

基于 YOLACT 网络的移动机器人视觉 SLAM 算法

桂昊,张庆永,袁一卿

(福建理工大学 机械与汽车工程学院,福建 福州 350118)

摘要: 提出一种室内动态场景的视觉 SLAM 算法,引入实例分割网络 YOLACT,剔除大部分动态点,利用多视图几何进一步过滤分割掩膜外未被剔除的动态特征点,使用剩余的静态特征点作为相机位姿估计;同时构建点云地图,转换并建立八叉树地图;使用背景修复以恢复被剔除动态物体后的背景。为验证算法的有效性,使用 TUM 数据集测试,并与 ORB-SLAM2 算法和其他处理动态场景的 SLAM 算法对比,结果表明,提出的算法在高动态数据集上表现良好。相较于 ORB-SLAM2 算法,提出的算法在室内动态场景中的定位精度提升 93.06%,可应用于后期机器人定位导航使用。

关键词: 实例分割网络;SLAM;多视图几何;动态场景;静态稠密地图

中图分类号: TP242.6

文献标志码: A

文章编号: 2097-3853(2024)01-0065-09

YOLACT network-based algorithm of visual SLAM of mobile robots

GUI Hao, ZHANG Qingyong, YUAN Yiqing

(School of Mechanical and Automotive Engineering, Fujian University of Technology, Fuzhou 350118, China)

Abstract: A visual SLAM algorithm for indoor dynamic scenes was proposed. The instance segmentation network YOLACT was introduced to eliminate most of the dynamic points. The multi-view geometry was used to further filter the dynamic feature points that were not eliminated outside the segmentation mask. The remaining static feature points were used as camera pose estimation. At the same time, the point cloud map was constructed, the octree map was transformed and established; background repair was used to restore the background after dynamic objects were removed. Finally, in order to verify the effectiveness of the proposed algorithm, the TUM dataset was used for testing, and compared with the ORB-SLAM2 algorithm and other SLAM algorithms processing dynamic scenarios, and results show that the proposed algorithm performs well on the highly dynamic dataset. Compared with the ORB-SLAM2 algorithm, the positioning accuracy of the proposed algorithm in indoor dynamic scenes is improved by 93.06%, and it can be applied to the later use of robot positioning and navigation.

Keywords: instance segmentation network; SLAM; multi-view geometry; dynamic scenes; static dense maps

SLAM(simultaneous localization and mapping, SLAM)即同步定位与建图,指的是当机器人或车辆在未知环境中,以不需要任何先验环境或者物体信息的情况下,以自身位姿为基础,建立一个环境度量地图^[1-2],实现机器人或车辆载体自主定位、导航和避障等动作的过程。目前,SLAM 算法在较低动态动作或者检测的动态点出现时无法精

准定位,精准定位相机位姿是关键。

近年来,学者们提出了一些动态环境下的视觉 SLAM 系统。Dai 等^[3]提出一种点云相关性的分割方法将静态和动态点分割开,以提高动态环境下的定位鲁棒性与准确性。Zhang 等^[4]提出一种光流残差来实现对相机自我运动估计,从而分割静态与动态目标,并完成静态背景重建。

收稿日期:2023-09-26

基金项目:福建省省级科技项目(GY-Z21004);福建工程学院科研项目(GY-Z20170)

第一作者简介:桂昊(1997—),男,湖北黄冈人,硕士研究生,研究方向:移动机器人同步建图与定位。

通信作者:张庆永(1980—),男,山东枣庄人,副教授,博士,研究方向:智能汽车、汽车系统动力学。

Bescos 等^[5]提出将实例分割网络与多视角几何结合,实现了动态特征点的剔除以及背景修复的功能。Chang 等^[6]使用轻量级目标检测网络,检测动态环境中的目标,通过剔除达到相机运动定位的目的。Wen 等^[7]提出一种几何语义的方法,实现动态点与静态点的分离,重建语义地图。房立金等^[8]通过光流法与实例分割网络的结合,剔除动态物体,提高了定位精度。DP-SLAM^[9]使用移动概率传播模型进行动态点检测,去除动态点以提升相机定位精度。Fan 等^[10]提出使用语义模型获取目标的边界框和掩码,得到更详细的区域分割结果,从而实现更精确的跟踪。

本文通过研究实例分割网络融合的 SLAM 算法,引入多视图几何策略对动态物体分割,通过消融实验对比,分析轨迹位姿误差的结果,并与国外先进算法对比,为基于 YOLACT 网络视觉 SLAM 算法提供理论与实验基础。

1 系统框架

1.1 系统主要结构

ORB-SLAM2,主要由跟踪、局部地图和回环 3 个线程组成,3 个线程结合使相机轨迹位姿和实时稀疏地图保持全局一致性,是较为优秀的 SLAM 系统之一(图 1)。

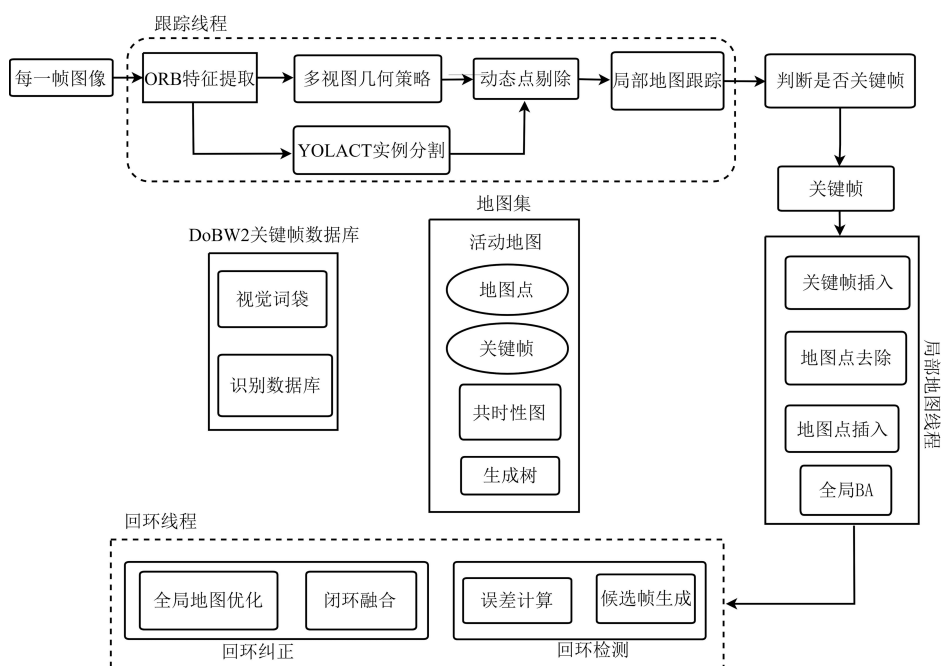


图 1 算法总体框架

Fig.1 Overall framework of algorithm

本文在原有的 ORB-SLAM2 系统跟踪线程上结合了 YOLACT 实例分割和多视图几何策略剔除动态物体,剩余的静态特征点用于相机位姿估计,动态点的引入会添加一些不可预测的误差,比如累计误差等,从而降低整个系统的精度。局部地图线程将上一个线程产生的关键帧插入,遍历后择优建图。回环线程判断是否产生回环并优化全局地图和校正闭环融合。

如图 2 所示,RGB 图像帧和深度图帧进入系统,跟踪线程对 RGB 图像提取 ORB 特征点,实例分割线程,根据先验信息识别人类别,然后创建一个掩膜。使用多视图结合的方法检测图像中可能

剩余的动态特征点,剔除掉剩下的动态点,利用位姿估计,使用前历史帧和深度图进行背景修复。同时使用深度图和剔除动态点之后的剩余静态特征点建立稠密点云地图,并转换为八叉树地图,方便后续机器人在环境中能获得更多的语义信息,实现导航功能等。

1.2 使用 YOLACT 分割潜在物体

对检测动态对象使用实例分割网络 YOLACT^[11]获得图像的像素语义分割,在 COCO 数据集上训练的 YOLACT 可以分割多达 80 个类别,它的分割处理速度在 30 帧/s 左右,接近实时性处理网络。标签包括:人、自行车、汽车、椅子、

电脑等。与目标检测算法相比,实例分割能够创建更细致的物体轮廓,不会因目标检测框过大而删除过多的静态特征点;同时与语义分割相比,实例分割可以得到每个指定物体的实例掩膜,解决不同场景下同类对象具有不同运动状态的问

题。如图 3(a)所示,输入 RGB 图像,图 3(b)显示了经过 YOLACT 处理后的输出图像,包括检测物体的掩膜、矩形框、标签和置信度。图 3(c)显示了通过语义先验将动态物体对象掩膜合成最终掩膜。

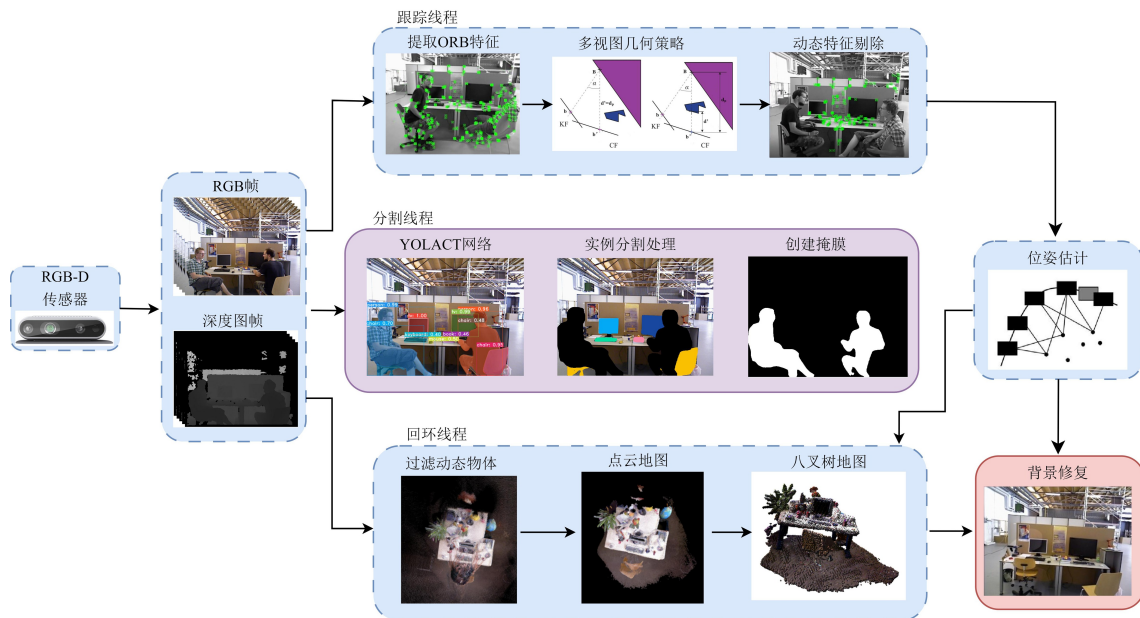


图 2 系统框架演示

Fig.2 System framework demonstration

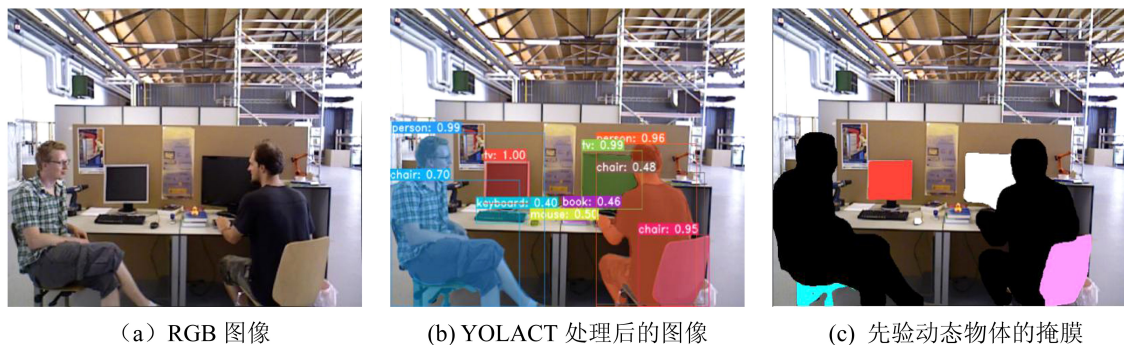


图 3 实例分割网络的处理

Fig.3 Instance segmentation network processing

1.3 结合多视图几何和实例分割网络对动态内容分割

对于传入的每一个输入帧,如图 4 所示,把之前关键帧 KF 中的每个关键点都投影到当前帧 CF,得到关键点 b' 和投影深度值 d_p ,每个关键点所对应的 3D 点是 B 。接着计算 B 和 b' 反投影之间的视差角度 α 。如果这个角度大于 30° 则该点可能被遮挡,从那时起将被忽略。

在 TUM 数据集中,视差角度大于 30° 的静态

对象被认为是动态的,因为它们的视点差异。考虑到重投影误差,根据深度图获得投影点 b' ,到所对应 3D 点 B 之间的实际测量深度为 d' ,根据

$$\|d' - d_p\| > d_{thre} \quad (1)$$

式中, d_{thre} 为实际测量深度与计算投影深度之差,称为阈值,此阈值由 $0.7 \times \text{Precision} + 0.3 \times \text{Recall}$ ^[12] 决定,并设为 0.4。如果两者之差超过阈值 d_{thre} ,则认为该点是被动态对象干扰的动态关键点。图 5 为多视角几何动态检测效果,经掩膜之后动态

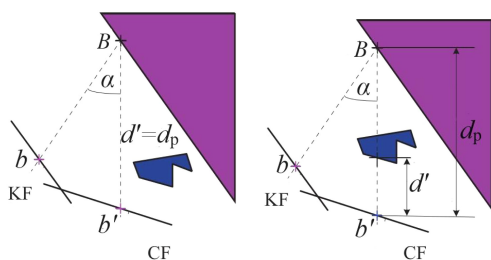


图 4 多视图几何策略的动态点检测

Fig.4 Dynamic point detection for multi-view geometry strategy

特征点已被剔除。如图 5 中的场景,当人坐在椅子上,椅子跟随人轻微晃动,椅子此时从静态变成动态物体,但不能将椅子视为动态物体对象剔除,否则将缺失大量静态点,造成后续定位误差大精度低的问题。如图 5 所示,5(a)为仅使用实例分割网络分割,此时椅子上还存在特征点;图 5(b)为实例分割创造的掩膜,图 5(c)为使用多视图几何策略,此时椅子上已经没有特征点,图 5(d)为结合了多视图几何和实例分割,此时椅子上已没有特征点,证明此融合方法的有效性。

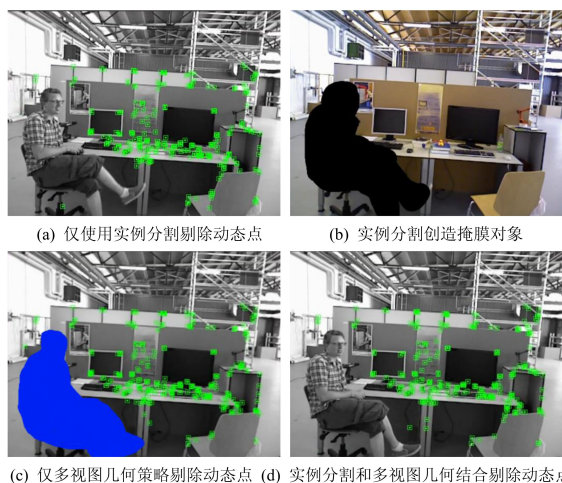


图 5 使用实例分割和多视图几何策略剔除动态点

Fig.5 Eliminating dynamic points using instance segmentation and multi-view geometry strategies

1.4 背景修复

由于动态物体对象的剔除,造成了背景空洞的出现,针对之前被剔除的动态对象,使用 20 个历史关键帧中的静态信息重新修复了被动态对象挡住的背景,利用运动估计和相机位姿信息,将动态物体从场景中分离出来,生成一个背景估计,将此背景估计应用到原始图像中,还原出纯净的背景。当 20 个历史关键帧没有出现之前帧的深

度信息,将会导致运动估计失效,使背景修复失败或者背景修复不完全等情况。如图 6 所示,图 6(a)为修复 RGB 图,由于此时两人出现身体交叉的情况,导致依旧会出现人体部分,发生相机位姿计算不准确,造成背景没有修复完全。图 6(b)为两人坐在椅子上,依据之前的历史关键帧让其修复完全。



(a) 背景未修复完全

(b) 背景修复完全

图 6 背景修复

Fig.6 Background repairing

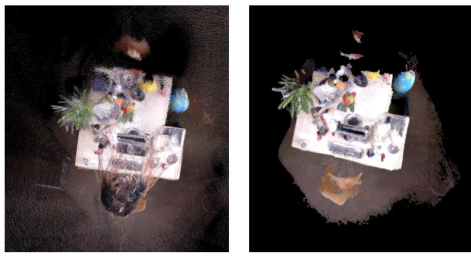
1.5 稠密地图和八叉树地图构建

地图构建是视觉 SLAM 系统中的关键功能之一。点云地图是通过 6 个参数,包括三维坐标 (X, Y, Z) 和颜色 (R, G, B) 组成的多点地图。得到关键帧位姿信息后,利用深度图和 RGB 帧,将剩余的静态特征点建立点云地图。如图 7 所示,图 7(a)为未剔除动态物体的点云地图,图 7(b)为剔除点云地图,将点云地图转化为不同分辨率的八叉树地图,八叉树地图具有网格精度可变的特点,根据不同细粒度的网格进行建模,如图 7(c)为分辨率 0.1 像素/m 的八叉树地图,图 7(d)为分辨率 0.5 像素/m 的八叉树地图。同时八叉树具有可压缩的特点,在导航中占用较小空间地图,其中点云地图如图 7(b)所占空间为 60Mb,八叉树地图如图 7(c)(d)所占空间分别为 0.626、0.021 Mb。

2 实验与分析

2.1 实验环境和评价指标

对 TUM RGB-D 公开数据集^[13],其中 RGB 图像和深度图像分辨率都为 640×480 像素,地面真实轨迹由高速捕获装置记录。高精度捕获系统和惯性测量系统实时获得的实际数据集中轨迹可以看作是相机实际位置的数据。本文提出的 SLAM 系统在 Ubuntu 20.04 系统上进行实验,该系统配备为 AMD R7 5800 CPU, NVIDIA RTX3060 LAPTOP GPU 和 32Gb RAM。



(a) 未剔除动态对象的点云地图 (b) 本文剔除的点云地图



(c) 分辨率 0.1 像素/m 的八叉树地图 (d) 分辨率 0.5 像素/m 的八叉树地图

图 7 点云地图和八叉树地图

Fig.7 Point cloud map and octree

SLAM 算法最关注对精度的评估,本文使用两个精度指标:绝对轨迹误差(A TE)和相对轨迹误差(RPE)表示。相对轨迹误差用于计算两个相同时间段上位姿真实轨迹值和估计轨迹值之间的差距,此数值用于表示算法系统的漂移量,用于计算第 i 帧的相对轨迹误差。

$$E_i = (Q_i^{-1} Q_{t+\Delta})^{-1} (P_i^{-1} P_{t+\Delta}) \quad (2)$$

式中, E_i 为 i 时刻的相对位姿估计误差; Q_i 为 t 时刻的真实位姿, m ; P_i 为 t 时刻的估计位姿, m ; Δ 为时间间隔参数,取值为 1 s。

计算绝对轨迹误差时,需要将真实地面值和估计值进行时间戳对齐,此值评估的是算法系统估计值和算法系统真实地面值之间的重合度,能明确表示算法的精准性和全局轨迹一致性。式(3)计算第 i 帧的绝对轨迹误差。

$$F_i = Q_i^{-1} S P_i \quad (3)$$

式中, F_i 为 i 时刻的相对位姿估计误差; Q_i 为 t 时刻的真实位姿, m ; S 为刚体变换对应的最小二乘解; P_i 为 t 时刻的估计位姿, m 。

使用均方根误差(RMSE)、中位数(Median)、标准偏差(STD)作为评估这 3 个指标的参数,其中均方根误差是用来衡量观测值与真值之间的偏差,如式(4)所示:

$$\text{RMSE}(F_{1:n}) = \left(\frac{1}{n} \sum_{i=1}^n \|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} \quad (4)$$

式中, $F_{1:n}$ 为从第 1 帧数据到整个系统的第 n 帧数

据的位姿估计误差; $\text{trans}(F_i)$ 为位姿误差分量, m 。

中位数能用来衡量估计位姿的离群值,反应系统整体的中心位置,当总体数据帧 n 为奇数时,中位数计算如式(5)所示。

$$\text{Median} = \left(\|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} \quad (5)$$

当总体数据帧数 n 为偶数时,中位数计算如式(6)所示:

$$\text{Median} = \frac{1}{2} \left[\left(\|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} + \left(\|\text{trans}(F_i)\|^2 \right)^{\frac{1}{2}} \right] \quad (6)$$

标准偏差用来衡量估计位姿相比于真实轨迹的离散程度,可以反应系统的稳定性,计算如式(7)所示。

$$\text{STD} = \left[\frac{1}{n} \sum_{i=1}^n \left(\|\text{trans}(F_i)\|^2 - \overline{\|\text{trans}(F_i)\|^2} \right)^2 \right]^{\frac{1}{2}} \quad (7)$$

其中, $\overline{\|\text{trans}(F_i)\|^2}$ 为位姿误差分量的平均值。整个实验使用 $\text{evo}^{[14]}$ 工具计算。

对本文提出的算法在低和高动态环境数据集上仿真实验,使用 evo 工具将本文算法与 ORB-SLAM2 所估计的真实地面轨迹 groundtruth 进行对比。表 1 中给出的改进部分 Improve 是本文算法相较于 ORB-SLAM2 的提升效果,公式为:

$$\text{Improve} = \left(1 - \frac{\alpha}{\beta} \right) \times 100\% \quad (8)$$

式中: α 为本文提出算法的轨迹误差, m ; β 为 ORB-SLAM2 的轨迹误差, m ; Improve 为提升效果, %。

2.2 消融实验

以 ORB-SLAM2 为基础组,将仅使用实例分割网络 YOLACT 称为 ORB+YOLACT;将仅使用多视图几何称为 ORB+Mutil-view;两者相结合即为本文提出的算法,设定为 Our,其中平均误差为所有数据集产生的绝对轨迹误差总和的平均数,其式为:

$$\text{Average Error} = \frac{1}{n} \left(\sum_{i=1}^n \text{RMSE} \right) \quad (9)$$

式中, n 为数据集序列个数,个;RMSE 为式(4)均方根误差, m 。

表 1 为消融实验对照表,由表可看出,仅使用 YOLACT 过滤掉了大部分的动态特征点,在高动

态数据集下平均提升 92.28%，在低动态数据集下，仅使用多视图几何算法下 SLAM 系统性能也优于 ORB-SLAM2，ORB + Mutil-view 算法提升了 41.28%，证明了多视图几何策略的有效性。本文提出算法在高动态数据集下，提升尤为明显，在整

个实验中，平均提升 93.06%，低动态数据集中，实验人员只轻微的扭动和摆动身体，只有少数特征点，没有达到真实动态场景，故提升率不明显，但在高动态数据集中，实验人员大幅度走动，重叠身体，更能反映真实动态场景。

表 1 消融实验对比结果

Tab.1 Comparative results of ablation experiments

环境	数据集	绝对轨迹误差/m			
		ORB-SLAM2	ORB+YOLACT	ORB+Multi-view	Our
低动态	fr2_desk_with_person	0.071 6	0.073 9	0.006 0	0.073 3
	fr3/s/half	0.024 7	0.021 2	0.014 3	0.019 9
	fr3/s/xyz	0.009 1	0.014 7	0.009 4	0.014 6
	fr3/s/static	0.009 0	0.006 5	0.007 3	0.006 0
高动态	fr3/w/half	0.648 1	0.027 6	0.388 1	0.023 5
	fr3/w/rpy	0.789 8	0.029 3	0.650 3	0.028 4
	fr3/w/static	0.409 0	0.016 6	0.069 5	0.006 0
	fr3/w/xyz	0.719 8	0.017 3	0.429 3	0.014 3
	Average Error	2.681 1	0.207 1	1.574 2	0.186 0
	Improve/%	—	92.28	41.28	93.06

2.3 相机位姿轨迹

图 8 展示了本文提出的 SLAM 系统与 ORB-SLAM2 在同一环境数据集下的轨迹与相机真实地面移动轨迹的比较。其中横坐标和纵坐标代表着三维坐标系中的 x 方向位移和 y 方向位移，图中蓝色线代表算法估计轨迹，黑色线代表真实地面轨迹，红色线段代表真实轨迹与估计轨迹之间的误差差值，此红色线段越短代表误差越小。图 8 为高动态数据集 fr3/w/xyz 实验，其中图 8(a) 为本文提出算法轨迹误差，图 8(b) 为 ORB-SLAM2 算法轨迹误差，可以看出，ORB-SLAM2 算法在高动态场景下，没有剔除动态物体机制，导致发生了明显的轨迹漂移，产生了较大的误差。本文提出的 SLAM 系统轨迹误差明显小于 ORB-SLAM2，其估计轨迹更接近真实地面移动轨迹，明显看出轨迹误差较小，并且在整个轨迹段中保持了轨迹稳定，所体现本文提出系统精度较高。

2.4 实验数据分析与对比

表 2 将 ORB-SLAM2 与本文提出算法的绝对

轨迹误差进行了均方根误差、中值误差和标准差对比。在低动态环境中，数据集中物体产生较小的动态动作，ORB-SLAM2 算法能保留更多静态点参与定位，使其绝对轨迹误差和相对轨迹平移误差优于本文提出算法。在高动态环境下，数据集中物体产生大幅度动态动作，本文提出算法能剔除大部分动态点，比 ORB-SLAM2 算法的位姿估计精度有大幅度提升，平均提升 97.95%。在表 3 和表 4 中，Improve 为式 (8) 提升效果，对相对轨迹误差 PRE 的平移部分和旋转部分进行了对比，在低动态场景下，本文提出算法较 ORB-SLAM2 算法没有明显的优势。在高动态场景下，本文提出算法剔除大部分动态物体，其中平移部分和旋转部分平均提升了 96.93% 和 95.41%，证明本文提出算法优于 ORB-SLAM2 算法。

将本文算法与国外 DynaSLAM^[5]、DS-SLAM^[12]、RDS-SLAM^[15] 等较为优秀的动态场景处理算法进行比较，对比结果如图 9 所示，高度越低，误差越小。

表 2 ORB-SLAM2 与本文算法的绝对轨迹误差 (ATE) 对比

Tab.2 Comparison of absolute trajectory error (ATE) of ORB-SLAM2 and that of the proposed algorithm

环境	数据集	绝对轨迹误差/m						Improve/%		
		ORB-SLAM2			Our			RMSE	Median	STD
		RMSE	Median	STD	RMSE	Median	STD			
低动态	fr2_desk_with_person	0.071 6	0.070 3	0.016 0	0.073 3	0.070 5	0.016 6	-2.37	-0.28	-3.75
	fr3/shalf	0.024 7	0.018 8	0.013 1	0.018 5	0.013 7	0.009 7	25.11	26.70	25.41
	fr3/s/xyz	0.008 3	0.006 4	0.004 3	0.011 0	0.008 9	0.005 2	-32.55	-39.85	-20.96
	fr3/s/static	0.009 1	0.006 8	0.004 6	0.006 0	0.005 0	0.002 7	33.98	25.59	40.95
高动态	fr3/w/half	0.648 1	0.493 2	0.290 0	0.016 4	0.013 2	0.007 3	97.48	97.31	97.48
	fr3/w/rpy	0.789 9	0.529 8	0.345 8	0.028 4	0.020 3	0.016 0	96.40	96.17	95.38
	fr3/w/static	0.409 0	0.306 4	0.172 9	0.006 0	0.005 0	0.002 8	98.53	98.37	98.38
	fr3/w/xyz	0.719 8	0.445 2	0.419 0	0.014 3	0.010 6	0.007 4	98.01	97.62	98.23

表 3 ORB-SLAM2 与本文算法的相对轨迹平移误差 (RPE) 对比

Tab.3 Comparison of relative trajectory pose error (RPE) of ORB-SLAM2 and that of the proposed algorithm

环境	数据集	相对轨迹平移误差/m						Improve/%		
		ORB-SLAM2			Our			RMSE	Median	STD
		RMSE	Median	STD	RMSE	Median	STD			
低动态	fr2_desk_with_person	0.100 6	0.086 5	0.050 1	0.102 4	0.087 1	0.051 6	-1.79	-0.69	-2.99
	fr3/shalf	0.035 8	0.025 1	0.020 5	0.028 5	0.022 1	0.013 6	20.39	11.95	33.66
	fr3/s/xyz	0.013 4	0.011 2	0.006 0	0.021 6	0.019 0	0.008 8	-61.19	-69.64	-46.67
	fr3/s/static	0.013 8	0.009 8	0.007 2	0.008 7	0.007 4	0.003 8	36.96	24.49	47.22
高动态	fr3/w/half	0.977 8	0.711 6	0.580 6	0.033 9	0.027 6	0.015 5	96.53	96.12	97.33
	fr3/w/rpy	1.046 5	0.840 7	0.562 7	0.040 4	0.030 3	0.020 8	96.14	96.40	96.30
	fr3/w/static	0.584 9	0.094 2	0.420 8	0.008 7	0.007 3	0.003 8	98.51	92.25	99.10
	fr3/w/xyz	1.065 9	0.798 3	0.628 5	0.020 7	0.016 2	0.010 1	98.06	97.97	98.39

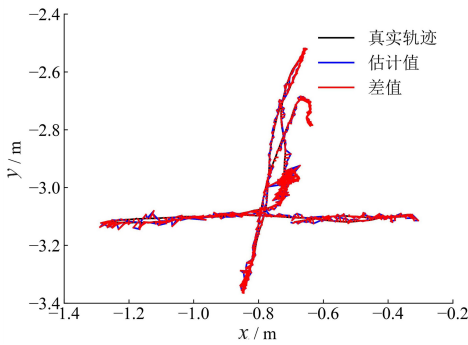
表 4 ORB-SLAM2 与本文算法的相对轨迹旋转误差 (RRE) 对比

Tab.4 Comparison of relative trajectory rotation error (RRE) of ORB-SLAM2 and that of the proposed algorithm

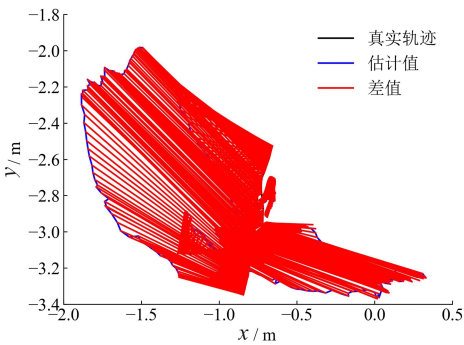
环境	数据集	相对轨迹旋转误差/(°)						Improve/%		
		ORB-SLAM2			Our			RMSE	Median	STD
		RMSE	Median	STD	RMSE	Median	STD			
低动态	fr2_desk_with_person	0.749	0.619	0.362	0.695	0.566	0.333	7.13	8.48	8.03
	fr3/shalf	0.873	0.719	0.387	0.797	0.644	0.367	8.72	10.49	5.24
	fr3/s/xyz	0.575	0.414	0.303	0.634	0.490	0.311	-10.27	-18.44	-2.47
	fr3/s/static	0.371	0.317	0.161	0.323	0.270	0.144	13.14	15.06	10.32

续表

环境	数据集	相对轨迹旋转误差/($^{\circ}$)						Improve/%		
		ORB-SLAM2			Our			RMSE	Median	STD
		RMSE	Median	STD	RMSE	Median	STD			
高动态	fr3/w/half	21.153	16.239	11.844	0.834	0.705	0.361	96.06	95.66	96.96
	fr3/w/rpy	18.502	13.991	10.606	0.919	0.690	0.493	95.03	95.07	95.35
	fr3/w/static	10.706	1.424	7.712	0.273	0.229	0.120	97.45	83.92	98.45
	fr3/w/xyz	20.467	15.268	12.028	0.630	0.429	0.385	96.92	97.19	96.8



(a) 本文提出算法轨迹误差



(b) ORB-SLAM2算法轨迹误差

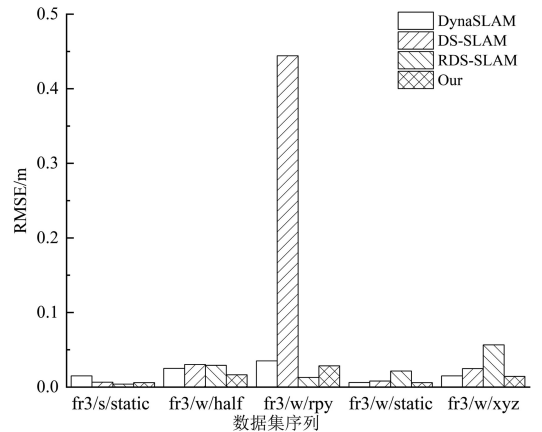
图 8 不同算法同一环境下绝对轨迹误差对比图

Fig.8 Comparison of absolute trajectory error of different algorithms in the same environment

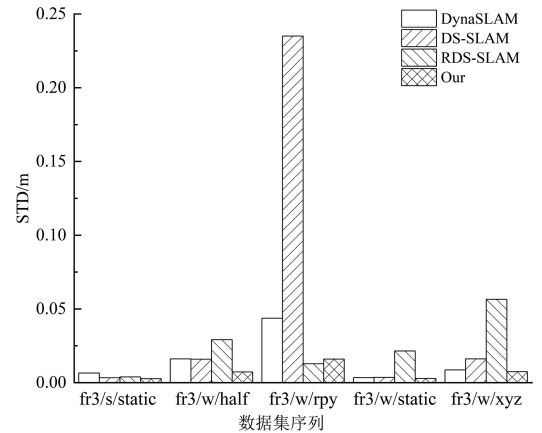
由图 9 可看出,本文提出算法在数据集 fr3/s/static 柱状图高度接近其他国外优秀算法,在 fr3/w/half、fr3/w/static、fr3/w/xyz 序列数据集上,本文提出算法误差皆低于其他算法,实验结果误差更小。因此,在同一数据集序列下实验,与其他算法相比较,本文提出算法具有一定的优势。

3 结论

1) 动态环境下引入 YOLACT 实例分割网络,其分割处理速度在 30 帧/s 左右获得相机帧中的图像语义信息,对图像中的运动实例物体进行分



(a) RMSE误差对比



(b) STD误差对比

图 9 本文算法与其他 3 种算法的绝对轨迹误差对比图

Fig.9 Comparison of absolute trajectory errors of the proposed algorithm and the other three algorithms

割。相较于 ORB-SLAM2 算法,在高动态和低动态数据集实验下,绝对轨迹误差精度平均提升 92.28%;通过引入多视图几何策略,对比 ORB-SLAM2 算法,绝对轨迹误差精度平均提升 41.28%;将 YOLACT 实例分割网络和多视图几何策略相融合,其绝对轨迹误差精度平均提升 93.06%。

2)在高动态数据集实验下,本文提出的算法相较于 ORB-SLAM2 算法,绝对轨迹误差精度平均提升了 97.95%,平移部分和旋转部分误差精度平均提升了 96.93%和 95.41%。对比其他算法,在高动态数据集序列下 fr3/w/half、fr3/w/static、fr3/w/xyz 的误差对比,本文提出算法具有一定的

优势。

3)本文提出算法为了可视化应用,将背景修复为动态物体遮挡之前的场景,并且利用剩余的静态特征点创建了点云地图模块,为后续移动机器人提供信息。

参考文献:

- [1] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age[J]. *IEEE Transactions on Robotics*, 2016, 32(6): 1309-1332.
- [2] MUR-ARTAL R, MONTIEL J M, TARDOS J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. *IEEE Transactions on Robotics*, 2015, 31(5): 1147-1163.
- [3] DAI W C, ZHANG Y, LI P, et al. RGB-D SLAM in dynamic environments using point correlations[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(1): 373-389.
- [4] ZHANG T W, ZHANG H Y, LI Y, et al. FlowFusion: dynamic dense RGB-D SLAM based on optical flow[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). Paris, France: IEEE, 2020: 7322-7328.
- [5] BESCOS B, FACIL 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] CHANG Z Y, WU H L, SUN Y L, et al. RGB-D visual SLAM based on Yolov4-tiny in indoor dynamic environment[J]. *Micromachines*, 2022, 13(2): 230.
- [7] WEN S H, LI P J, ZHAO Y J, et al. Semantic visual SLAM in dynamic environment[J]. *Autonomous Robots*, 2021, 45(4): 493-504.
- [8] 房立金, 刘博, 万应才. 基于深度学习的动态场景语义 SLAM[J]. *华中科技大学学报(自然科学版)*, 2020, 48(1): 121-126.
- [9] LI A, WANG J K, XU M, et al. DP-SLAM: a visual SLAM with moving probability towards dynamic environments[J]. *Information Sciences*, 2021, 556: 128-142.
- [10] FAN Y C, ZHANG Q C, TANG Y L, et al. Blitz-SLAM: a semantic SLAM in dynamic environments[J]. *Pattern Recognition*, 2022, 121: 108225.
- [11] BOLYA D, ZHOU C, XIAO F Y, et al. YOLACT: real-time instance segmentation[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South): IEEE, 2019: 9157-9166.
- [12] YU C, 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.
- [13] 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. Vilamoura - Algarve, Portugal: IEEE, 2012: 573-580.
- [14] GRUPP M. evo: Python package for the evaluation of odometry and slam [EB/OL].[2022-01-20].<https://github.com/MichaelGrupp/evo>.
- [15] LIU Y B, MIURA J. RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods[J]. *IEEE Access*, 2021, 9: 23772-23785.

(责任编辑: 陈雯)