment, further ensuring the stable tracking of the system. Experimental results on the public dataset TUM RGB-D show that the improved algorithm has improved the positioning accuracy compared with the ORB-SLAM3 algorithm, with a maximum improvement of 60.41% in low dynamic environments and a maximum improvement of 94.65% in high dynamic environments. Compared with other dynamic feature removal algorithms, higher positioning accuracy is achieved in most sequences, and it is also more advantageous in real-time performance. The proposed algorithm effectively solves the problem of SLAM coping with indoor dynamic environments and improves the positioning accuracy of SLAM.

Keywords: simultaneous localization and mapping; ORB-SLAM3; YOLOv8-Seg; high and low dynamic environment; duplicate map point elimination

0 引言

同时定位与建图(simultaneous localization and

mapping, SLAM)是自动驾驶、机器人、VR等应用中一项基础技术^[1]。许多较为经典SLAM算法基本都是假设环境是静态的,并且只考虑刚性、不可移动的物体。然而,在

收稿日期:2024-07-29

* 基金项目:国家自然科学基金(U23A20280, 62161007, 62061010)、广西科技厅项目(桂科AD22080061, 桂科AA23062038, 桂科AB23026120)、广西精密导航与应用重点实验室基金(DH202308)项目资助

动态场景下,SLAM 的姿态估计和定位依然面临一系列重要的挑战,这些挑战影响了其鲁棒性和准确性^[2]。

近年来,越来越多的 SLAM(同步定位与地图构建)研究人员开始尝试将深度学习的技术集成到 SLAM 系统中^[3]。在动态场景中,语义视觉 SLAM 逐渐成为研究热点。该方法通过利用图像的语义信息来辅助去除动态特征点,从而提高 SLAM 系统的鲁棒性和准确性^[4]。图像语义分割算法能够提供像素级别的语义分类结果,这些结果可以用来解析图像中每个像素的先验属性。因此,通过对场景中各类物体进行语义分类,可以有效区分出潜在的动态物体。例如,场景中的人物通常具有较高的运动概率,而建筑物则相对静止。

目前,随着深度学习技术的快速发展,结合深度学习的 SLAM 方法层出不穷,并涌现出许多优秀的研究成果。DynaSLAM^[5]是一个基于 ORB-SLAM2^[6]基础上加入实例分割 Msak R-CNN^[7] 卷积神经网络的 SLAM 系统。DynaSLAM 通过 Msak R-CNN 计算先验动态对逐像素语义分割,然后加入多视图几何方法,两个方法的结合系统有了更加精确的运动分割,使得系统的跟踪和建图的精度进一步的提升。同样,由 Yu 等^[8]提出的 DS-SLAM 采用 SegNet^[9]网络做语义分割任务,该系统加入运动一致性检测,将不满足对极约束的特征点,即动态点,如果这些动态点落入语义分割结果的掩膜内,说明检测到了动态目标,再对动态目标上的特征点进行剔除提高跟踪精度。由 Zhong 等^[10]提出的 Detect-SLAM 同样是基于 ORB-SLAM2 所构建,他们将 SSD^[11]加入到视觉 SLAM 中,将关键点分为四种状态:高置信度静态、低置信度静态、低置信度动态和高置信度动态。为考虑检测延迟和连续帧的时空一致性,他们仅在关键帧的彩色图像上用 SSD 检测,并在跟踪线程中逐帧传播概率。获得检测结果后,将关键帧插入局部地图并更新其移动概率,随后更新与关键帧匹配的局部地图中的 3D 点的移动概率。Chang 等^[12]引入实时实例分割网络 YOLACT^[13],然后利用几何约束和自适应密集光流法检测动态目标,并对动态目标特征进行去除。RDS-SLAM 由 Liu 等^[14]提出,该算法框架是基于 ORB-SLAM3^[15]之上,加入语义线程和语义优化线程可实现鲁棒跟踪和建图。该算法还应用了移动概率更新和传播语义信息,将这些信息存储在地图中,并利用数据关联算法排除跟踪中的异常值。

上面所提针对动态环境结合深度学习的 SLAM 算法均是利用深度学习检测图像的物体,并对物体分类,再由潜在动态对象所引起的动态特征进行处理或是在地图中对这些动态特征带来具有移动概率的地图点进行处理。这些算法在原有的经典 SLAM 算法基础上加入深度学习来应对动态环境,在低动态环境运行时,定位精度上并不一定优于原算法。这类算法没有考虑做场景运动状况的区分,当场景中可以发生高动态的目标(例如人类)处于低动态(例如人

保持坐着,动作幅度很小)时,且在这种状态保持较长时间时,如果将这类目标上的特征都剔除就会少了许多特征匹配的所需要的特征,从而使得跟踪的精度受到影响。

针对动态环境,经典 SLAM 算法鲁棒性不足问题以及主流结合深度学习算法未考虑到做运动状况区分而过度剔除低动态运动物体特征问题。本文提出基于 ORB-SLAM3 框架下,一种可以区分室内先验动态对象高低动态的视觉 SLAM 算法。该算法可实现区分环境的高低动态,由动态状况决定是否决定加入深度学习进行动态特征处理,还加入一种重复地图点消除算法处理局部跟踪中因动态特征带来的重复地图点问题。

1 系统框架

本文的 SLAM 系统如图 1 所示,在原 ORB-SLAM3 的系统框架下增加了两个线程,所以整体系统共有 5 个线程,分别为:区分高低动态线程、实例分割线程、跟踪线程、局部建图线程、回环检测线程。在图 1 中蓝色部分是在原 ORB-SLAM3 算法上增加或更改的部分。

区分高低动态线程根据多帧的传入图像信息检测场景中是处于高动态,还是低动态,为后续运动物体上的特征点决定是否剔除提供一个前提。

实例分割线程采用的是 YOLOv8n-Seg 轻量级实例分割网络,根据该网络不仅可以实现对输入的 RGB 图像进行语义分割,还可以进行目标检测。因此,可以对图像出现各类物体进行语义分类,同时进行目标检测框选出各类物体,如图 2 所示。实例分割线程的加入是为实现先验动态对象的识别,以及为动态特征点剔除给予动态特征剔除对象。

在跟踪线程中加入动态特征点剔除和在局部地图跟踪部分加入重复地图点消除。根据区分高低动态标志,在场景高动态时针对高动态对象所带来动态特征,使用动态特征点剔除可减少系统跟踪过程中因动态特征而受到的影响。重复地图点消除是为解决未使用动态特征剔除(低动态下)或未被剔除的动态特征点引起的在局部地图跟踪过程中出现可能影响跟踪精度重复性地图点。

1.1 场景高低动态区分

对于视觉 SLAM 来说,图像序列中的运动物体相对于背景来说的变化是不相同的。在多帧输入时,会显现出运动物体上的特征和静态特征运动方向和运动速度上会不一致^[16]。因此对于如何区分环境的高低动态,主要考虑的因素是运动物体的特征和静态特征的运动变化的不一致,变化较大可认为高动态,反之可认为低动态。

区分高低动态环境分为 3 步。

步骤 1)当前帧通过 LK(Lucas-Kanade)光流法^[17]跟踪上一帧。LK 光流法是通过关联单个像素来捕捉两个连续帧之间的运动动态。由灰度不变假设, $I(x, y, t)$ 表示 t 时刻灰度值,由 t 到 $t + dt$ 时刻可建立式(1):

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (1)$$

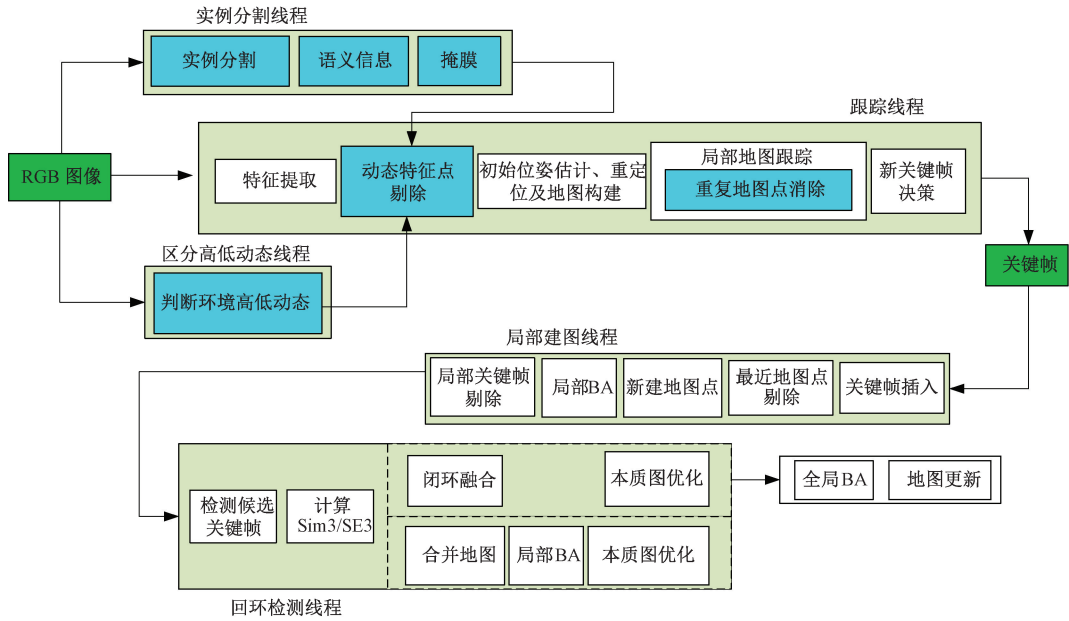


图 1 系统框图

Fig. 1 System architecture

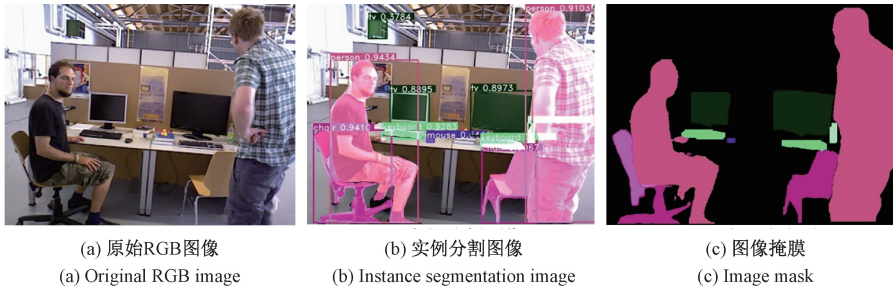


图 2 YOLOv8n-Seg 实例分割结果

Fig. 2 YOLOv8n-Seg instance segmentation results

对式(1)泰勒展开保留一阶项得到式(2):

$$I(x + dx, y + dy, t + dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (2)$$

灰度不变假设, 式(2)右边第二项为 0。所以有式(3):

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} = -\frac{\partial I}{\partial t} \quad (3)$$

其中, $\frac{dx}{dt}$ 和 $\frac{dy}{dt}$ 分别是像素 x 轴速度及 y 轴速度用 u 、 v 表示, $\frac{\partial I}{\partial x}$ 和 $\frac{\partial I}{\partial y}$ 分别是 x 和 y 的方向梯度用 I_x 、 I_y 表示,

式(3)可改写为式(4):

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \quad (4)$$

式(4)是针对单个像素点, 将整个窗口考虑进来后就可以得到式(5):

$$\begin{bmatrix} I_x & I_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk}, k = 1, \dots, \omega^2 \quad (5)$$

其中, ω 表示窗口大小, 该方程可以使用最小二乘解出运动速度 u 和 v 。 t 为离散时刻时, 可以估计多个图像中像素块的位置。通过多次迭代局部有效的像素梯度方程, 可以实现像素跟踪。

光流法跟踪上一帧特征点后, 使用 RANSAC(随机抽样一致)算法^[18]来消除离群特征点, 并计算最优的单一性, 再由单应矩阵分解得到的帧间位姿变换矩阵 $\mathbf{T}_{j,j-1} \in SE(2)$, 最后使用位姿变换矩阵做两帧匹配特征点的重投影误差, 流程如图 3 所示。重投影误差如式(6):

$$e_i = \|\mathbf{T}_{j,j-1} u_{j-1,i} - u_{j,i}\| \quad (6)$$

其中, $u_{j,i}$ 表示当前图像帧的特征点像素位置, e_i 表示重投影误差, j 表示当前帧。

步骤 2) 每帧中的特征点是否为动态点由重投影误差大于阈值做决定如式(7):

$$H_i = \begin{cases} 1, & e_i \geq a \\ 0, & \text{其他} \end{cases} \quad (7)$$

其中, a 表示一个阈值, 重投误差大于 a 时表示该特征点为动态点, H_i 表示特征点是否为动态点的标志。本文设置 a 为 1.7 个像素。

当一帧中的动态点数目占比该帧总特征点百分比到达一定阈值就可以判断为该帧属于动态的, 如式 (8):

$$HF_j = \begin{cases} 1, & \frac{\sum_i H_i}{W_j} > b \\ 0, & \text{其他} \end{cases} \quad (8)$$

其中, b 表示动态阈值, W_j 表示该帧的所有特征点, HF_j 表示该帧是否属于动态的。本文设置 b 为 10%。

步骤 3) 单靠前后两帧是很难判断为高动态的, 本文采取了输入连续多帧, 当这些帧中属于 HF_j 等于 1 的数量超过输入帧数量一半的时候则判断当前帧属于高动态, 本文设置判断连续 10 帧。因为需要多帧输入才可判断当前状态是否未为高动态, 因此, 程序一开始时要对区分高低动态线程初始化, 最开始 10 帧的状态根据第 10 帧判断为全高动态, 或者全为低动态。



图 3 计算帧间重投影误差

Fig. 3 Calculating inter-frame reprojection error

1.2 动态特征点剔除

YOLOv8 是由 Ultralytics 在 2023 年推出的, YOLOv8-Seg 是 YOLOv8 的实例分割模型。YOLOv8-Seg 包含两个分割头, 用于预测输入图像的语义分割掩膜。在保持高效的同时, YOLOv8-Seg 在目标检测和语义分割基准方面可实现较为先进的性能^[19]。

在实例分割线程中, 使用轻量级模型 YOLOv8n-Seg 对 RGB 图像进行目标检测和语义分割。为了减少处理时间, 程序中使用了 TensorRT^[20], 一种 GPU 加速方法来加速程序的处理速度。本文主要针对的是室内环境, 所以在场景中主要的先验动态对象是人类。因此, 可以根据语义标签选定只包含人物的掩膜如图 4 所示。



图 4 根据语义标签所选择的掩膜

Fig. 4 Masks selected based on semantic labels

动态特征点剔除流程如图 5 所示, 跟踪线程中动态特征点剔除是由区分高低动态线和实例分割线程共同决定的, 判断当前帧是否为高动态, 再决定使用实例分割结果进行人物上的特征点剔除。这样当避免人物处于低动态情况下时, 人物上的特征点不会被剔除。保证后续的匹配跟踪有足够多的特征点, 相对于在低动态情况下完全剔别人物上的特征点, 区分动态情况后再做处理跟踪会更加稳定。

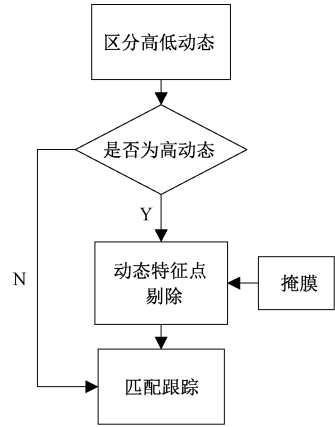


图 5 动态特征剔除流程

Fig. 5 Dynamic feature elimination process

1.3 局部跟踪中的重复地图点消除

SLAM 系统在运行过程中, 每帧图像的特征点一般会有与之对应的地图点。场景中运动物体上的特征点, 系统在特征匹配和跟踪时可能会在地图留下重复的地图点。在低动态场景、非所选语义物体运动 (例如椅子因人导致的间接运动) 以及掩膜边缘提取的特征点都可能留下重复地图点。因此本文借鉴了 Ni 等^[21]所提地图点消除算法。

在跟踪线程运行到局部跟踪时, 先获取当前帧的局部关键帧, 再由局部关键帧获取到局部地图点。在 ORB-SLAM3 算法系统中每个地图点是有与之对应的描述子。ORB-SLAM3 在特征匹配时会使用 DBoW2^[22] 词袋加速搜索匹配, 同样获取到当前帧地图点和局部地图点的描述子后, 可以使用 DBoW2 加速匹配当前地图点和局部地图点。当前帧地图点描述子和局部地图点描述子都属于同一节点 (Nodeld) 时, 根据特征向量的索引找同一个 Nodeld 的地图点特征描述子, 计算当前帧地图点描述子和局部地图点描述子之间的距离, 即计算它们的汉明相关距离, 计算距离如式 (9):

$$dist(cd, ld_k) = \|cd - ld_k\| \quad (9)$$

其中, cd 表示当前帧地图点描述子, ld_k 表示局部地图点描述子。根据距离公式可计算当前帧的每个地图点的描述子和局部地图点描述子距离, 根据这些距离集合去除大于 0 且小于距离集合中位数的距离, 这样可以消除动态特征带来重复性的地图点。重复地图点消除流程如图 6 所示。

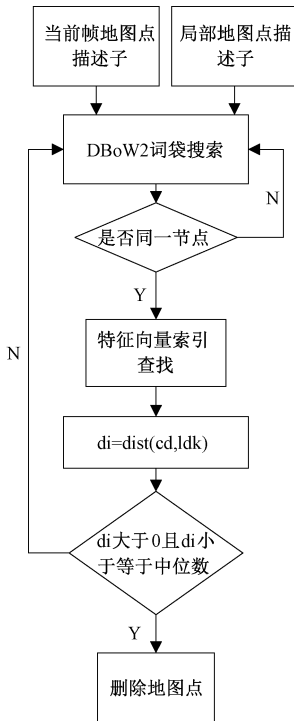


图 6 重复地图点消除流程

Fig. 6 Duplicate map point elimination process

2 实验测试

算法运行环境为 Intel(R) Core(TM) i5-13500HX CPU 最大频率为 4.7 GHz, 16 G 内存, GPU 为英伟达 RXT4060, 操作系统为 Ubuntu18.04, CUDA11.3。实验测试分为 4 部分: 区分高低动态环境及高动态下特征剔除, 重复地图点消除结果分析, 评估系统在 TUM RGB-D 数据集的定位精度, 最后实时性分析。

2.1 区分高低动态环境及高动态下特征剔除

图 7 所显示绿色方框和圆点所构成就是图像特征点的标记。

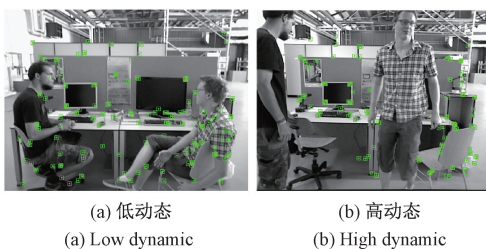


图 7 高低动态环境下所保留的特征点对比

Fig. 7 Comparison of feature points retained in high and low dynamic environments

如图 7 (a) 所示, 当环境处于低动态时, 即人物处于坐着的状态, 且并没有大幅度动作, 本文算法系统会对人物上的特征点进行保留。而对于环境处于高动态情况下, 系统会使用掩膜将人物上特征点进行剔除, 如图 7 (b) 所示。

2.2 重复地图点消除结果分析

在图 8 中展示了重复地图点消除算法前后对比, 在动态环境下, 系统的匹配跟踪会在地图中留下重复性的地图点。

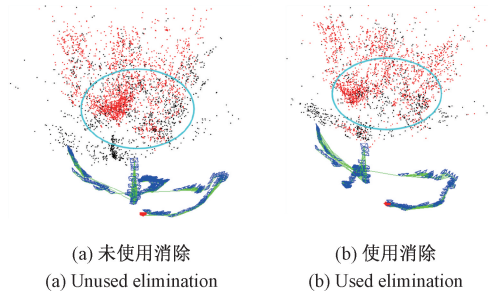


图 8 未使用与使用重复地图点消除算法的地图对比

Fig. 8 Comparison of maps without and with duplicate map point elimination algorithm

如图 8 (a) 所示, 在未使用消除算法的情况下, 蓝色椭圆圈部分是运动物体经过的地方, 出现了重复性地图点使得地图中的地图点较为密集。图 8 (b) 是使用消除算法后的结果, 可看出在使用算法后同 (a) 的同一区域对比, 地图点变得稀疏了。

2.3 评估系统性能

本文定位精度实验使用公开数据集 TUM RGB-D 数据集^[23]。TUM 中 fr3 序列是动态场景数据集, 序列中包括 RGB 图像、深度图像和与序列对应的相机运动轨迹的真值。动态场景的 fr3 序列中又可分为高动态 fr3_walking 序列和低动态 fr3_sitting 序列, fr3_walking 序列场景主要表现为人物大部分时间处于走动状态, 而 fr3_sitting 序列表现为几乎大部分时间人物是坐着交谈和做一些幅度较小的手势的。

SLAM 系统的评估通常采用绝对轨迹误差 (absolute trajectory error, ATE) 做对比, 因此, 本文算法将与 ORB-SLAM3、DynaSLAM 以及 RDS-SLAM 做系统性能对比, 使用 ATE 来做为的性能评估指标。

如表 1 所示, 本文给出了 ORB-SLAM3、DynaSLAM、RDS-SLAM 以及本文算法关于 ATE 的均方根误差 (RMSE) 和标准差 (std) 对比。根据表 1 的数据, 表 2 所展示的是本文算法和相同基于 ORB-SLAM3 上所设计的 RDS-SLAM 相对于 ORB-SLAM3 的在定位精度上的提升。表 1 中数据 RDS-SLAM 数据来源于原文和 He 等^[24] 的文章数据与两篇文章的 ORB-SLAM3 对比提升百分比与本文 ORB-SLAM3 的数据相乘计算得到。

表 1 和 2 数据以粗体标记表现最好算法, “—” 标记代表数据在该方法不可用。由表 1 和 2 可以看出, 相较于 ORB-SLAM3 算法, 本文算法在 fr3 序列中均比 ORB-SLAM3 算法的结果有提升。在 fr3_sitting 序列, 即在低动态场景, 本文算法相对 ORB-SLAM3, 均方根误差 (RMSE)

表 1 TUM 数据集序列的绝对轨迹误差 (rmse, std) 对比

Table 1 Comparison of absolute trajectory errors (rmse, std) of sequence in TUM dataset

m

序列	ORB-SLAM3		DynaSLAM		RDS-SLAM		本文算法	
	rmse	std	rmse	std	rmse	std	rmse	std
fr3/s_static	0.011 1	0.005 4	0.007 5	0.003 5	0.010 4	0.005 4	0.008 5	0.004 2
fr3/s_xyz	0.016 4	0.007 0	0.022 3	0.010 4	0.019 1	0.008 5	0.014 9	0.007 0
fr3/s_half	0.053 3	0.018 5	0.029 7	0.016 0	0.063 0	0.023 3	0.021 1	0.008 3
fr3/s_rpy	0.027 5	0.017 0	0.068 5	0.044 4	—	—	0.026 0	0.0161
fr3/w_static	0.172 0	0.061 3	0.010 8	0.004 5	0.034 3	0.013 8	0.009 2	0.004 0
fr3/w_xyz	0.395 8	0.234 2	0.031 5	0.016 7	0.010 4	0.006 7	0.026 1	0.014 4
fr3/w_half	0.533 2	0.168 7	0.029 2	0.013 4	0.024 8	0.009 2	0.029 1	0.012 9
fr3/w_rpy	0.710 1	0.363 2	0.046 9	0.026 2	0.040 9	0.026 9	0.038 3	0.020 4

表 2 本文算法与 RDS-SLAM 在 fr3 序列中定位精度相对 ORB-SLAM3 提升

Table 2 The positioning accuracy of this algorithm and RDS-SLAM in fr3 sequences is improved compared with ORB-SLAM3

%

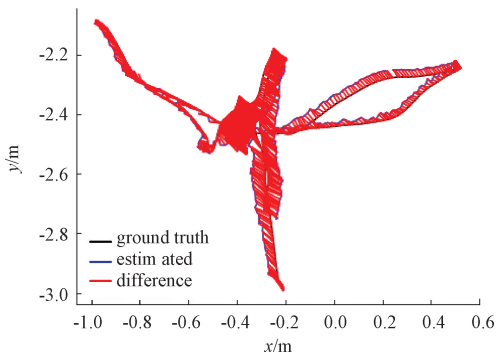
算法	s_static	s_xyz	s_half	s_rpy	w_static	w_xyz	w_half	w_rpy
RDS-SLAM	6.67	-16.67	-18.25	—	80.08	97.39	95.34	94.24
本文	23.42	9.15	60.41	5.45	94.65	93.41	94.54	94.61

更小,精度最大提升 60.41%,这个最大提升表现在 fr3_sitting_halfsphere 中,是因为该序列中的人物有一小段时间动作幅度较大引起重投误差超过阈值触发了动态特征剔除,并结合重复地图点消除相对于 ORB-SLAM3 使用 RANSAC 算法排除异常值,在效果上会有更高的提升。在高动态场景 fr3_walking 序列中,定位精度都提升 93% 以上,最大提升 94.65%。

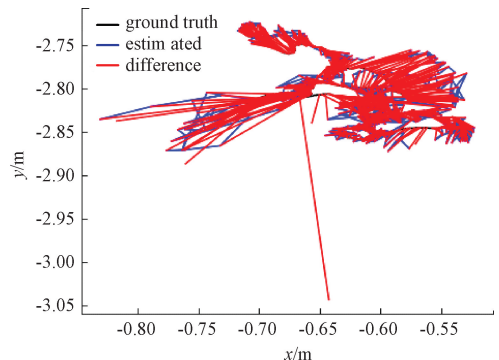
由表 1 数据可看出在绝大部分序列上本文算法的绝对轨迹误差是小于较经典的针对动态环境 SLAM 算法 DynaSLAM,数据表明本文算法的定位精度优于 DynaSLAM。在 fr3_sitting_xyz 和 fr3_sitting_rpy 序列中出现 DynaSLAM 轨迹误差大于不主要针对动态环境的 ORB-SLAM3,尤其 fr3_sitting_rpy 序列误差显著大于 ORB-SLAM3,说明 DynaSLAM 在低动态环境下出现过度剔除低动态物体特征。在表 2 中 RDS-SLAM 在低动态环境 fr3_sitting 序列下,出现两个序列定位精度相对 ORB-

SLAM3 提升的百分比是负数,说明了该算法在低动态环境下过度删除处于低速运行对象上的特征。基于 RANSAC 算法和共视关系的 ORB-SLAM3 机制能够在低动态场景中有效过滤部分异常的动态特征点,因此在低动态环境下 RDS-SLAM 去除低动态物体的特征会影响定位精度。在高动态 fr3_walking 序列中,本文算法定位精度上与 RDS-SLAM 的结果差别不显著,在部分序列上有优势。

图 9 和 10 是 ORB-SLAM3 和本文算法估计轨迹同真实轨迹的误差比较图,图中蓝线代表算法估计轨迹,黑线代表真实轨迹,红线代表估计轨迹和真实轨迹之间的误差,误差越大红线就越长。图 9 代表低动态环境,相对 ORB-SLAM3 估计轨迹和真实值的误差上,本文算法降低的误差不算明显,但是从细节上看本文算法估计轨迹是更接近真实的。图 10 代表高动态环境下,使用实例分割算法本文算法在估计轨迹上都明显优于 ORB-SLAM3。



(a) ORB-SLAM3/s_half



(b) ORB-SLAM3/s_rpy

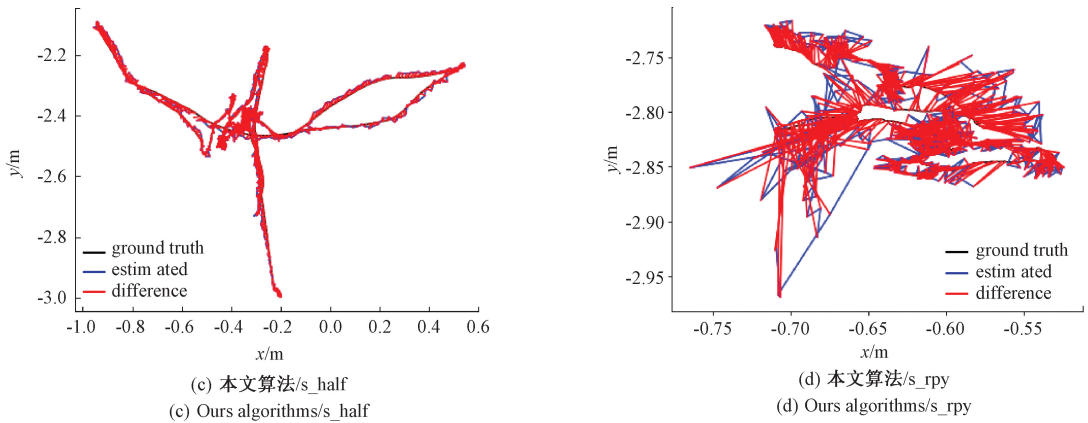


图 9 低动态情况下绝对轨迹误差对比图

Fig. 9 Comparison chart of absolute trajectory error under low dynamic conditions

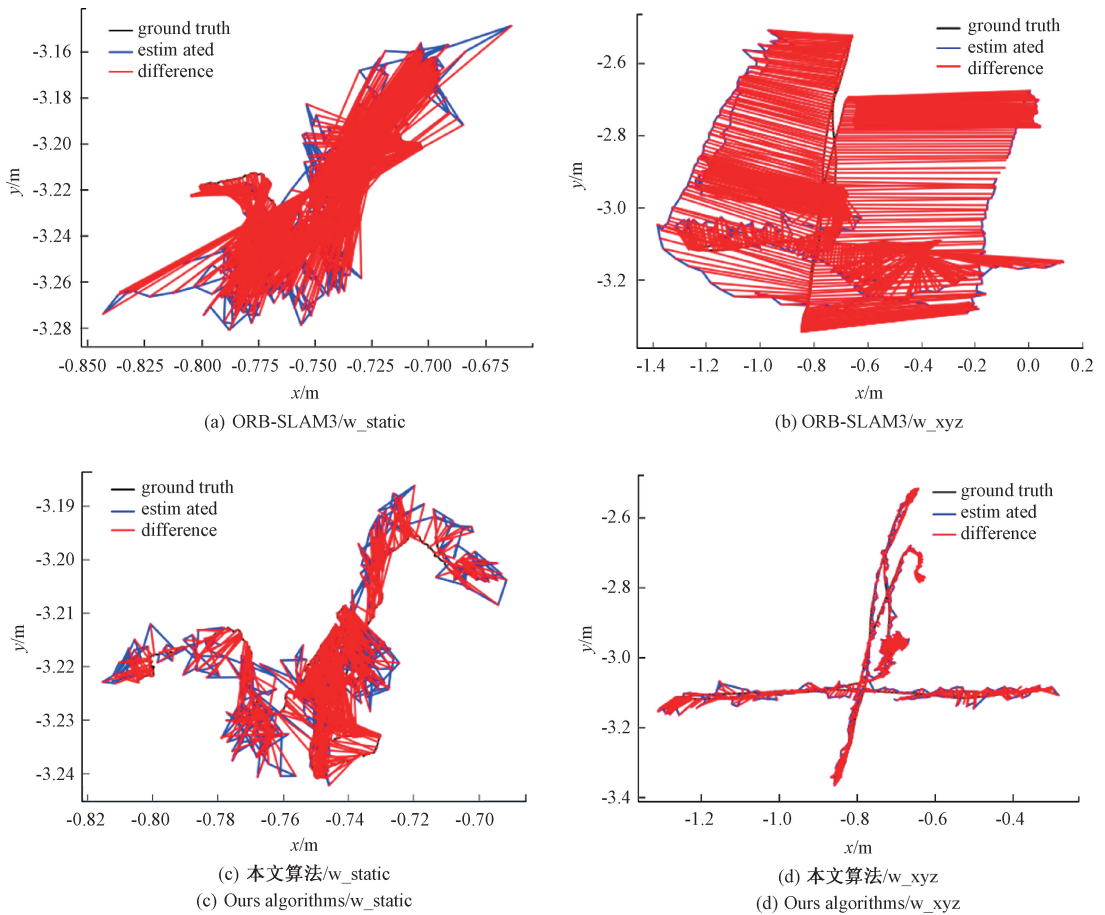


图 10 高动态情况下绝对轨迹误差对比图

Fig. 10 Comparison chart of absolute trajectory error under high dynamic conditions

2.4 实时性分析

本文采用 YOLOv8n-Seg 实例分割模型处理图片的速度是平均每帧 9.86 ms。本文算法相对于 ORBSLAM3 算法多加入区分高低动态线程和实例分割线程,因此算法的复杂度增加了,系统运行时间也会增加。本文算法使用了 TensorRT 加速网络处理图像的速度,在 fr3_walking_xyz

序列的平均每帧的跟踪时间为 28.32 ms,速度能达到 35 fps。原 ORB-SLAM3 算法平均每帧跟踪时间为 16.73 ms,本文算法与其对比,跟踪时间上多了 12 ms 左右,但仍然满足视觉 SLAM 的实时性。与同样使用实例分割网络 Mask R-CNN 的 DynaSLAM 以及 RDS-SLAM 相比,DynaSLAM 平均每帧跟踪速度大于 1 000 ms,RDS-

SLAM 平均每帧跟踪速度 50~65 ms,在实时性上,本文算法明显占优。

3 结 论

本文提出了一种可区分室内高低动态环境的视觉 SLAM,在该系统中加入了区分环境高低动态线程和实例分割线程完成区分环境动态状况和完成对图像的实例分割,且在跟踪线程中做动态点剔除,最后使用重复地图点消除算法消除在地图中的出现的重复地图点。本文 SLAM 系统解决了在低动态环境下,不会对先验动态对象上的特征进行剔除,保证跟踪的稳定,在高动态环境下,做动态特征剔除,可明显降低运动物体带来的动态特征对 SLAM 系统跟踪的影响。在局部跟踪加上的重复地图点消除算法可对因动态特征在地图中留下的重复性地图点进行消除,促使系统跟踪更加稳定。本文算法现只针对室内环境,针对室外,环境会更复杂,环境中先验动态对象会更多,下一步可考虑扩展到针对室外场景。在实时性上,采用的 YOLOv8n-Seg 实例分割网络使用 TensorRT 进行加速处理,系统实时性有了保证。

参考文献

- [1] PENG J SH, CHEN D H, YANG Q, et al. Visual SLAM based on object detection network: A review [J]. *Computers, Materials & Continua*, 2023, 77 (3): 3209-3236.
- [2] AI Y B, RUI T, YANG X Q, et al. Visual SLAM in dynamic environments based on object detection[J]. *Defence Technology*, 2020, 17(5): 1712-1721.
- [3] 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(8):51-62.
FENG ZH, XU X Y, ZHENG Y X, et al. Multi-object monocular SLAM base on instance segmentation and 3D reconstruction in dynamic sence[J]. *Chinese Journal of Scientific Instrument*, 2023, 44(8): 51-62.
- [4] CUI L Y, MA CH W. SDF-SLAM: Semantic depth filter SLAM for dynamic environments [J]. *IEEE Access*, 2020, 8: 95301-95311.
- [5] BESCÓS 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.
- [6] 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.
- [7] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN [C]. *IEEE International Conference on Computer Vision*, 2017: 2961-2969.
- [8] YU CH, LIU Z X, LIU X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), 2018: 1168-1174.
- [9] 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.
- [10] ZHONG F W, WANG SH, ZHANG Z Q, et al. Detect-SLAM: Making object detection and SLAM mutually beneficial[C]. 2018 IEEE Winter Conference on Applications of Computer Vision(WACV), 2018: 1001-1010.
- [11] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]. *Computer Vision-ECCV 2016; 14th European Conference*, 2016: 21-37.
- [12] CHANG J F, DONG N, LI D L. A real-time dynamic object segmentation framework for SLAM system in dynamic scenes [J]. *IEEE Transactions on Instrumentation and Measurement*, 2021, 70: 1-9.
- [13] BOLYA D, ZHOU CH, XIAO F Y, et al. YOLACT: Real-time instance segmentation [C]. *IEEE/CVF International Conference on Computer Vision*, 2019: 9157-9166.
- [14] LIU Y B, MIURA J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods [J]. *IEEE Access*, 2021, 9: 23772-23785.
- [15] 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.
- [16] SUN Y X, LIU M, MENG M Q H. Improving RGB-D SLAM in dynamic environments: A motion removal approach [J]. *Robotics and Autonomous Systems*, 2017, 89: 110-122.
- [17] BAKER S, MATTHEWS I. Lucas-Kanade 20 years on: A unifying framework[J]. *International Journal of Computer Vision*, 2004, 56: 221-255.
- [18] 张钧程,柯福阳,王旭. 改进 ORB 提取匹配算法的 SLAM 应用研究[J]. *电子测量技术*, 2024, 47(3): 91-101.
ZHANG J CH, KE F Y, WANG X. Research on SLAM application with improved ORB extraction and matching algorithms [J]. *Electronic Measurement Technology*, 2024, 47(3): 91-101.
- [19] HUSSAIN M. YOLOv1 to v8: Unveiling each

- variant-a comprehensive review of YOLO[J]. IEEE Access, 2024, 12: 42816-42833.
- [20] ZHANG Q, LI CH D. Semantic SLAM for mobile robots in dynamic environments based on visual camera sensors [J]. Measurement Science and Technology, 2023, 34(8): 085202.
- [21] NI J J, WANG L, WANG X T, et al. An improved visual slam based on map point reliability under dynamic environments[J]. Applied Sciences, 2023, 13(4): 2712.
- [22] GÁLVEZ-LÓPEZ D, TARDOS J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on Robotics, 2012, 28(5): 1188-1197.
- [23] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012: 573-580.
- [24] HE J M, LI M R, WANG Y Y, et al. OVD-SLAM: An online visual SLAM for dynamic environments[J]. IEEE Sensors Journal, 2023, 23(12): 13210-13219.

作者简介

符强,正高级实验师,硕士生导师,主要研究方向为图像处理、卫星导航和定位。

E-mail:2385698924@qq.com

曾凡治,硕士研究生,主要研究方向为视觉同时定位与建图(SLAM)。

E-mail:a1827870@163.com

纪元法(通信作者),教授,博士生导师,主要研究方向为卫星导航。

E-mail:2310396462@qq.com

任风华,副教授,硕士生导师,主要研究方向为图像处理。

E-mail:643749271@qq.com