钟富涛, 李泽滔, 牟刚. 基于 YOLO 特征点筛选的视觉 SLAM 算法[J]. 智能计算机与应用,2026,16(2):155-161. DOI:10.20169/j.issn.2095-2163.24040904

基于 YOLO 特征点筛选的视觉 SLAM 算法

钟富涛[1], 李泽滔[1], 牟 刚[2]
(1 贵州大学 电气工程学院, 贵阳 550025; 2 重庆公共运输职业学院 智慧交通学院, 重庆 402260)

摘 要: 针对传统的视觉 SLAM 算法在动态环境下的定位精度低的问题,本文提出了一种基于 YOLO 特征点筛选的视觉 SLAM 算法。该算法在 ORB-SLAM2 的基础上,加入 LK 光流法和 YOLO 目标检测算法对动态物体的特征点进行筛选,以提高算法在动态场景下的性能表现。实验结果显示,改进的算法在高动态环境下能够显著提升绝对轨迹误差的 RMSE 值,平均提升率达到 60.01%以上。

关键词: 视觉 SLAM; LK 光流法; YOLOv8; 动态特征点剔除
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 2095-2163(2026)02-0155-07

Visual SLAM algorithm based on YOLO feature point selection

ZHONG Futao[1], LI Zetao[1], MOU Gang[2]
(1 School of Electrical Engineering, Guizhou University, Guiyang 550025, China;
2 School of Intelligent Transportation, Chongqing Vocational College of Public Transportation, Chongqing 402260, China)

Abstract: This paper proposes a visual SLAM algorithm based on YOLO feature point filtering to address the low localization accuracy of traditional visual SLAM algorithms in dynamic environments. Building upon ORB-SLAM2, the algorithm incorporates Lucas-Kanade optical flow and YOLO object detection algorithms to filter feature points of dynamic objects, thereby enhancing the algorithm's performance in dynamic scenes. Experimental results demonstrate that the improved algorithm significantly increases the Root Mean Square Error (RMSE) of absolute trajectory in high dynamic environments, with an average improvement rate exceeding 60.01%.

Key words: visual SLAM; Lucas-Kanade optical flow method ; YOLOv8; dynamic feature point filtering

0 引 言

最近几年,辅助自动驾驶和智能机器人技术发展迅速,与之相关的及时定位与建图(Simultaneous Localization And Mapping,SLAM)技术已经广泛应用在运动机器人和自动驾驶车辆的地图构建和定位研究中[1-2]。由于成本低和效果显著,基于视觉的 SLAM 成为目前研究的热点[3-4]。目前现有视觉 SLAM 技术通常假设环境是静止不变的或者大部分是静态的。但是,当环境中存在动态物体时,如行人、车辆或者其它运动的对象,地图构建的定位精度可能会不准确,甚至在某些极端的情况下可能会导致 SLAM 建图和定位失败[5-8]。

针对这一问题,相关研究人员提出了一些异常值拒绝算法。Mur-Artal 等学者[9]提出在 ORB-SLAM2 中使用的 RANSAC (Random Sample Consensus),简单地将所有动态物体视为异常值。但却只能处理动态目标里缓慢运动的物体。Wang 等学者[10]提出了一种新的稠密 RGB-D SLAM 框架,其中通过基于点轨迹对图像中的目标进行聚类来检测动态对象,然后将其从姿态估计过程中加以排除。Sun 等学者[11]提出了具有运动去除模块的 DVO-SLAM 算法,该模块计算连续 RGB 图像的强度差异,以检测动态对象的边缘进行分割。Dai 等学者[12]提出了 DSLAM 算法,通过使用 Delaunay 三角剖分算法计算特征点的相邻关系来区分动态特征和静态特征。然而,上述算法仍然无法有效地区分动态目标和静止不动的物体。当真实环境中存在大

作者简介: 钟富涛(1998—),男,硕士研究生,主要研究方向:深度学习,图像处理; 牟 刚(1977—),男,副教授,主要研究方向:人工智能。
通信作者: 李泽滔(1960—),男,博士,教授,博士生导师,主要研究方向:智能电网,计算机控制技术。Email:1758604018@qq.com。

收稿日期: 2024-04-09

量动态物体时,这些方法的性能会变得比较差。

近年来,基于多层神经网络的深度学习发展迅速[13]。一些 SLAM 算法已经使用深度学习获取语义信息来处理定位和建图过程中的动态物体[14-15]。Brasch 等学者[16]提出了一个带有深度学习网络的 SLAM 框架,其中提取的语义信息和概率模型用于提取出环境中的动态物体。此外,Lianos 等学者[17]提出了另一个新的视觉里程计框架,将语义约束融合到姿态和地图优化过程中,以减少由动态物体引起的平移漂移。Bescos 等学者[18]提出了 DynaSLAM 算法,使用 Mask-RCNN 结合多视图几何来分割环境中的动态物体。Yu 等学者[19]提出了 DS-SLAM 算法,采用 SegNet 进行语义分割,并通过运动一致性检查区别出环境中的动态物体。Henein 等学者[20]提出了一种基于 Mask-RCNN 的新型基于特征的无模型感知对象的动态 SLAM 算法,用于估计刚体对象的运动。Zhang 等学者[21]使用光流残差为相机跟踪和背景重建提供 RGB-D 点云中的动态分割。虽然这些方法可以从场景中提取大多数可能的可移动物体,但却没有考虑这些对象的实际运动状态,当这些物体静止时,剔除相应的特征有可能会降低姿态估计的准确性。

针对当前动态环境视觉 SLAM 系统的定位精度低和实时性差的问题,本文在 ORB-SLAM2 的基础上提出了一种基于特征点筛选的改进算法。该算法使用 LK 光流法结合目标检测算法对动态物体的特征点进行筛选,以提高算法在动态场景下的性能表现。

本文主要工作如下:

(1)在 ORB-SLAM2 中引入 LK 光流法,对传入图像中的动态特征点和静态特征点进行初步的区分。

(2)加入 YOLOv8 目标检测算法,对影响建图的动态特征点进行进一步的筛选和剔除。

1 ORB-SLAM2 框架

ORB-SLAM2 全称 Oriented fast and Rotated Brief Simultaneous Localization and Mapping),即面向 FAST 和旋转 Brief 的同时定位与地图构建技术。这是一种用于单目、立体和 RGB-D 相机的实时 SLAM 算法,设计用于在未知环境中构建或更新地图的同时,实时追踪传感器在其中的位置。算法主要包括三大线程。

(1)跟踪线程。在 ORB-SLAM2 系统中,跟踪线程扮演着至关重要的角色,通过分析连续视频流中的画面来提取关键的 ORB 特征点,并将这些特征点与已经构建好的地图进行匹配,进而估算出相机的当前位置和朝向。这个过程不仅包括了 ORB 特征点的检测,而且还涉及到了特征点与地图之间的匹配。通过这种方式,系统能够以高效和精确的方式实现视觉里程计。

(2)局部地图构建线程。当跟踪线程成功识别并标记出新的关键帧后,ORB-SLAM2 的局部地图构建线程随即启动,主要就是对这些新关键帧进行处理。该线程通过三角测量法生成新的地图点,并利用局部 Bundle Adjustment(BA)技术对新加入的关键帧及地图点进行位置优化,以确保地图的准确性和稳定性。

(3)闭环检测线程。ORB-SLAM2 中设有闭环检测线程,作为系统内第 3 个核心线程,其主要作用是识别出机器人或相机回到之前探索过的区域,即发现闭环,这对于修正在长时间导航过程中积累的位置偏差及增强地图的整体一致性至关重要。一旦发现闭环,该线程便会激活全局位姿图优化机制,通过调整全部关键帧的姿态参数,以最小化因闭环产生的定位误差,这一过程显著提升了地图的精度。

这 3 个线程的设计允许 ORB-SLAM2 在保持实时性的同时,能够有效地处理大规模环境的同时定位与地图构建任务,使其成为一个强大且灵活的 SLAM 系统。ORB-SLAM2 的结构如图 1 所示。

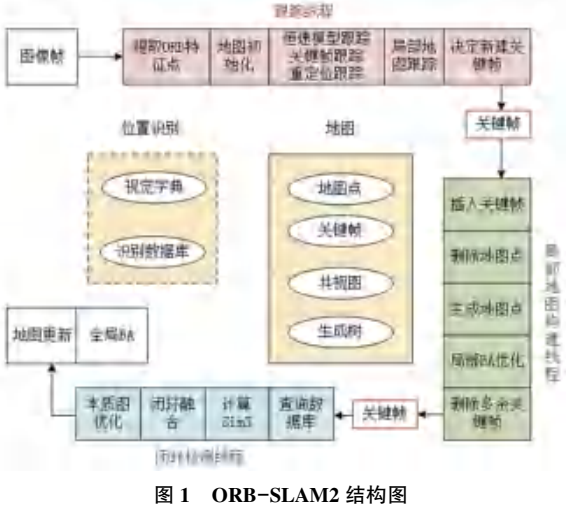


图 1 ORB-SLAM2 结构图

Fig. 1 Structure diagram of ORB-SLAM2

2 问题描述

SLAM 建图的关键是估计相机观测到的场景移动过程中的位姿变化。视觉里程计通过分析连续的图像序列来估计相机的运动,包括旋转和平移参数,从而能够推导出相机在空间中的移动轨迹。

视觉里程计的核心机制依赖于从图像中提取特征点并进行匹配，通过解析这些匹配特征点对之间的极线几何关系，进而推导出相机位姿的旋转和平移参数。但快速运动的物体会干扰到这一过程。

动态目标阻碍示意如图 2 所示，描绘了相机视角中有动态目标存在与否的 2 种情形。其中，相机运动前后所成的像是 I_1 和 I_2，p_1、p_2 分别是观测目标点 P 在连续的前后 2 帧里所成的象点。l、l' 是前后 2 帧所成象点对目标的观测深度。在小车进入相机的视野前，没有动态目标影响，P 点对目标的观测深度为：

$$l' = l_{\text{true}} \tag{1}$$

小车进入视野后，动态目标遮挡影响了观测深度的测定。实际的观测深度变为了 l_{false}，小于 l_{true}。由于动态目标遮挡导致的错误观测，后续 SLAM 建图过程中的位姿估计会受到很大的负面影响。

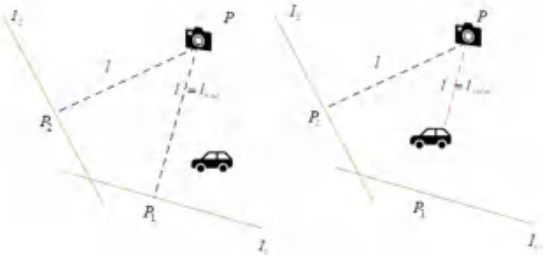
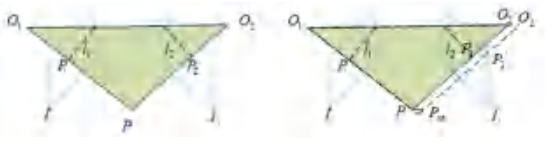


图 2　动态目标阻碍示意图

Fig. 2　Schematic diagram of dynamic target obstruction

动态目标干扰极线约束示意如图 3 所示，分别是相机在无动态目标干扰和有动态目标干扰情况下，观测目标点所形成的极线几何关系。其中，点 P 是目标在三维空间中的实际位置，O_1、O_2 是相机运动前后在三维空间的所处位置，I_1 和 I_2 是相机分别在 O_1、O_2 处观测成的图像。$\boldsymbol{p}_1$ 和 $\boldsymbol{p}_2$ 是 P 点在 I_1、I_2 两个图像中分别所成的像点。$P-O_1-O_2$ 三点确定了一个极平面，l_1、l_2 分别是 I_1、I_2 两图像平面和极平面的交线，命名为极线。当 P 点是静态不动的状态时，$\boldsymbol{p}_1$ 和 $\boldsymbol{p}_2$ 之间的匹配是准确无误的。视觉 SLAM 确定位姿的过程就是依赖正确的匹配点对（$\boldsymbol{p}_1$，$\boldsymbol{p}_2$），经过对极约束 $\boldsymbol{p}_2{}^{\mathrm{T}}(\boldsymbol{K}^{-1})^{\mathrm{T}}\boldsymbol{t}^{\wedge}\boldsymbol{R}\boldsymbol{K}^{-1}\boldsymbol{p}_1 = 0$（其中 $\boldsymbol{K}$ 是相机的内参，通常情况是已知的）的计算，最终得到相机从 O_1 移动到 O_2 时的旋转平移参数（$\boldsymbol{R}$，$\boldsymbol{t}$）的一个过程。当环境是静态的时候，这一过程不会出现太大的问题。但当环境中存在动态目标时。在实际的相机运动中，动态目标 P 点是运动的，从 P 点运动到了 P_{mv} 点，导致 2 个图像中的匹配点变成了 $\boldsymbol{p}_1$ 和 $\boldsymbol{p}_2{}'$。此时的极平面也变成了 $P_{mv}-O_1-O_2{}'$。这会导致后面计算出的相机位姿结果是从 O_1 到 $O_2{}'$ 的旋转平移，而不是实际空间中的 O_1 到 O_2 的平移旋转，这就导致了对相机位姿的错误计算结果。



（a）无动态目标干扰　　（b）有动态目标干扰

图 3　动态目标干扰极线约束示意图

Fig. 3　Schematic diagram of dynamic target interference limit line constraint

3　LK 光流法剔除动态特征点

视觉 SLAM 技术依赖于摄像设备在其移动路径上不断地采集图像序列，并对捕获到的图像进行深度加工处理。在这一系列的处理中，图像中的静态特征点展现出相对一致的运动轨迹，而与之形成鲜明对比的是，动态特征点的运动轨迹则充满了不可预测性。借助于光流技术对特征点运动模式的精确分析，这一方法能够有效辨识出图像中静态与动态特征点的差异。

LK 光流算法（Lucas-Kanade 方法）是一种计算视频序列中物体运动估计的技术，其核心假设建立在物体在连续帧间的运动是连贯且平稳的基础上。这意味着，可以将一小段时间内的图像部分移动视为平移运动，并假定这一过程中图像的亮度保持不变。通过这种方式，LK 光流算法致力于测量图像中每个像素点的移动速率、即光流，从而精确追踪物体在画面中的运动轨迹（如图 4 所示）。

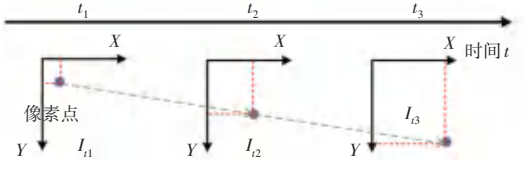


图 4　LK 光流法

Fig. 4　LK optical flow method

该方法建立在 3 个基本假设之上：首先，是灰度恒定性假设，即认为图像序列中同一位置的像素在连续帧中保持其灰度值不变；其次，小运动假设指出，相邻 2 帧之间的像素位移非常小；最后，空间一致性假设表明，邻近像素之间的运动模式是相似的。这些假设为光流估计提供了理论基础，使得通过分析像素的灰度变化来推导其运动变化成为可能。在灰度不变时，推得的公式如下：

$$I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) = I(x,y,t) \tag{2}$$

其中,相邻帧之间对应时间是 t 和 $t+\mathrm{d}t$。相邻帧中像素点的位置分别是 $I(x,y,t)$ 和 $I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t)$。对式(2)进行泰勒级数展开,只保留一阶项,可以得到:

$$I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t)\approx I(x,y,t)+\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial y}\mathrm{d}y \tag{3}$$

进一步整合得到:

$$\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial y}\mathrm{d}y=0 \tag{4}$$

其中,$\mathrm{d}x/\mathrm{d}t$ 与 $\mathrm{d}y/\mathrm{d}t$ 分别表示像素在 x 轴和 y 轴上运动的速度,记为 u、v。图像在该点 x 和 y 方向上的梯度是 $\partial I/\partial x$、$\partial I/\partial y$,记作 I_x 和 I_y。$\partial I/\partial t$ 是图像灰度随时间变化的量,记作 I_t。具体公式如下:

$$\begin{bmatrix} I_x & I_y \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}=-I_t \tag{5}$$

由于该方程有 2 个未知数,对其求解至少需要 2 个方程。因此,根据上述后 2 条需要满足的条件,研究认为在该像素周围的一个窗口内 w 的像素与该像素具有相同的运动状态。因此,可以构建 w^2 个方程,对此可以表示为:

$$\begin{cases} \begin{bmatrix} I_x & I_y \end{bmatrix}_k\begin{bmatrix} u \\ v \end{bmatrix}=-I_{tk},\ k=1,\cdots,w^2 \\ \boldsymbol{A}\begin{bmatrix} u \\ v \end{bmatrix}=-\boldsymbol{b} \end{cases} \tag{6}$$

$$\boldsymbol{A}=\begin{bmatrix} [I_x,I_y]_1 \\ \vdots \\ [I_x,I_y]_k \end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix} I_{t1} \\ \vdots \\ I_{tk} \end{bmatrix}$$

基于最小二乘求解可得结果,计算公式为:

$$\begin{bmatrix} u \\ v \end{bmatrix}^*=-(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \tag{7}$$

4 YOLOv8 检测动态目标

在上一小节,探讨了利用光流法追踪图像中的特征点以去除几何上的异常点。但是,这项技术在去除动态物体表面的移动点方面存在局限。通过融入目标检测算法,可以挖掘图像的深层语义信息,进而有效辨识出图像中的动态物体。基于此,本小节将采纳卷积神经网络为基础的目标检测方法 YOLOv8,精确识别并提取图像内动态物体的细节信息。从而提高动态特征点识别与剔除的准确性。

通过了 200 个训练周期后,YOLOv8 模型最终达到了稳定收敛的状态。借助这个精细训练过的网络,对不同动态环境(静态、低动态与高动态)下的图像进行了检测分析,检测结果以原图及其对应的检测图像的形式展现,如图 5 所示。

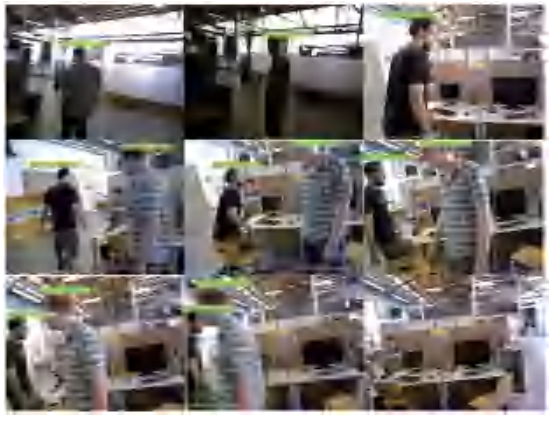


图 5 动态目标检测结果图

Fig. 5 Diagrams of dynamic target detection results

5 动态特征点剔除流程

在经典视觉 SLAM 系统中,利用静态特征点匹配来获得精准的相机位姿对于构建准确的地图和定位是尤为关键的。但是,在动态环境下,动态物体的出现会导致 SLAM 系统错误地识别了动态特征点,进而影响相机位姿的精度,最终影响到了定位和建图。为了增强系统的定位精度与稳定性,需要先辨识并排除这些动态特征点,随后再进行特征匹配。

基于卷积神经网络的目标检测技术可以有效识别图像中常见的动态物体(如行人、车辆等),同时将其整体轮廓框选出来。但这一方法无法判断识别出来的物体是否处于运动状态。LK 光流法可以在一定程度上区别静态特征点和动态特征点。然而,当动态物体在图像中所占比例过高时,依此方法计算得到的基础矩阵可能会有较大误差,从而影响动态特征点的准确性判断。因此,仅依赖目标检测或光流法并不能完全解决问题。本文将这 2 种技术结合应用,以达到在动态特征点检测上的更高精确度,从而为视觉 SLAM 系统在动态环境下的应用提供了一种更为可靠的解决方案。

动态特征点的剔除流程如图 6 所示。首先,系统将接收的 RGB 图像导入动态目标检测流程,目的是识别图像中的动态物体并提取其轮廓信息。同时,系统利用光流技术对图像中的特征点进行追踪,以识别出那些移动中的特征点。紧接着,系统会评估检测区域内的移动特征点数量,并将其与预设的阈值进行比较:若某个区域内的移动特征点数量高于阈值,该区域便被识别为含有动态物体,并将其内的特征点排除;反之,如果该数量低于阈值,该区域

则被视为静态物体，其内的特征点得以保留。通过这一过程，能够筛选出稳定的静态特征点集，从而为相机定位提供更高的精确度。

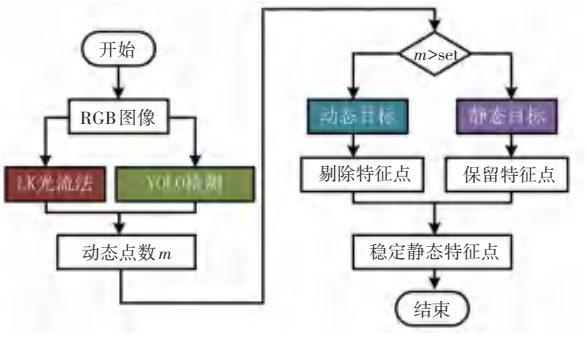


图 6　动态特征点剔除流程

Fig. 6　Flow chart of dynamic feature point filtering

6　位姿估计实验与分析

本节针对 SLAM 系统在动态环境遭遇挑战，提出改进措施，并通过与 ORB-SLAM2 性能比较，展示改进算法性能提升。性能提升衡量的具体计算公式为：

$$\eta = \left(1 - \frac{\beta}{\alpha}\right) \times 100\% \tag{8}$$

其中，η 表示性能提升程度；α 表示 ORB-SLAM2 算法的轨迹误差；β 表示本小节改进后算法的轨迹误差。

在 ORB 特征点提取过程中，特征点的识别具有一定的随机性，这导致每次特征匹配结果都有所不同。所以对相同数据序列进行多次实验时，会产生细微的定位差异。为了提升实验结果的准确性，本节对每个数据序列进行 5 轮实验，并采用这些实验结果的平均值作为评估的最终指标。虽然单次实验结果存在差异，但认为这些差异对整体定位精度的影响微乎其微。通过与 ORB-SLAM2 算法的比较分析，本节特别强调了在绝对轨迹误差(ATE)和相对位姿误差(RPE)两个关键性能指标上的对比结果。ATE 的计算公式如下：

$$F_i = Q_i^{-1} S P_i \tag{9}$$

其中，F_i 表示 i 时刻 ATE 的值；Q_i 表示 i 时刻的实际轨迹；P_i 表示 i 时刻算法估计的轨迹。RPE 的计算公式为：

$$E_i := \triangle Q_{i+\triangle t}^{-1} \triangle P_{i+\triangle t} = (Q_i^{-1} Q_{i+\triangle t})^{-1} (P_i^{-1} P_{i+\triangle t}) \tag{10}$$

为了验证算法的有效性，选择了无动态物体的静态场景以及低动态和高动态场景进行测试。具体的实验成果详见表 1、表 2，展现了 2 种算法性能的直接对比。其中，RMSE 表示均方根误差，S. D. 表示标准差。

表 1　绝对轨迹误差对比表

Table 1　Comparison of absolute trajectory error results

数据集		ORB-SLAM2		本文算法		性能提升	
		RMSE	S. D.	RMSE	S. D.	RMSE/%	S. D./%
静态	fr1_desk	0.015 4	0.008 6	0.017 8	0.008 8	-15.44	-2.14
	fr1_room	0.053 7	0.027 6	0.051 3	0.026 9	4.31	2.37
低动态	fr3_sitting_tpy	0.025 2	0.013 5	0.022 6	0.012 7	10.42	5.53
	fr3_sitting_static	0.007 6	0.004 0	0.006 4	0.003 6	15.58	8.62
高动态	fr3_walking_tpy	0.155 7	0.166 7	0.075 2	0.078 9	51.67	52.64
	fr3_walking_static	0.022 5	0.012 4	0.007 1	0.003 4	68.34	72.58

表 2　相对位姿误差对比表

Table 2　Comparison of relative pose errors results

数据集		ORB-SLAM2		本文算法		性能提升	
		RMSE	S. D.	RMSE	S. D.	RMSE/%	S. D./%
静态	fr1_desk	0.009 5	0.005 8	0.009 3	0.005 7	1.43	1.51
	fr1_room	0.011 3	0.007 9	0.011 0	0.007 6	2.35	2.76
低动态	fr3_sitting_tpy	0.014 7	0.009 8	0.013 0	0.009 2	11.53	5.64
	fr3_sitting_static	0.005 4	0.002 7	0.004 4	0.002 5	17.68	7.21
高动态	fr3_walking_tpy	0.027 7	0.018 1	0.010 4	0.007 2	62.35	59.78
	fr3_walking_static	0.031 5	0.021 3	0.008 3	0.005 4	73.55	74.61

通过对表 1、表 2 中的数据分析，在处理高动态环境下的序列时，本小节所提出的改进算法相较于 ORB-SLAM2 算法，在绝对轨迹误差的均方根误差上平均实现了 60.01%的显著降低，同时，其标准差亦减少了 62.61%。这些结果充分证实了在高动态场景处理中，本算法在稳定性和准确度方面得到了

显著提高。而在低动态场景的序列处理上,相比 ORB-SLAM2 算法,本算法在 ATE 的均方根误差方面平均降低了 5.57%,标准差降低了 0.12%。此结果表明,尽管改进后的算法能有效地将场景中轻微移动的动态对象识别为异常值而进行排除,优化了算法性能,但在低动态场景下的性能提升并不如高动态场景那样显著。

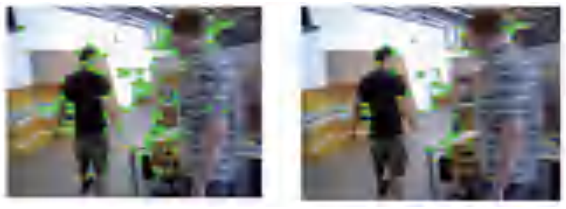
针对静态场景,所提出的改进算法与 ORB-SLAM2 算法的性能基本持平,有时略显逊色。这一现象可能源于图像处理过程中的语义分割误差,错误地将静态物体识别为动态物体,进而一定程度上影响了 SLAM 算法的定位精度。

表 2 的数据对比展现了在高动态环境中,本小节提出的算法与 ORB-SLAM2 相比,在相对位姿误差方面取得了显著的性能提升。这一成果进一步强化了本算法在处理动态性显著的场景下的优势,并有效提升了定位精度及地图构建的质量。然而,在低动态场景下,尽管本研究的算法依旧表现出了性能上的轻微提升,这种改进并不像在高动态场景下那般突出。这一现象可能源于低动态场景中动态物体的影响较为有限,而 ORB-SLAM2 算法已经在这种环境下展现出较为优异的处理能力,导致本研究算法的优势未能充分显现。对于静态场景而言,本研究算法与 ORB-SLAM2 的性能表现相似,说明在无动态物体干扰的环境中,2 个算法在维持 SLAM 系统的稳定性和准确性方面具有相当的效能。

通过对比剔除动态特征点前后的图像(如图 7 所示),可以直观地观察到算法在不同动态环境下的表现和效果。图 7 中,左列图片为原始算法的效果图,右列是改进后的算法效果图。

(a) 静态场景特征点对比图

(b) 低动态场景特征点对比图



(c) 动态场景特征点对比图

图 7 动态特征点剔除对比效果图

Fig. 7 Diagram of dynamic feature point filtering comparison

7 结束语

本文在已有的 ORB-SLAM2 算法的基础上加以改进,提出了一种剔除动态特征点的 SLAM 算法。首先对于 ORB-SLAM2 的整体框架进行了简要的介绍,然后使用 LK 光流法结合 YOLO 检测动态目标法剔除建图中的动态特征点。实验结果表明,在处理高动态环境下的序列时,本文所提出的改进算法相较于 ORB-SLAM2 算法,在绝对轨迹误差的均方根误差方面平均实现了 60.01%的显著降低,同时,其标准差亦减少了 62.61%。这些结果充分证实了在高动态场景处理中,本算法在稳定性和准确度方面的显著提高。而在低动态场景的序列处理上,相比 ORB-SLAM2 算法,本算法在 ATE 的均方根误差方面平均降低了 13.00%,标准差降低了 7.08%。此结果表明,尽管改进后的算法能有效地将场景中轻微移动的动态对象识别为异常值而进行排除,算法性能也得以优化,但在低动态场景下的性能提升并不如高动态场景那样显著。

参考文献

[1] 李延真,石立国,徐志根,等. 移动机器人视觉 SLAM 研究综述[J]. 智能计算机与应用,2022,12(7):40-45.

[2] 陈文佑,章伟,胡陟,等. 一种融合深度相机与激光雷达的室内移动机器人建图与导航方法[J]. 智能计算机与应用,2021,11(4):159-163.

[3] 曹昊哲,刘全攀. 基于半直接法的无人集群协同视觉 SLAM 算法[J]. 兵工学报,2023,44(11):3345-3358.

[4] 吴昊,王浩,苏醒,等. 自动驾驶系统中视觉感知模块的安全测试[J]. 计算机研究与发展,2022,59(5):1133-1147.

[5] 李勇,吴海波,李万,等. 动态场景下基于加权静态的视觉 SLAM 算法[J]. 激光与光电子学进展,2024,61(4):513-522.

[6] 罗彤,骆云志,沈明川,等. 面向动态物体场景的视觉 SLAM 方法[J]. 兵工自动化,2023,42(4):93-96.

[7] 朱东莹,钟勇,杨观赐,等. 动态环境下视觉定位与建图的运动分割研究进展[J]. 计算机应用,2023,43(8):2537-2545.

[8] 王朋,郝伟龙,倪翠,等. 视觉 SLAM 方法综述[J]. 北京航空航天大学学报,2024,50(2):359-367.

[9] 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.
[10] WANG Youbing，HUANG Shoudong. Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios[C]//Proceedings of 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). Piscataway，NJ：IEEE，2014：1841-1846.
[11] SUN Yuxiang，LIU Ming，MENG Q H. Improving RGB-D SLAM in dynamic environments：A motion removal approach[J]. Robotics and Autonomous Systems，2016，89：110-122.
[12] DAI Weichen，ZHANG Yu，LI Ping，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.
[13] 田瑞，张云洲，杨凌昊，等. 物体级语义视觉 SLAM 研究综述[J]. 控制理论与应用，2023，40(12)：2160-2171.
[14] 石征锦，王晟霖，武晨，等. 基于深度学习的实时同步定位与建图算法研究[J]. 信息技术与信息化，2024(1)：207-211.
[15] 田瑞，张云洲，杨凌昊，等. 物体级语义视觉 SLAM 研究综述[J]. 控制理论与应用，2023，40(12)：2160-2171.
[16] BRASCH N，BOZIC A，LALLEMAND J，et al. Semantic monocular SLAM for highly dynamic environments [C]// Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway，NJ：IEEE，2018：393-400.
[17] LIANOS K N，SCHONBERGER J L，POLLEFEYS M，et al. Vso：Visual semantic odometry[C]//Proceedings of the European Conference on Computer Vision (ECCV). Cham：Springer，2018：234-250.
[18] 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.
[19] YU Chao，LIU Zuxin，LIU Xinjun，et al. DS-SLAM：A semantic visual SLAM towards dynamic environments[C]//Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway，NJ：IEEE，2018：1168-1174.
[20] HENEIN M，ZHANG Jun，MAHONY R，et al. Dynamic SLAM：The need for speed[C]//Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA). Piscataway，NJ：IEEE，2020：2123-2129.
[21] ZHANG Tianwen，ZHANG Huayan，LI Yang，et al. Flowfusion：Dynamic dense RGB-D slam based on optical flow[C]//Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA). Piscataway，NJ：IEEE，2020：7322-7328.