DOI:10.19651/j.cnki.emt.2518163

基于深度约束与光流跟踪的视觉 SLAM 方法*

尹显波 王中元
(中国矿业大学环境与测绘学院 徐州 221116)

摘 要:同步定位与建图(SLAM)是机器人自主导航的关键,然而传统的 SLAM 系统通常是针对静态环境设计的,当存在动态物体时,动态特征点会造成错误的数据关联从而导致精度和可靠性降低。并且当前的解决方案中依旧存在潜在动态对象无法检测,且动态对象占据主体时所保留的有用特征点不足等问题。为了克服这些限制,提出了一种基于 ORB-SLAM2 的视觉 SLAM 系统。首先利用 yolov8 目标检测提供语义信息,结合深度信息进行深度约束生成动态掩码;然后,基于动态概率进行特征点的四叉树均匀分配,剔除动态特征点的同时保留更多有用特征;最后,利用光流跟踪检测与剔除潜在动态对象上的特征点。其中动态掩码与关键帧结合实现运动分割,构建干净的密集点云地图。在 TUM 和 Bonn 数据集下的实验结果表明,相比于 ORB-SLAM2,在高度动态场景中平均定位精度提高超过 90%,并且在相对静止的场景中表现依旧可靠。此外,在保持实时运行的同时其性能对于当前同类别的先进方法也能有所提升。

关键词:视觉 SLAM;动态环境;目标检测;深度约束;四叉树分配;光流跟踪
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

Visual SLAM approach based on depth constraints and optical flow tracking

Yin Xianbo Wang Zhongyuan
(School of Environment and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: Simultaneous localization and mapping (SLAM) is the key to autonomous robot navigation. However, traditional SLAM systems are typically designed for static environments, when dynamic objects are present, dynamic feature points can lead to incorrect data associations, reducing accuracy and reliability. Existing solutions still face challenges such as undetected potentially dynamic objects and an insufficient number of useful feature points when dynamic objects dominate the scene. To overcome these limitations, this study proposes a vision SLAM system based on ORB-SLAM2. Firstly, yolov8 object detection is utilized to provide semantic information, which is combined with depth information for depth constraints to generate dynamic masks; next, a quadtree-based uniform allocation of feature points is implemented based on dynamic probability, ensuring the removal of dynamic feature points while preserving more useful features; finally, optical flow tracking is utilized to detect and reject feature points on potentially dynamic objects. In which the dynamic mask is combined with keyframes to realize motion segmentation, thus constructing clean and dense point cloud maps. Experimental results on the TUM and Bonn datasets demonstrate that, compared to ORB-SLAM2, the average localization accuracy improves by over 90% in highly dynamic scenes while maintaining reliable performance in relatively static environments. Additionally, the improved system achieves real-time performance and outperforms other state-of-the-art methods in its category.

Keywords: visual SLAM; dynamic environment; object detection; depth constraints; quadtree assignment; optical flow tracking

0 引 言

智能移动机器人能够在未知环境中稳健运行需同时具备环境感知和自我位置估计的能力。同步定位与建图[1](simultaneous localization and mapping,SLAM)就是其中的核心技术之一,其利用传感器数据,使得机器人能够同时

收稿日期:2025-02-24
*基金项目:国家自然科学基金(42274048)、江苏省重点研发计划项目(BE2022716)、中国矿业大学研究生创新计划项目(2025WLJCRCZL219)资助

确定自身位置并构建环境地图。这一重要技术在自动驾驶、机器人导航和虚拟现实(virtual reality,VR)等领域得到了广泛的应用。SLAM 技术按照传感器类别又可细分为激光 SLAM[2]和视觉 SLAM。其中视觉 SLAM 以相机为主要传感器,因其低成本和丰富的信息获取能力受到了广泛关注,并且随着计算机视觉技术的进步,实际应用中对场景环境感知和语义信息需求的快速增加,视觉 SLAM 变得尤为重要,从而衍生出了许多优秀的开源方法如 ORB-SLAM2[3]、LSD-SLAM[4]、DSO[5]等,然而,这些方法都是建立在假设环境是静态的或运动速度很低的情况下。但是这一假设在现实场景中根本不成立,因为在现实环境中,行人、车辆等动态物体无处不在。所提取到的动态特征点会引入显著的相机位姿估计误差,导致定位失败从而影响后续的地图构建。为了解决动态环境带来的挑战,提高 SLAM 系统的稳定性和准确性,各种解决方案应运而生。

在传统的几何方法中,几何约束模型是基于静态特征构建的,动态特征与静态特征相比存在显著偏差,因此得以区分,如 ORB-SLAM2 利用随机样本一致性(random sample consensus,RANSAC)方法对动态特征点进行过滤,但是当动态对象占据主体时初始转换估计将由动态特征决定,效果不佳。Cheng 等[6]提出一种新颖的稀疏运动移除(sparse motion removal,SMR)模型,该模型基于贝叶斯框架检测输入帧的动态和静态区域,通过考虑连续帧之间的相似性和当前帧与参考帧之间的差异来区分动态点和静态点。文献[7]使用 Delaunay 三角剖分从所有地图点构建稀疏图,利用地图点之间的相关性,将属于静态场景的特征点与属于不同运动物体的特征点分离。Du 等[8]利用 Graph-Cut RANSAC[9]将动态点与静态点区分并进行初始姿态估计,然后通过条件随机场根据长期一致性进一步移除动态点。以上方法具有良好的实时性和低资源消耗,但是解决问题的能力有限。

随着计算机视觉的快速发展,研究人员逐渐将目标检测和语义分割技术应用到 SLAM 中。如基于语义分割的 DynaSLAM[10]系统利用 Mask-RCNN 生成动态对象的掩码并结合多视角几何的方法剔除动态点,显著提升了系统性能,但是语义分割耗时较长,同时剔除的特征点过多易造成跟踪丢失。DS-SLAM[11]结合了 SegNet[12]语义分割网络和移动一致性检查方法,以减少动态物体的影响,从而在动态环境中提高定位精度。然而,仅通过排除动态特征点来构建密集地图仍会产生噪声。为了解决这一问题,Xie 等[13]利用深度信息掩码修复算法来解决语义分割网络中的欠分割问题实现运动对象分割,并将该算法与 Lucas-Kanade(LK)光流技术相结合提高对移动物体的检测能力。SEG-SLAM[14]构建了动态物体的目标检测与语义分割融合模块,并采用极线几何技术与深度信息来制定特定的动态特征点剔除策略,在动态场景中定位与地图构建的准确性上展现了显著的提升,但是这种方法的资源消耗较大。

另一方面,基于目标检测的方法相对于语义分割具备不错的实时性,COEB-SLAM[15]利用目标检测结合图像模糊作为约束识别动态物体,然后使用极线约束去除动态特征点。Sun 等[16]利用轻量化目标检测生成语义信息,结合几何约束与基于特征点深度的 RANSAC 有效减轻了动态特征的影响。文献[17]采用改进的目标检测网络 YOLOv5s 以及动态逻辑决策消除动态特征点,随后仅利用静态特征点进行姿态估计与地图构建。但是以上方法仍存在潜在动态对象无法检测与剔除问题,影响系统的定位精度。

当前的大多数方法都是特征提取与均匀分配后,将不满足条件的动态特征点,但是当动态对象占据主体时会导致保留的有用特征太少,易造成跟踪丢失问题。针对以上不足,提出一种基于深度约束与光流跟踪的视觉 SLAM 系统,主要内容如下:

1)利用 YOLOv8 进行目标检测,同时结合深度差异分析实现深度约束,以生成动态掩码。此外,对掩码进行膨胀处理,扩展边界区域,从而更精确地分割动态对象。

2)基于动态掩码与上一帧光流检测结果,提出一种基于动态概率的四叉树均匀分配法,在剔除动态特征的同时保留更多静态特征点。

3)针对潜在的动态对象无法检测与剔除问题,利用稀疏光流跟踪的方法按照动静态特征点的光流向量差异进行再剔除,提升系统的定位精度。

4)关键帧图像与对应的动态掩码结合实现运动分割,并传入密集建图中构建干净的密集环境地图,解决点云中的运动漂移问题。

1 系统框架

本文所提出的算法是基于传统的 ORB-SLAM2 进行改进的,其在静态环境的假设下具有优异的性能。ORB-SLAM2 主要包含 3 个并行线程:跟踪、建图、闭环检测,3 个线程的结合可以实现相机轨迹和地图的实时全局一致性。本文的改进则是在其基础上增加了目标检测线程和密集建图线程。此外,在跟踪线程中引入特征点基于动态概率的四叉树均匀分配法,同时增加稀疏光流跟踪模块检测与剔除潜在动态对象。系统概述如图 1 所示,嵌入两个额外线程并改进跟踪线程的目的是为了过滤掉不稳定的动态特征点,并保留足够的静态特征点用来估计相机位姿,同时保证后续的跟踪与建图稳健进行。

在目标检测线程中使用 TensorRT 加速的 yolov8 对每帧图像进行目标检测以提取对象的语义信息,保证系统能够实时运行,对象被划分为动态对象、潜在动态对象、静止对象 3 类。其中动态对象包括行人、车和动物等,潜在动态对象指通常静止但在外力作用下可能移动的物体(如箱子、椅子),而静止对象则为始终保持不动的物体(如桌子、空调)。

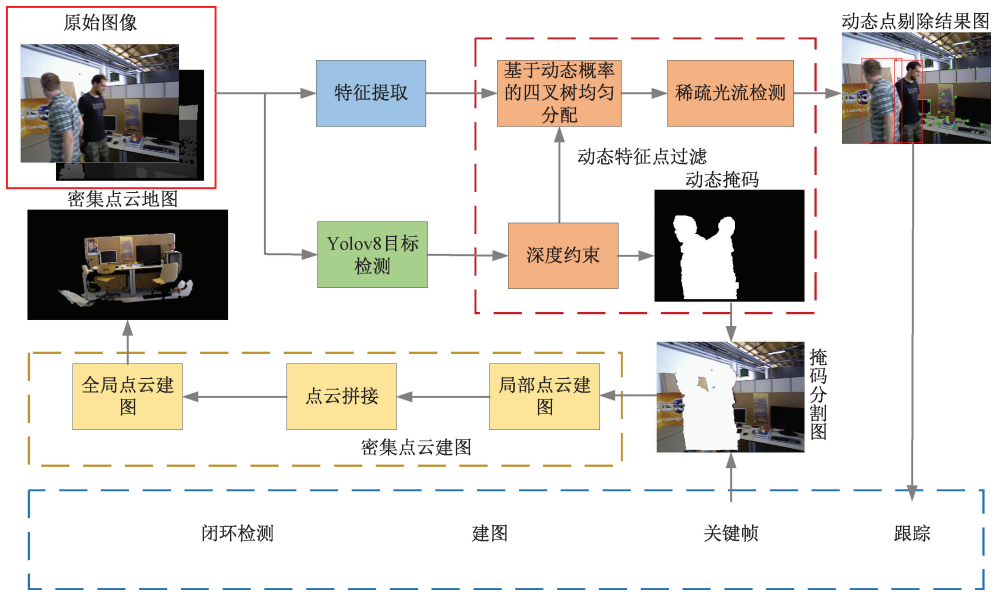


图 1　系统框架

Fig. 1　System framework

在跟踪线程中，首先根据语义信息以及深度信息差异实现深度约束生成动态掩码，然后将图像以及动态掩码传入高斯金字塔中，进行特征提取，基于动态概率进行特征点的四叉树均匀分配，避免动态特征点占据过多四叉树节点从而保留更多有用的静态特征点。此外，利用稀疏光流跟踪法检测潜在动态对象特征点的运动状态，并剔除动态特征点，从而仅利用静态特征点进行后续的跟踪、建图以及闭环检测。

在密集建图线程中，将关键帧图像以及对应的动态掩码相结合实现运动分割，将分割后的图像作为输入构建局部点云地图，接着根据解算的相机位姿进行点云配准，最后得到干净的全局密集点云地图。

2　基于深度约束的动态检测

2.1　YOLOv8 目标检测

YOLO(you only look once)[18] 系列是目标检测算法的绝佳示例，在计算机视觉领域取得了巨大的成功。基于此，研究人员对该方法进行了改进并添加了新的模块，提出了许多经典模型。YOLOv8 是 Ultralytics 公司于 2023 年 1 月 10 日发布的一种算法。与 YOLO 系列以前的优秀型号(如 YOLOv5 和 YOLOv7)相比，YOLOv8 是具有更高检测精度和速度的先进模型。YOLOv8 的整体结构主要由输入端、Backbone、Neck、Head 组成。输入端采用了增强的数据处理技术，对背景和小目标进行增强，通过随机缩放、裁剪和排列图像来提高检测效果。根据网络的宽度和深度，YOLOv8 可以分为 n、s、m、l、x 等多个版本，综合比较精度与检测速度，选择 YOLOv8l 作为本文的目标检测模型。此外 YOLOv8 能够检测 80 种预定义的常见类别，如人、车辆、动物、家具，涵盖了日常生活中的大部分常见物体。为了提升系统的实时性，本文使用 NVIDIA GPU 的 TensorRT 加速方法将 YOLOv8 的原始网络模型(Pytorch 格式)转换为高度优化的 TensorRT 引擎(Engine 格式)，其中包括将 YOLOv8 的 Backbone 和 Neck 中的卷积层、归一化层和激活函数融合为复合层，减少 GPU 内核的调用次数，提升并行效率，并将模型权重和激活值从 FP32 转换为 FP16，降低内存占用和计算量，转换后的模型以 C++接口的方式调用，更加方便部署到 SLAM 系统中，从而达到加速的效果。

2.2　深度约束

利用上述详细介绍的目标检测模块，可以提取语义信息，包括类别和边界框坐标。Dynamic-SLAM[19] 采用目标检测识别环境中的移动目标，并直接移除动态边界框内的所有特征点，以提高系统精度。然而，这种策略会将除框内静态特征点作为异常值剔除，影响系统精度。

YOLO-SLAM[20] 采用深度一致性约束计算深度阈值以分离动态目标框内被错误分类的静态特征点，然而，由于深度相机的采集误差，对象边界上的特征点难以被准确分类，同时无法解决密集点云中的漂移问题。针对上述问题，本文提出一种改进方案，即基于深度约束的动态掩码分割方法，并对生成的掩码进行膨胀处理，以更全面地覆盖动态区域。如图 2 所示，RGB-D 相机捕获的深度图像根据深度差异清晰地描绘了物体轮廓，行人与背景之间的深度具有明显区别。基于此，本文使用深度约束生成动态掩码，具体过程如算法 1 所示。

首先，输入预定义的动态边界框信息和当前帧的深度图像。然后统计边界框内像素的深度值，接着计算平均深

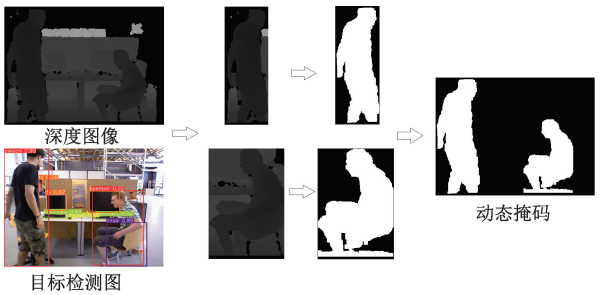


图 2　深度约束示意图

Fig. 2　Depth constraints schematic

算法 1 深度约束

输入:当前帧的深度图像 D,当前帧的动态边界框 Box,深度阈值 d_ϕ;

输出:当前帧的动态对象掩码 $Mask$;

1. $\overline{d}$ =CalculateMeanValue(Box, D);
2. σ =CalculateMeanDifference($\overline{d}$, Box, D);
3. for each pixel point i in Box do:
4.　if($d_i < d_\phi$)then
5.　　$Mask_i$ =1;
6.　else
7.　　$Mask_i$ =0;
8.　end if
9. end for

度,如式(1)所示。

$$\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{1}$$

其中,n 表示边界框内有效像素的数量,d_i 表示深度图中像素 $i=(u_i,v_i)$ 处的深度值,$\overline{d}$ 代表边界框内的平均深度。由于边界框中的大多数像素对应于人体,因此平均深度可以较为准确地近似人体所在位置的深度。然后计算平均深度的标准差,如式(2)所示。

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i-\overline{d})^2} \tag{2}$$

其中,σ 表示标准差,反映深度值的偏差。在进入循环后,对边界框内的所有像素进行遍历,并对深度差值落在预设阈值内的像素标记为掩码点,将其掩码值设为 1。这表示这些位置的特征点是动态的,在后续步骤中将被移除。相反,超出该阈值的点被赋予掩码值 0,表示其为稳定的背景点。计算过程如式(3)所示。

$$Mask = \begin{cases} 1, & d_i < d_\phi \\ 0, & d_i > d_\phi \end{cases} \tag{3}$$

其中,d_ϕ 表示深度阈值,由式(4)确定。

$$d_\phi = \overline{d} + \partial \times \sigma \tag{4}$$

其中,∂ 是一个自定义阈值,经过大量的实验评估后,最终设定为 0.6。按照上述步骤,可在检测框内获得动态掩码。当存在多个动态检测框时,需要分别处理每个掩码并将其拼接在一起,因为不同动态对象的位置可能导致深度信息存在显著差异。最终获得完整的动态掩码,如图 2 最后一列所示。

2.3　基于动态概率的四叉树均匀分配

通过上述深度约束,可以生成动态对象的掩码,通过判断经特征提取与均匀分配后的特征点是否落入掩码区域,可以剔除动态特征点,这是目前最常用动态特征剔除策略,如 DynaSLAM、先前的研究[21]等系统中都有使用。然而当场景中动态对象占据主体时,如场景中出现多个行人以及行人从镜头前走过等情况,大部分提取的特征分布在动态对象上,此时,传统的掩码剔除方法会导致有效特征点过少,进而引发跟踪丢失并严重影响系统的稳定性。因此本文提出一种基于动态概率的四叉树均匀分配法,其流程如图 3 所示。ORB-SLAM2 系统在跟踪线程中首先对高斯金字塔中每一层图像提取特征点并进行四叉树均匀分配,待全部特征提取完毕之后再生成描述子。

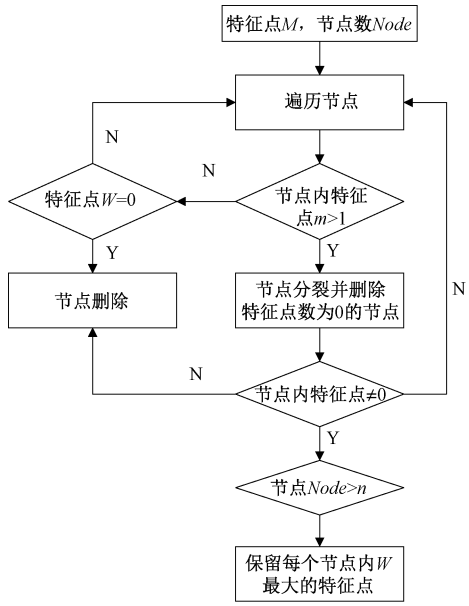


图 3　改进四叉树分配算法流程图

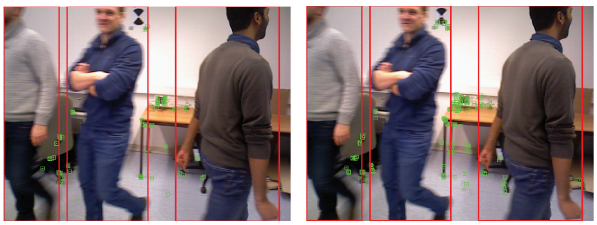
Fig. 3　Flowchart of improved quadtree distribution algorithm

而本文先将上述的动态掩码传入高斯金字塔中,在特征点的均匀分配前,根据式(5)计算特征点基于动态概率的权值 W:

$$W = (1-H_{po}) \times \left(1-\frac{1}{2}H_{pr}\right) \times w_r \tag{5}$$

其中,H_{pr} 为先验的动态概率,由上一帧中的光流检测决定,默认情况下为 0,H_{po} 为检测框以及动态掩码决定,当特征点位于掩码上时为 1,特征点不在掩码上但检测类别为潜在动态对象时为 0.5,其他为 0,w_r 为特征响应值。因此当出现动态对象或潜在动态对象时,其对应的特征点权值 $W=0$ 或 $W \leqslant 0.5w_r$。

由于节点数阈值 N 在每次分配时是固定的，因此在节点分裂前，优先剔除特征点权值为 0 的点所在节点，减少动态特征点的分配开销以及占据的四叉树节点。当分裂后的节点数满足阈值 N 后，优先保留各节点内权值 W 最大的特征点，通过图 4 的对比可以看出，基于动态概率的四叉树均匀分配策略所保留的静态特征点显著多于直接分配后通过掩码剔除动态特征的方式，意味着系统能够提取更多稳定且有效的环境信息，从而避免因特征点不足所造成的跟踪丢失问题，并显著提升系统的定位精度。



(a) 改进前　(b) 改进后
(a) Before improvements　(b) After improvements

图 4　改进四叉树前后对比图

Fig. 4　Comparison before and after improved quadtree

3　基于光流跟踪的动态检测

通过上述方法能够剔除场景中的预定义动态对象的特征点，然而在某些情况下，仍存在未被检测或剔除的潜在动态对象特征点从而影响系统性能。为进一步提高动态特征点的检测能力，本文引入 LK 光流跟踪方法进行特征检测，该方法是基于灰度不变性假设的，即空间中同一点在不同时刻、不同图像上的像素灰度值是固定不变的，则有：

$$I(x+\mathrm{d}x,y+\mathrm{d}y,t+\mathrm{d}t)=I(x,y,t) \tag{6}$$

对左边进行泰勒展开并保留一阶项，同时左右相减可以得到：

$$\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial t}\mathrm{d}t=0 \tag{7}$$

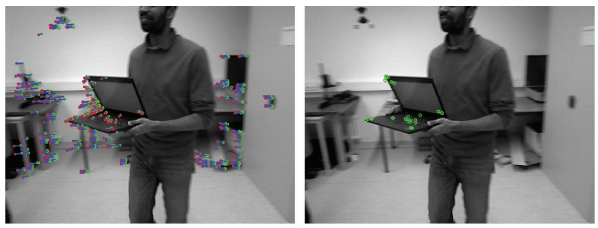
两边同时除以 $\mathrm{d}t$ 得到：

$$\frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}=-\frac{\partial I}{\partial t} \tag{8}$$

式中：$\mathrm{d}x/\mathrm{d}t$、$\mathrm{d}y/\mathrm{d}t$ 分别表示像素点在 x 轴和 y 轴上的移动速度，$\partial I/\partial x$、$\partial I/\partial y$ 分别表示图像在该点处的 x 轴和 y 轴上的梯度。然而仅靠一个点无法计算出像素点运动，因此假设该点所在的 $n\times n$ 窗口内的像素点具有相同的运动，可以得到 n^2 个方程：

$$\left[\frac{\partial I}{\partial x}\ \frac{\partial I}{\partial y}\right]_k\begin{bmatrix}\dfrac{\mathrm{d}x}{\mathrm{d}t}\\[2ex]\dfrac{\mathrm{d}y}{\mathrm{d}t}\end{bmatrix}=-\frac{\partial I}{\partial t_k},k=1,\cdots,n^2 \tag{9}$$

接着通过求解最小二乘法就可以得到像素点的运动，从而追踪到该点在下一帧图像中的位置，如图 5(a)所示，用不同颜色表示前后帧的特征点并将两者相连。



(a) 光流跟踪点　(b) 异常点
(a) Optical flow tracking points　(b) Anomalous points

图 5　光流检测结果图

Fig. 5　Graph of optical flow detection results

然而，动态特征点与静态特征点的移动通常存在显著差异，因此可以通过比较特征点的光流向量来检测动态特征点，首先计算特征点的移动向量 $\boldsymbol{F}$：

$$\boldsymbol{F}=(\mathrm{d}x,\mathrm{d}y)=(x_{t+1}-x_t,y_{t+1}-y_t) \tag{10}$$

采用直方图统计的形式统计移动向量得出静态背景的运动模型 $\boldsymbol{F}_m$。然后分别计算各个光流向量与主光流向量之间的模长差 ΔL_i 和夹角 θ_i：

$$\Delta L_i=\sqrt{\mathrm{d}x_m^2+\mathrm{d}y_m^2}-\sqrt{\mathrm{d}x_i^2+\mathrm{d}y_i^2} \tag{11}$$

$$\theta_i=\arccos\left(\frac{\mathrm{d}x_i\mathrm{d}x_m+\mathrm{d}y_i\mathrm{d}y_m}{\sqrt{\mathrm{d}x_i^2+\mathrm{d}y_i^2}\sqrt{\mathrm{d}x_m^2+\mathrm{d}y_m^2}}\right) \tag{12}$$

其中，$(\mathrm{d}x_i,\mathrm{d}y_i)$ 表示第 i 个光流向量 $\boldsymbol{F}_i$，接着分别比较模长差 ΔL_i 和夹角 θ_i 的绝对值与设定阈值之间的大小关系判断该点是否为动态点，如式(13)所示。

$$H_{pr}=\begin{cases}1,\ |\ \Delta L_i\ |>\xi\\1,\ |\ \theta_i\ |>\gamma\\0,\text{其他}\end{cases} \tag{13}$$

式中：ξ 表示距离阈值，γ 表示夹角阈值，经过多次实验分析与统计，本文中 ξ 设为 2，γ 设为 10°。其中检测到的异常点如图 5(b)所示。最后将 H_{pr} 为 1 的特征点剔除，保留静态特征点进行后续的跟踪与优化。

4　实验与分析

4.1　实验设备

本研究的实验均在个人笔记本电脑上运行，该电脑配备 8 GB 内存、NVIDIA RTX 4060 GPU 和 Core i9-13900HX 处理器，操作系统为 Ubuntu 20.04。其中 GPU 加速仅用于目标检测阶段，目标检测使用 yolov8l 模型，通过 TensorRT 转化为 Engine 格式，并以 C++接口形式部署到 SLAM 中，避免了 python 与 C++之间复杂通信问题。

4.2　数据集

TUM RGB-D 数据集广泛用于 SLAM 研究，由 Kinect 相机采集，包括深度图像、RGB 图像和真实轨迹的相机运动数据，涵盖 halfsphere、rpy、static 和 xyz 四种相机运动模式，并根据人物状态划分为高动态环境和低运动环境。相比之下，Bonn RGB-D 数据集包含 24 个动态序列，使用 Optitrack Prime 13 运动捕捉系统录制，包含更多运动物体，因此，在 TUM 数据集上表现良好的方法在该数据集下

可能效果不同。本文选择了 9 个具有显著运动特征的序列进行实验,其中包括 crowd(多人随机行走)、moving_no_box(搬运箱子)、person_tracking(相机跟随行人)和 synchronous(相机固定,两人同步移动)等场景。

4.3 轨迹误差评估

为了定量评估算法的有效性,本文使用 EVO 评估工具对估计轨迹进行分析,并选取绝对轨迹误差(absolute trajectory error, ATE)和相对位姿误差(relative pose error,RPE)作为评估指标,ATE 直接衡量估计轨迹与真实轨迹之间的偏差,RPE 衡量短时间间隔内两帧之间的位姿变化精度。为了更直观地量化误差,分别以均方根误差(root mean square error, RMSE)、标准差(standard deviation,S. D.)作为评价指标,以衡量误差的平均水平及其波动性。

表 1 列出了本文方法以及基准方法 ORB-SLAM2 在 TUM 数据集各序列下的 ATE 的 RMSE 和 S. D. 及精度提升比率,表 2 分别展示了两种方法的相对平移误差(relative translation error, RTE)以及相对旋转误差(relative rotation error,RRE)。从实验结果可以看出,相较于 ORB-SLAM2,本文方法在高度动态序列下取得了显著改进,其中 ATE 的 RMSE 平均降低 95.85%,同时 RTE 和 RRE 的 RMSE 也分别降低 93.30%和 91.01%。尽管 ORB-SLAM2 在相对静止的序列下可凭借 RANSC 有效剔除离群点,表现出较好的稳定性,但实验表明,本文方法在相对静止的序列下仍能进一步提升精度,展现出更优的适应性与鲁棒性。

表 1 TUM 数据集的绝对轨迹误差(ATE)结果

Table 1 Results of absolute trajectory error(ATE)on TUM datasets m

序列	ORB-SLAM2		本文		提升率/%	
	RMSE	S. D.	RMSE	S. D.	RMSE	S. D.
fr3_s_static	0.007 6	0.003 9	0.005 8	0.002 9	23.68	25.64
fr3_w_xyz	0.675 2	0.321 7	0.013 9	0.006 7	97.94	97.92
fr3_w_static	0.309 7	0.152 7	0.006 7	0.003 1	97.84	97.97
fr3_w_rpy	0.658 3	0.285 3	0.031 4	0.016 8	95.23	94.11
fr3_w_half	0.311 4	0.132 7	0.023 7	0.011 3	92.39	91.48

表 2 TUM 数据集的相对平移误差(RTE)和相对旋转误差(RRE)结果

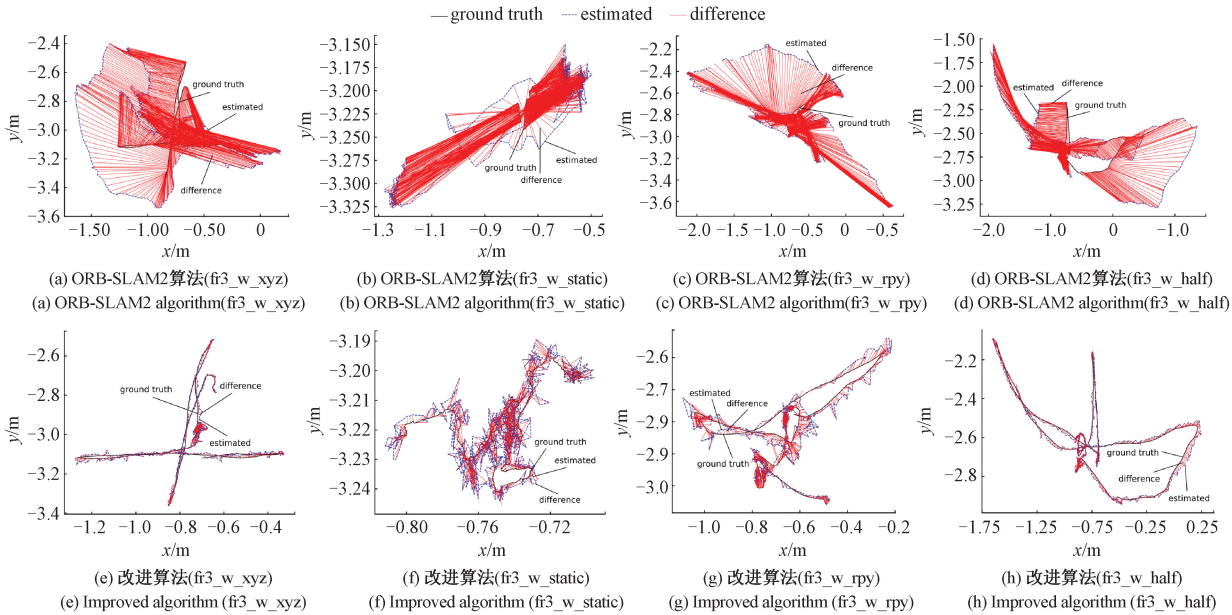
Table 2 Results of relative translation error(RTE)and relative rotation error(RRE)on TUM datasets

(m/s,(°)/s)

序列	ORB-SLAM2		本文		提升率/%	
	RTE	RRE	RTE	RRE	RTE	RRE
fr3_s_static	0.008 7	0.275 4	0.006 9	0.268 1	20.69	2.65
fr3_w_xyz	0.402 1	7.909 9	0.018 2	0.594 4	95.47	92.49
fr3_w_static	0.188 8	3.424 5	0.009 1	0.252 4	95.18	92.63
fr3_w_rpy	0.353 7	6.961 7	0.041 5	0.867 5	88.27	87.54
fr3_w_half	0.415 8	8.385 3	0.023 8	0.724 6	94.28	91.36

如图 6 所示,为了更直观地体现所提出算法的有效性,本文对比了两种方法在高度动态序列下生成的绝对轨迹误差图,其中黑线表示真实轨迹,蓝色虚线表示估计轨迹,红色细线表示真实轨迹与估计轨迹之间的误差。从图中可以看出,ORB-SLAM2 存在明显的轨迹误差,而本文方法在所有序列中的相机运动轨迹估计更加准确,与地面真实轨迹高度一致。此外,图 7 展示了相机相对平移误差随时间的变化,在大多数时间段内,本文方法的相对平移误差更低,进一步验证了其在动态环境中能够有效减少动态对象的干扰,提升定位精度和鲁棒性。

表 3 对比了本文方法与 YOLO-SLAM、SG-SLAM[22]、COEB-SLAM、文献[16]在 TUM 各序列下绝对轨迹误差的 RMSE,以上所列均是动态环境下基于目标检测进行改进的视觉 SLAM 方法。为了更直观地展示性能差异,将每个序列下最优方案的数据用黑色加粗表示,次优的方案用下划线表示。从结果可以看出,相较于 YOLO-SLAM、SG-SLAM、COEB-SLAM,本文算法在所有序列中的误差均最小,且相对于 3 种方法中的最优解误差分别降低了 0.002、0.007、0.004、0.010、0.031 m。与文献[16]相比,本文算法在大多数序列中表现最优,仅在 fr3_w_rpy 序列中略逊一筹,达到次优的效果。这可能是由于深度阈值在不同场景中的适用性存在差异,导致某些静态



(a) ORB-SLAM2算法(fr3_w_xyz)　(b) ORB-SLAM2算法(fr3_w_static)　(c) ORB-SLAM2算法(fr3_w_rpy)　(d) ORB-SLAM2算法(fr3_w_half)

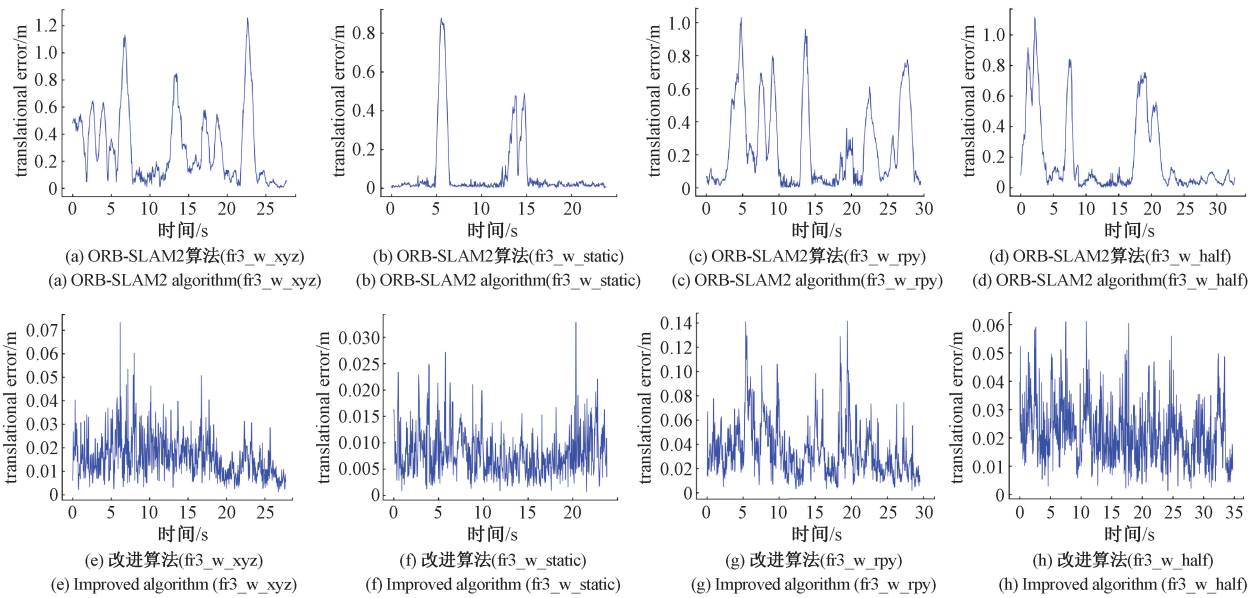
(a) ORB-SLAM2 algorithm(fr3_w_xyz)　(b) ORB-SLAM2 algorithm(fr3_w_static)　(c) ORB-SLAM2 algorithm(fr3_w_rpy)　(d) ORB-SLAM2 algorithm(fr3_w_half)

(e) 改进算法(fr3_w_xyz)　(f) 改进算法(fr3_w_static)　(g) 改进算法(fr3_w_rpy)　(h) 改进算法(fr3_w_half)

(e) Improved algorithm (fr3_w_xyz)　(f) Improved algorithm (fr3_w_static)　(g) Improved algorithm (fr3_w_rpy)　(h) Improved algorithm (fr3_w_half)

图 6　高度动态下绝对轨迹误差对比图

Fig. 6　Comparison of absolute trajectory errors in highly dynamic environments



(a) ORB-SLAM2算法(fr3_w_xyz)　(b) ORB-SLAM2算法(fr3_w_static)　(c) ORB-SLAM2算法(fr3_w_rpy)　(d) ORB-SLAM2算法(fr3_w_half)

(a) ORB-SLAM2 algorithm(fr3_w_xyz)　(b) ORB-SLAM2 algorithm(fr3_w_static)　(c) ORB-SLAM2 algorithm(fr3_w_rpy)　(d) ORB-SLAM2 algorithm(fr3_w_half)

(e) 改进算法(fr3_w_xyz)　(f) 改进算法(fr3_w_static)　(g) 改进算法(fr3_w_rpy)　(h) 改进算法(fr3_w_half)

(e) Improved algorithm (fr3_w_xyz)　(f) Improved algorithm (fr3_w_static)　(g) Improved algorithm (fr3_w_rpy)　(h) Improved algorithm (fr3_w_half)

图 7　高度动态下相对轨迹误差对比图

Fig. 7　Comparison of relative trajectory errors in highly dynamic environments

表 3　各种方法在 TUM 数据集下的绝对轨迹误差(ATE)度量结果

Table 3　Absolute trajectory error(ATE)metric results for various methods on the TUM dataset　m

序列	YOLO-SLAM	SG-SLAM	COEB-SLAM	文献[16]	本文
fr3_s_static	0.006 6	0.006 0	0.007 3	0.006 6	**0.005 8**
fr3_w_xyz	0.014 6	0.015 2	0.016 0	0.015 1	**0.013 9**
fr3_w_static	0.007 3	0.007 3	0.007 1	0.007 2	**0.006 7**
fr3_w_rpy	0.216 4	0.032 4	0.032 6	**0.026 4**	0.031 4
fr3_w_half	0.028 3	0.026 8	0.028 2	0.025 8	**0.023 7**

特征点被误删除,从而影响系统性能。

为了验证动态特征点去除算法的泛化能力,本文在环境更加复杂的 Bonn 数据集上进一步实验。并将 ATE 结果与 ORB-SLAM2、YOLO-SLAM 和 SG-SLAM 等算法进行对比,表 4 展示了 9 个序列下 ATE 的 RMSE 以及相对于 ORB-SLAM2 的提升率。与 ORB-SLAM2 系统相比,在高度动态的八个序列中,平均绝对轨迹误差减少了 97.12%,对于运动幅度较小的 moving_no_box2 序列,绝对轨迹误差也减少了 45.47%。此外,图 8 展示了不同方法对动态特征点的剔除情况,图 8(a)、(c)、(e)、(g)表明,当动态对象占据主体时大多数为动态特征点,若仅依靠掩码剔除可能导致特征点不足而影响系统性能,而本文方法不仅能准确剔除动态特征点,还保留了更多静态特征点,图 8(b)、(d)、(f)、(h)进一步展示了光流检测能够准确去除潜在动态对象(挪动的椅子、搬动的箱子)上的特征点,从而显著提高整体精度。

表 4 各种方法在 Bonn 数据集下的绝对轨迹误差(ATE)度量结果

Table 4 Absolute trajectory error(ATE)metric results for various methods on the Bonn dataset m

序列	ORB-SLAM2	YOLO-SLAM	SG-SLAM	文献[16]	本文	提升率/%
crowd1	1.032 9	0.033	0.023 4	0.019	**0.016 1**	98.44
crowd2	1.336 3	0.423	0.058 4	0.025	**0.023 3**	98.26
crowd3	0.797 6	0.069	0.031 9	0.032	**0.021 8**	97.27
moving_no_box	0.252 3	0.027	0.019 2	0.023	**0.016 0**	93.66
moving_no_box2	0.049 7	0.035	0.029 9	0.029	**0.027 1**	45.47
person_tracking	0.754 1	0.157	0.040 0	**0.038**	**0.037 7**	95.00
person_tracking2	0.846 3	0.037	0.037 6	0.038	**0.035 7**	95.78
synchronous	0.822 5	0.014	0.322 9	0.013	**0.008 6**	98.95
synchronous2	1.548 5	**0.007**	0.016 4	**0.007**	**0.006 7**	99.57

(a) 两人出现(ORB-SLAM2) (a) Couple's appearance (ORB-SLAM2)
(b) 单人挪动椅子(ORB-SLAM2) (b) Single scooting chair (ORB-SLAM2)
(c) 三人行走(ORB-SLAM2) (c) Three people walking (ORB-SLAM2)
(d) 单人搬动箱子(ORB-SLAM2) (d) Single person lifting box (ORB-SLAM2)
(e) 两人出现(本文) (e) Couple's appearance(Ours)
(f) 单人挪动椅子(本文) (f) Single scooting chair(Ours)
(g) 三人行走(本文) (g) Three people walking(Ours)
(h) 单人搬动箱子(本文) (h) Single person lifting box(Ours)

图 8 TUM 和 Bonn 数据集下动态特征点剔除情况对比图

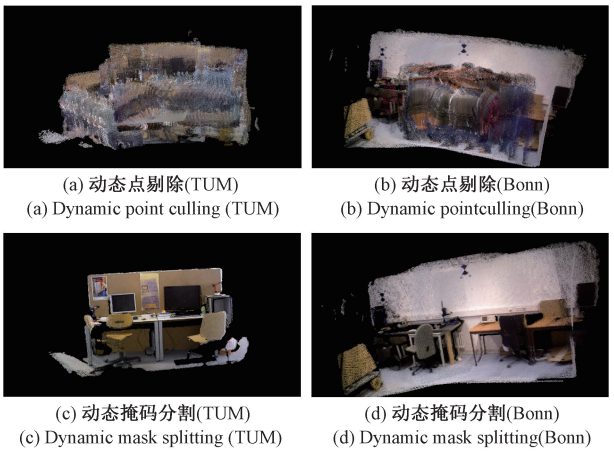
Fig. 8 Comparison of dynamic feature point rejection under TUM and Bonn dataset

与其他三种先进的 SLAM 系统相比,本文提出的算法在大多数动态序列上都表现出显著的优势,相对于 YOLO-SLAM 和 SG-SLAM,本文方法在最大误差上的改进分别达到了 0.399 7、0.314 3 m,显著降低了轨迹估计误差。在 person_tracking 和 synchronous2 序列中的轨迹精度也能够与文献[16]相媲美。综上实验结果均表明,本文算法能够显著削弱动态对象的影响,不仅能准确剔除动态特征点,还能够保留更多稳定的静态特征点,确保系统在复杂环境下依然具备良好的跟踪能力与定位精度。

4.4 密集点云建图评估

此外,对系统在构建稠密点云方面的性能进一步评估。图 9(a)、(b)展示了仅进行动态点剔除后生成的稠密点云地图,而图 9(c)、(d)则显示了结合运动掩码对运动物体进行分割后,再与原始关键帧图像融合后生成的稠密点

云地图。仅去除动态特征点虽然提升了系统的定位和跟踪性能，但由于动态点并不占据完整的动态像素，点云中仍然存在大量噪声。相比之下，使用运动掩码去除运动物体后生成的稠密点云不含运动噪声。这些结果表明，本文算法能够在动态环境中成功构建精确且纯静态的稠密点云。



(a) 动态点剔除(TUM)
(a) Dynamic point culling (TUM)
(b) 动态点剔除(Bonn)
(b) Dynamic pointculling(Bonn)
(c) 动态掩码分割(TUM)
(c) Dynamic mask splitting (TUM)
(d) 动态掩码分割(Bonn)
(d) Dynamic mask splitting(Bonn)

图 9　密集点云地图

Fig. 9　Map of dense point cloud

4.5　运行时间评估

系统的实际应用不得不考虑时间成本，因此本文对比了 3 种方法处理每帧图像所消耗时间，如表 5 所示。RGB-SLAM2 采用几何约束法仅需 24.53 ms，因此速度最快，原 YOLOv8l(Pytorch 格式)目标检测需 15.80 ms，耗时成本高的同时还需保证 C++与 python 间实时通信，而本文结合 TensorRT 加速后的目标检测处理仅需 5.86 ms，光流检测仅需 3.65 ms，结合动态特征过滤模块(包括光流检测)的跟踪线程仅需要 19.16 ms，仅比 ORB-SLAM2 的跟踪线程多 1.81 ms，因为减少的特征点跟踪平衡了增加模块的时间消耗，系统每帧的平均处理时间仅需 31.28 ms，可实现 32 Hz/s 的处理速度，因此认为本文算法满足实时运行条件。

表 5　运行时间对比

Table 5　Runtime comparison　　ms

方法	目标检测	光流检测	跟踪	总计
ORB-SLAM2	—	—	17.35	24.53
YOLOv8l	15.80	3.71	19.28	42.67
改进算法	5.86	3.65	19.16	31.28

5　结　　论

本文提出一种基于 ORB-SLAM2 框架改进的 SLAM 系统，利用 TensorRT 加速的 yolov8 提取语义信息，结合深度图像通过深度约束生成动态掩码，并提出一种基于动态概率的特征点四叉树均匀分配法，在剔除动态点的同时增加更多有用特征点，针对潜在动态对象上的特征点，利用光流跟踪进行检测与剔除。在 TUM 和 Bonn 公共数据集上的实验结果表明，该方法在高度动态环境下能够显著提升定位精度，在相对静止的场景中也能表现出较强的鲁棒性。与当前同类型算法相比，在保持实时运行的同时精度小幅提升，并且动态掩码与关键帧图像融合实现运动分割，能够准确消除密集点云中的运动漂移，构建完整密集点云地图。然而，该系统仍然存在一些局限性，未考虑到动态对象上特征点的真实运动属性，在预定义动态对象保持静止时系统性能会有所下降。

参考文献

[1] 杨雪梅，李帅永. 移动机器人视觉 SLAM 回环检测原理、现状及趋势[J]. 电子测量与仪器学报，2022，36(8)：1-12.
YANG X M，LI SH Y. Principle，current situation and trend of visual SLAM loop closure detection for mobile robot[J]. Journal of Electronic Measurement and Instrumentation，2022，36(8)：1-12.

[2] 韩超，陈敏，黄宇昊，等. 基于全局特征描述子的激光 SLAM 回环检测方法[J]. 上海交通大学学报，2022，56(10)：1379-1387.
HAN CH，CHEN M，HUANG Y H，et al. Loop closure detection method for lidar SLAM based on global feature descriptor[J]. Journal of Shanghai Jiao Tong University，2022，56(10)：1379-1387.

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

[4] ENGEL J，SCHOPS T，CREMERS D. LSD-SLAM：Large-scale direct monocular SLAM[C]. European Conference on Computer Vision，2014：834-849.

[5] ENGEL J，KOLTUN V，CREMERS D. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2017，40(3)：611-625.

[6] CHENG J Y，WANG C Q，MENG Q H. Robust visual localization in dynamic environments based on sparse motion removal[J]. IEEE Transactions on Automation Science and Engineering，2020，17(2)：658-669.

[7] DAI W CH，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.

[8] DU ZH J，HUANG SH SH，MU T J，et al. Accurate dynamic SLAM using CRF-based long-term consistency[J]. IEEE Transactions on Visualization

and Computer Graphics, 2022, 28(4): 1745-1757.

[9] BARATH D, MATAS J. Graph-cut RANSAC[C]. 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), 2018: 6733-6741.

[10] 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.

[11] YU C, LIU Z X, LIU X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]. 25th IEEE International Conference on Intelligent Robots and Systems(IROS), 2018: 1168-1174.

[12] 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.

[13] XIE W F, LIU X P, ZHENG M H. Moving object segmentation and detection for robust RGBD-SLAM in dynamic environments [J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-8.

[14] CONG P CH, LI J X, LIU J J, et al. SEG-SLAM: Dynamic indoor RGB-D visual SLAM integrating geometric and yolov5-based semantic information[J]. Sensors, 2024, 24(7): 2102.

[15] MIN F Y, WU Z B, LI D P, et al. COEB-SLAM: A robust VSLAM in dynamic environments combined object detection, epipolar geometry constraint, and blur filtering [J]. IEEE Sensors Journal, 2023, 23(21): 26279-26291.

[16] SUN H L, FAN Q W, ZHANG H Q, et al. A real-time visual SLAM based on semantic information and geometric information in dynamic environment[J]. Journal of Real-Time Image Processing, 2024, 21(5): 169.

[17] ZHU Y P, CHENG P, ZHUANG J, et al. Visual simultaneous localization and mapping optimization method based on object detection in dynamic scene[J]. Applied Sciences, 2024, 14(5): 1787.

[18] SAFALDIN M, ZAGHDEN N, MEJDOUB M. An improved YOLOv8 to detect moving objects[J]. IEEE Access, 2024, 12: 59782-59806.

[19] XIAO L H, WANG J G, QIU X S, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.

[20] WU W X, GUO L, GAO H L, et al. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint [J]. Neural Computing and Applications, 2022, 34(8): 6011-6026.

[21] 王爽, 刘云平, 张柄棋, 等. 基于全景分割与多视图几何的动态 SLAM 方法[J]. 电子测量技术, 2024, 47(24): 149-159.
WANG SH, LIU Y P, ZHANG B Q, et al. Dynamic SLAM approach based on panoptic segmentation and multi-view geometry [J]. Electronic Measurement Technology, 2024, 47(24): 149-159.

[22] CHENG SH H, SUN CH H, ZHANG S J, et al. SG-SLAM: A real-time RGB-D visual SLAM toward dynamic scenes with semantic and geometric information [J]. IEEE Transactions on Instrumentation and Measurement, 2023, 72: 1-12.

作者简介

尹显波,硕士研究生,主要研究方向为视觉 SLAM。
E-mail:yxb210@cumt.edu.cn

王中元(通信作者),副教授,硕士生导师,博士,主要研究方向为室内外融合定位、导航与 SLAM。
E-mail:wzy95002@163.com