基于 YOLOv8 同步动态检测与局部语义视觉 SLAM

杨海波,曹雏清
安徽工程大学 计算机与信息学院,安徽 芜湖 241000

摘 要:目的 视觉 SLAM 作为自动驾驶和移动机器人的核心技术之一,传统算法无法应对高度动态的环境,也缺乏地图的语义信息,解决动态物体对 SLAM 系统的影响是研究的主要目标,也是当前热点问题之一。**方法** 提出一个新的基于 YOLOv8 同步动态检测与局部语义分割的方法,来实现动态环境下的位姿估计与局部语义建图。首先,通过应用 YOLOv8 对输入图像进行同步动态检测和语义分割,使用目标检测结果的目标框对动态特征点进行剔除,再运用静态特征点进行姿态估计,然后在系统的语义建图线程中,对语义分割后的图像加入扩张掩模,最后使用点云库进行语义地图的构建,从而产生能够应用于实际场景的语义地图。**结果** 在 TUM 数据集中进行了比较试验,数据显示:这种方法相对于传统方法能提高 98.1%的位姿准确率,并且在实时性测试中,本文算法的速度也优于同类算法,而且可以在同一时间创建出局部语义地图。**结论** 基于 YOLOv8 同步动态检测与局部语义的方法来处理常规场景下的动态物体对 SLAM 系统的影响十分有效,且实时性高,但对于一些特殊场景如摄像机大幅旋转等,由于目标检测的失效而导致动态特征剔除失败,从而系统精度降低。

关键词:视觉 SLAM;语义分割;目标检测;语义地图
中图分类号:TP520 **文献标识码:**A doi:10.16055/j.issn.1672-058X.2025.0001.004

Synhronous Dynamic Detection and Local Semantic Visual SLAM Based on YOLOv8

YANG Haibo, CAO Chuqing
School of Computer and Information, Anhui Polytechnic University, Anhui Wuhu 241000, China

Abstract: Objective Visual SLAM, one of the core technologies for autonomous driving and mobile robots, is currently unable to cope with highly dynamic environments with traditional algorithms and lacks semantic information in maps. Addressing the impact of dynamic objects on SLAM systems is the main objective of this study, which is also one of the current hot topics. **Methods** A novel method based on YOLOv8 synchronous dynamic detection and local semantic segmentation was proposed to realize the position estimation and local semantic map building in dynamic environments. Firstly, synchronous dynamic detection and semantic segmentation were performed on input images using YOLOv8. Dynamic feature points were eliminated using the target frame of the target detection result, and then static feature points were applied for pose estimation. Then, an expansion mask was added to the semantically segmented image in the semantic mapping thread of the system. Finally, a point cloud library was used for the construction of semantic maps, thereby generating semantic maps applicable to real-world scenarios. **Results** Comparative tests were conducted in the TUM dataset, and the data showed that this method can improve the position accuracy by 98.1% compared with traditional methods. Moreover, the speed of the proposed algorithm was superior in real-time testing compared with similar algorithms, and it can create local semantic maps simultaneously. **Conclusion** The method based on YOLOv8 for synchronous dynamic

收稿日期:2023-09-15 **修回日期:**2023-11-14 **文章编号:**1672-058X(2025)01-0028-07
基金项目:国家自然科学基金面上项目(62073101);安徽省教育厅科学研究重点项目(KJ2020A0364).
作者简介:杨海波(1996—),男,安徽淮南人,硕士研究生, 从事视觉 SLAM 研究.
通讯作者:曹雏清(1982—),男, 江苏南京人,高级工程师,博士,从事智能机器人系统研究. Email: 465917277@qq.com.
引用格式:杨海波,曹雏清. 基于 YOLOv8 同步动态检测与局部语义视觉 SLAM[J]. 重庆工商大学学报(自然科学版),2025,42(1):28-34.
YANG Haibo, CAO Chuqing. Synhronous dynamic detection and local semantic visual SLAM based on YOLOv8[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2025, 42(1): 28-34.

detection and local semantics is highly effective in addressing the impact of dynamic objects on SLAM systems in typical scenes, with high real-time performance. However, in some special scenarios such as significant camera rotation, the failure of object detection leads to the failure of dynamic feature removal, resulting in reduced system accuracy.

Keywords: visual SLAM; semantic segmentation; target detection; semantic maps

1 引　言

视觉 SLAM(Simultaneous Localization and Mapping)是利用摄像机进行同步定位与建图的一种技术。当前阶段下,最常用的视觉 SLAM 技术可被归纳为两个类别:特征点法(Feature Point Method)以及直接法(Direct Approach)[1]。例如, MonoSLAM[2]、PTAM[3]、ORB-SLAM3[4]等都属于以特征点法为基础的算法。首先出现的 MonoSLAM 是一个实时单目相机位姿估计方案,然而其通过应用扩大的卡尔曼过滤器来实现最优,导致此项技术的运算过程繁复并且不能应对大场景或物体运动状态多变的环境条件。PTAM 利用非线性优化策略,但是因为它过大的数据计算并缺乏全局优化,难以达到高效运行的效果。最后提到的 ORB-SLAM3 则采用 ORB 特征描述子的匹配机制用于跟踪、建图、回环检测及其全局优化的过程当中。此算法同样也运用到了非线性的优化模式,从而保证了一个高的准确性和良好的地图绘制结果,同时还引入了一种 IMU 传感设备,以此增强系统的稳定性。基于特征点法的 SLAM 算法具有鲁棒性高、易实现等优点,但也存在建图不完整,难以应对动态物体等缺点,需要根据具体应用场景进行选择和优化。DVO-SLAM[5]、LSD-SLAM[6]都是基于直接法的视觉 SLAM 系统,DVO-SLAM 算法使用 RGB-D 相机进行稠密重建,通过最小化光度误差和深度误差提高位姿估计的精度。LSD-SLAM 算法能构建半稠密点云地图,其利用回环检测技术可生成大型场景的地图。此外,它还提出一种关键帧间漂移尺度公式的理论框架,以降低尺度漂移现象的影响率。然而,尽管这种基于直接法的 SLAM 算法拥有高实时性的优势,但是它的前提条件是假设同一场景中的像素灰度值在不同图像序列里保持恒定,这通常无法适应因光线变动所带来的影响,从而导致基于此技术的视觉 SLAM 系统的稳定性较弱,且在动态场景中定位效果差。

为应对以上问题,研究者们在基于深度学习的视觉 SLAM 方法上主要使用目标检测与语义分割两种。DynaSLAM[7]算法基于 ORB-SLAM2[8]算法,通过深度学习神经网络 Mask R-CNN[9]分割动态目标,采用多视图几何方法判断动态特征点,仅使用静态特征点进行跟踪与建图。虽然由 SU[10]等开发的 RTD-SLAM 系统加入了一个使用 YOLOv5S 的语义和光流特性识别组件来排除运动物体,从而提升其位姿估计的准确性和效率,然而该方法无法构建出有语义的地图信息。而 DS-SLAM[11]则采用 SegNet[12]作为一种图像分割技术,以获取具有特定含意的掩模,同时还运用极线几何原理去检测动态物体以此减少它们对于视觉里程计(Visual Odometer)所带来的影响。RDS-SLAM[13]是一种基于稀疏直接法和语义分割的实时视觉 SLAM 算法,结合稀疏直接法和语义分割技术,通过形态滤波器扩张掩膜,实现对动态物体的快速检测和排除,但实时性较差。文献[14]在 ORB-SLAM3 系统的基础上添加了语义分割模块,提高了动态场景下的定位精度,但系统实时性较差。

综上所述,面对动态场景,深度学习在视觉 SLAM 的现状是基于目标检测的方法,定位实时性高,但无法建立可被机器人识别的语义地图,而基于语义分割的方法可以建图但定位实时性较差。

本文提出一种基于 YOLOv8 的可同步进行目标检测与局部语义分割,实时性好的 SLAM 系统,既可以对潜在动态目标所在的图像区域进行目标检测、优化以减少动态目标对姿态估计的影响,同时,对图像进行局部语义分割,在点云图构建过程中剔除动态地图点,构建了不受动态对象影响的局部语义三维稠密地图。

2 动态环境下的位姿估计

本研究采用 ORB-SLAM3 作为动态环境视觉 SLAM 解决方案的基础,其结构由 5 部分组成,包括 YOLOv8 的目标检测及语义分割功能、视觉模块、局部地图模块、局部语义建图模块以及回环检测与多个地图合并模块。整个过程的数据流如下:首先,将 RGB-D 摄像机获取的图像信息传入视觉模块,执行 ORB 特征提取,与此同时,RGB 图像传输至 YOLOv8 模块,对运动目标进行目标检测和局部语义分割;然后,视觉模块会依据目标检测的结果来剔除动态特征点;最后,仅使用剩下的静态特征点进行相机的位姿推算。在局部建图模块里,提取满足条件的关键帧,并找到当前关键帧对应的局部语义分割图进行处理,然后进行局部 BA 与冗余关键帧剔除,最后将剔除后的关键帧与对应的局部语义分割图分别传入回环检测和多个地图融合模块,局部语义建图部分的关键帧会在回环检测和多个地图融合模块被用于优化并整合地图,而语义分割图像则会应用于局部语义建立地图的模块进行掩模扩张。当创建点云图的时候,利用先验信息来排除运动

物体的地图点。最终,通过位姿信息拼接点云,形成全局一致的局部语义地图(图 1)。

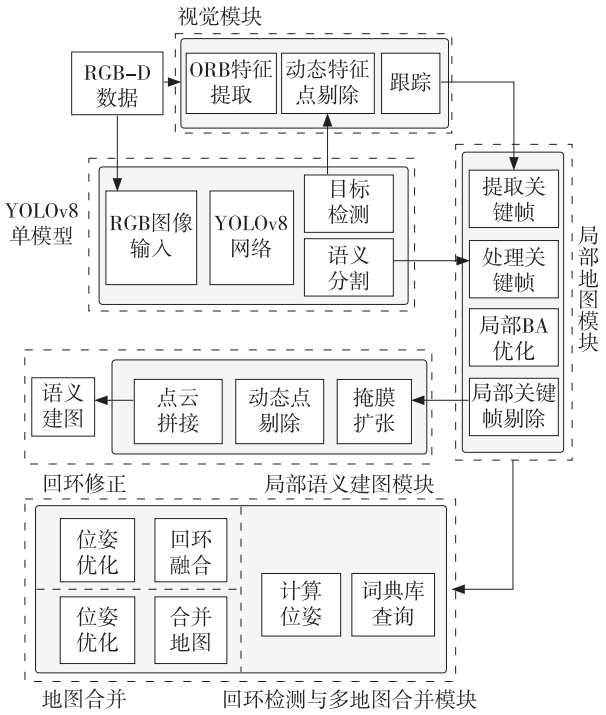

图 1 本文 SLAM 系统框架

Fig. 1 Framework of SLAM system in this paper

2.1 YOLOv8 同步目标检测与局部语义

基于传统神经网络的常规模型只能做一个目标检测功能或者语义分割功能,大多数目标检测算法在训练时都是基于人工预设锚框(anchor-based),不同场景泛化性差,而且对于需要定位与建图的动态场景,仅使用目标检测无法建立稠密语义地图,而传统语义分割算法在实时性上又比目标检测慢很多。为解决以上问题,本文采用不依赖锚框(anchor-free)的 YOLOv8,YOLOv8 集成了目标检测与语义分割功能,不同场景下泛化性好,在相同检测精度下,时间大大减少,并且在 SLAM 系统下可以自适应选择单独使用目标检测用于实时定位,或者是同步使用目标检测和局部语义分割进行定位与建图。

YOLOv8 的主要架构包括四大部分:输入层、主干网络、颈部和头部。其中,其主干网采用 CSPDarknet53,颈部则基于 PAN-FPN 的设计,至于头部,它包含了两类输出的模块,一类为目标框参数预测的解耦头设计,另一类则是用于语义分割掩模预测的部分。在 COCO 数据集中,目标检测 AP 为 53.9%,语义分割 AP 为 43.4%,实际应用速度可达 62 frame/s。本文采用 YOLOv8 的目标检测识别环境中的动态物体,室内动态环境主要是行人、动物。采用 YOLOv8 的语义分割去建立滤除动态点稠密语义地图,本研究利用 COCO VAL2017 数据集来训练 YOLOV8,目的是为了识别动态物体。该数据集包含了 80 种不同类型的物体,并从中提出动态物体的分类。重训练后的 YOLOv8 检测与分割结果如图 2 所示。

图 2 YOLOv8 检测与分割结果

Fig. 2 YOLOv8 detection and segmentation results

2.2 动态特征点剔除

如图 3 展示,本文只使用 YOLOV8 运行结果中的目标检测框来剔除动态特征点。假定每一个图像上所有提取出来的特征点的集合是 $A=\{p_1,p_2,\cdots,p_3\}$,每个特性点的坐标为 $p_i=(u_i,v_i)$。目标框可以被分类为两种定义:一种是动态目标框 $d_m=(l_m,t_m,r_m,b_m)$,另一种则是静态目标框 $d_s=(l_s,t_s,r_s,b_s)$。定义中,下标"m"表示的是运动物体,而下标"s"代表的是静态物体。"l"和"t"分别对应着目标框左侧和顶部水平坐标值,"r"和"b"则是目标框右侧横坐标值与底部边框的坐标值。通过设定目标框架,可以把图像上的关键点划分为两个集群,即位于行人群体内部的关键点构成一组,其余未包含于此或者属于其他实体群体的关键点构成了另一组。对于每一幅画面,都会依次检查所有的关键点并作出以下决策:若点 p_i 的条件符合 $\{(u,v)\mid l_m<u<r_m,t_m<v<b_m\}$,则 $p_i\in B$ 执行操作,移除此关键点;反之,则保持不动以用于位姿推测。

观察图 3(a)可知,一些特征点分布在人身上,这些特征点可能会干扰到机器人的定位精度,从而降低其准确性。而通过对比图 3(b)的显示结果,发现所有被移除的特征点均集中于静止对象上,这样能够显著减少运动目标对于机器人姿态估算产生的负面影响。

(a) 剔除前 (b) 剔除后

图 3 动态特征点剔除

Fig. 3 Dynamic feature point rejection

2.3　位姿计算

在估计机器人的位姿时,可以使用集合 C 中的静止特征点来推算。如果设定满足条件的特征点 $p_i=(u_i,v_i)$ 的空间点 $P_i(X_i,Y_i,Z_i)$,那么这些特征点与空间点之间的关系就是:

$$s_i\begin{bmatrix}u_i\\v_i\\1\end{bmatrix}=\boldsymbol{K}\exp(\hat{\xi_i})\begin{bmatrix}X_i\\Y_i\\Z_i\\1\end{bmatrix}+e \tag{1}$$

在这里,s 是尺度因子,$\boldsymbol{K}$ 是相机内参矩阵,ξ 是位姿的李代数,$\hat{}$是将李代数转化为矩阵的计算符,e 误差。

将式(1)改写为矩阵形式:

$$s_ip_i=\boldsymbol{K}\exp(\hat{\xi_i})P_i+e_i$$

将所有的误差加起来求和,构建出最小二乘问题。

$$\xi^*\arg\min\frac{1}{2}\sum_{i=1}^{n}\|p_i-\frac{1}{s_i}\boldsymbol{K}\exp(\hat{\xi_i})P_i\|_2^2 \tag{2}$$

因为 $p\in C$,所以式(2)中的 P_i 均为固定值,即 ξ^* 为所求的相机位姿。

3　动态环境下的局部语义建图

本文使用 TUM 数据集 Kinect-RGBD 相机采集的深度信息以及 YOLOV8 语义分割得到的语义信息,并在系统处理关键帧时,添加语义建图线程,使用 PCL 点云库来构建局部语义稠密地图。

3.1　形态滤波器扩展掩膜

由于在进行语义分割时,物体边界上的分割有时不可靠,有可能无法完全覆盖动态物体边界上的图像信息,如图 4 所示。

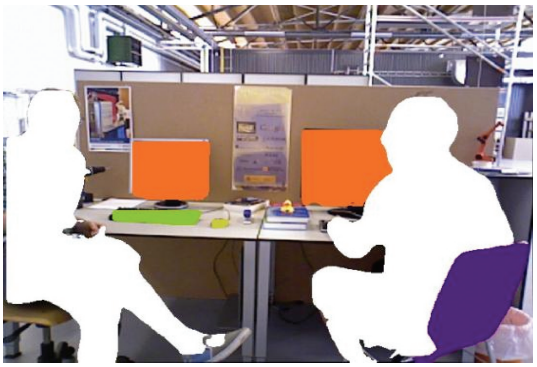

图 4　YOLOv8 语义分割结果

Fig. 4　YOLOv8 semantic segmentation result

因此,本文使用文献[13]的形态滤波器扩张掩模来包含动态物体的边缘。扩张掩模原理是检测一帧语义分割后图像里 RGB 值为(255,255,255)的像素点,然后将这部分像素点上下左右 50 个像素点的 RGB 值设置为(255,255,255)。为突出扩张部分的显示效果,图 5 将扩张掩模部分的像素 RGB 值设置为(0,255,0)。

图 5　添加形态滤波器的分割结果

Fig. 5　Segmentation results with morphological filters

在后续构建语义点云地图过程中,通过扩张掩模,建图线程会检测关键帧的每个像素点的 RGB 值是否为(255,255,255),若不是,则将此像素点投影至点云地图,以此使动态物体的点云可以完全剔除。

3.2　构建 3D 语义点云地图

使用 PCL 点云库构建点云地图,并将其中的三维地图点的数据格式设定为 pcl::point RGB(XYZ),而地图点被设定为 $p=(x,y,z,r,g,b)$,其中的 x、y、z 变量代表着地图点的空间位置,而变量 r、g、b 表示的是地图点 3 个颜色的具体数值。如果把图像上的每个像素 p 看作是与之相对应的三维空间点 P 的话,则两个点的关联方式可以描述成:

$$p=\frac{1}{2} \tag{3}$$

依据式(3),可以把像素点从原始视角投影至相机的 3D 空间,并将其转化为相机坐标系下的点云数据。然而,由于相机设备在运行过程中的坐标系统持续变动,需对每个单独的点云部分进行融合,形成在世界坐标下的统一地图,以便创建全局一致的点云地图。在动态的环境里,动态物体会不停地移动,导致地图变得杂乱无章,所以在判断是否为动态部分时,如果映射点符合条件$(r,g,b)=(255,255,255)$,则它们将会被排除出去。通过此方法,构建全局一致的 3D 语义点云地图,并应用于实际场景。

4　仿真实验与结果分析

为了确认本研究方法的实用性,利用 TUM 数据集进行相关试验。TUM 数据集是由 kinect 深度摄像机获取的各种室内环境图像序列,它能够提供相机的真实位姿值,从而协助研究人员评价 SLAM 系统的表现。本文使用该数据集的 4 种高度动态数据集(Walking_halfsphere, Walking_static, Walking_xyz, Walking_rpy)来进行实验。

4.1　位姿估计实验

研究采用绝对轨迹误差 ATE[15](Absolute Track Errors)为评估标准来衡量相机实际位置状态同 SLAM 系统预测位置状况间的差异程度。表 1 为本文算法与

ORB_SLAM3 在 3 个数据集上的测试结果对比,其中提升率的计算方法为

$$A_{\text{improvement}}=\frac{A_{\text{origin}}-A_{\text{our}}}{A_{\text{origin}}}\times 100\% \tag{4}$$

式(4)中,$A_{\text{improvement}}$、A_{origin}、A_{our} 依次代表在 ATE 下的提升率、原始值、本文算法得出的值。表格中 A_{RMSE}、A_{MEAN} 以及 A_{STD} 依次代表在 ATE 下的均方根误差、平均值以及标准差。

从表 1 中可得出结论:本文算法在动态数据集下的表现相较于 ORB-SLAM3 的位姿估计精度有了明显提高,其平均提升率达到 90.1%。而表 2 则展示了本文的方法与其他同类算法在 TUM 数据集下的实验结果对比:在尾缀为 xyz,与尾缀为 half 两个数据集中,本文算法表现更优,其根本原因在于在 COCO 数据集训练过程中,很少使用旋转后的图片对模型进行训练,导致 YOLOv8 在摄像头旋转比较大的场景里检测效果较差。如图 6 所示,ORB_SLAM3 和本研究提出的方法分别对 walking_xyz 序列进行了跟踪。

表 1 绝对轨迹误差对比

Table 1 Comparison of absolute trajectory error

数据集	ORB_SLAM3			本文算法			提升率/%		
	A_{RMSE}/m	A_{MEAN}/m	A_{STD}/m	A_{RMSE}/m	A_{MEAN}/m	A_{STD}/m	A_{RMSE}	A_{MEAN}	A_{STD}
Walking_xyz	0.677	0.586	0.340	0.016	0.014	0.007	97.6	97.6	97.9
Walking_rpy	0.767	0.719	0.283	0.219	0.194	0.102	71.4	73.0	63.9
Walking_half	0.395	0.359	0.346	0.026	0.022	0.013	93.4	93.8	96.2
Walking_static	0.354	0.148	0.165	0.007	0.006	0.003	98.0	95.9	98.1

表 2 同类算法对比

Table 2 Comparison of similar algorithms

数据集	DS-SLAM	DynaSLAM	RDS-SLAM	本文算法
	A_{RMSE}/m	A_{RMSE}/m	A_{RMSE}/m	A_{RMSE}/m
Walking_xyz	0.024 7	0.016 4	0.056 5	0.016 1
Walking_rpy	0.444 2	0.035 4	0.012 8	0.219
Walking_half	0.030 3	0.029 6	0.029 1	0.026 1
Walking_static	0.008 1	0.006 8	0.003 9	0.007 2

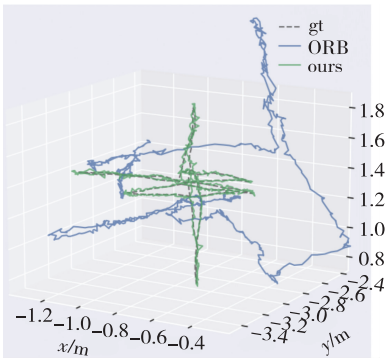

图 6 轨迹真实值与计算值曲线

Fig. 6 Curves of real and calculated values of trajectory

同样地,图 6 展示了在 X、Y、Z 三维坐标下的 ORB_SLAM3 和本研究方法与其真实轨迹之间的误差情况。可以看到:灰色虚线代表的是真实摄像机的位移真值,蓝色线条描绘出 ORB_SLAM3 的位移,绿色线条则显示出了本文方法的轨迹。结果表明本文方法能够非常精确地追踪真实轨迹。

最后,图 7 和图 8 分别为 ORB-SLAM3 和本研究方法的计算误差统计图。通过观察图 7,可以发现 ORB-SLAM3 的计算误差大约落在 0.6 m 范围内,而在图 8 中,可以看到本研究方法的计算误差约在 0.01~0.02 m 区间内。通过对比实验结果,可得出结论:本文算法在动态环境中定位准确率更高。

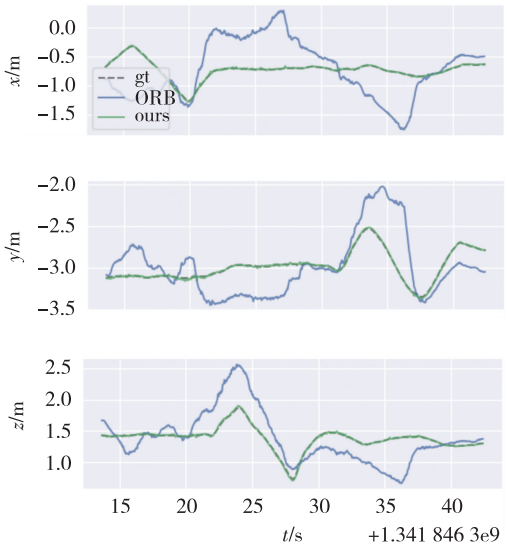

图 7 轨迹真实值与计算值分别在 *XYZ* 轴的误差

Fig. 7 Error between the real and calculated values of the trajectory in the *XYZ* axis, respectively

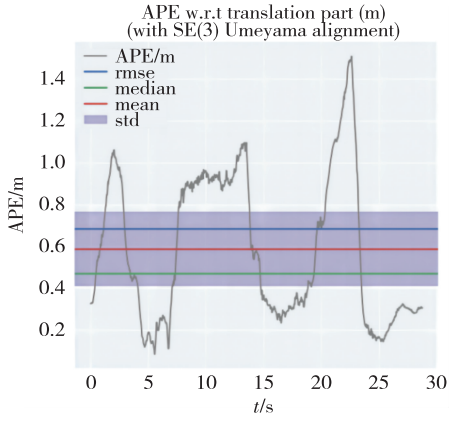

图 8　ORB 算法绝对估计误差分布

Fig. 8　Distribution of absolute estimation errors of ORB algorithm

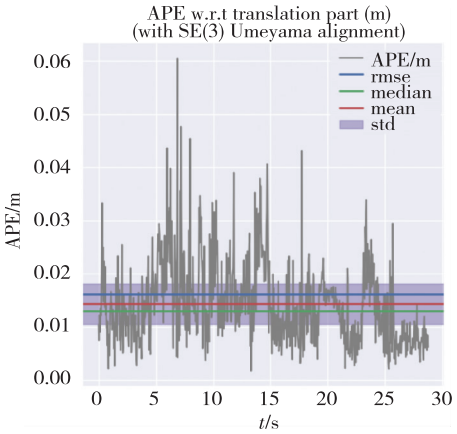

图 9　本文算法绝对估计误差分布

Fig. 9　Distribution of absolute estimation errors of the proposed algorithm

4.2　局部语义建图实验

本研究选取 Walking_xyz 序列来实现语义构图的图像数据集。该系列包含两个人在摄像机面前围绕一间办公桌行走的情况,是个高动态的图像序列。如果不处理这两个人的影响,那么地图构建可以参考图 10 的效果,可见大量“残像”被映射到地图上,使得机器人误以为前面存在障碍而不能通行。

图 10　滤除前建图效果

Fig. 10　Effect of building a map before filtering

如果直接在语义建图时应用语义分割后的 Mask 图像,则单帧图像里动态物体未被语义分割覆盖的部分会直接投影成三维点云,进而影响到语义建图导致“鬼影”,如图 11 所示。

然而,通过本文扩张掩膜的方法,利用语义分割去除行人和使用形态滤波器消除动态目标边界上的点云信息,可见滤除后的效果图 12,从而为椅子、计算机及水杯等特定物品提供了语义标签,这对机器人的路线规划与导航具有积极意义,同时也能应对完成诸如“从桌面上拿水杯”这类高级操作的场景。

图 11　未使用形态滤波器的建图效果

Fig. 11　Effect of building a map without morphological filters

图 12　本文方法建图效果

Fig. 12　Effect of building a map by the method proposed in this article

4.3　实时性评估

表 3 给出了 TUM 数据集测试下本文方法和同类方法每帧模型的推理和跟踪时间,本文方法采用两组模型,并在 RTX 2080Ti 显卡下进行测试。

本文方法第一组模型为 YOLO-Detect,只输出动态目标检测,第二组模型为 YOLO-Detect&Segment,既输出动态目标检测又输出语义分割图像。结果,第一组改进后的系统每帧跟踪时间最短,仅为 43.1 ms,第二组系统每帧跟踪时间为 49.4 ms,并且两组模型的推理时间也是同类系统中最优的。

表3 性能评估

Table 3 Performance evaluation

方法	GPU	网络模型	模型推理时间/ms	跟踪每帧的时间/ms
ORB-SLAM3	GeForce RTX2080Ti	—	—	21.3
DS-SLAM	P4000	SegNet	37.57	59.40
DynaSLAM	Tesla M40 GPU	Mask R-CNN	195.00	>300
RDS-SLAM	GeForce RTX 2080Ti	Mask R-CNN	200.00	50~65
本文方法	GeForce RTX2080Ti	YOLOv8-Detect	23.12	43.1
本文方法	GeForce RTX2080Ti	YOLOv8-Detect&Segment	32.42	49.4

试验结果表明:在满足实时性的要求下,本文采用YOLOv8同步目标检测与语义分割算法的跟踪延迟时间相较于同类使用语义分割的系统有明显提高,并且能够完成语义建图工作。

5 结 论

为克服传统视觉SLAM系统不能应对高度变化的环境和构建不了局部语义图景的难题,提出利用YOLOv8同步目标检测及局部语义分割的SLAM策略,并提供了YOLO-Detect、YOLO-Detect&Segment两种实时性方案,这两种方案在常见的动态数据集上表现出色,并可以根据实际应用需求选择合适的方案。

然而当面对有摄像机转动的动态环境时,系统稳定性有所下降。为使得系统更加鲁棒,未来将通过增加大量经过旋转处理的数据图像到神经网络的训练中以提高该网络模型在不同动态情境中的准确率,同时添加极线几何约束来进一步排除动态物体的影响。在实时性方面,未来将继续轻量化YOLOv8网络来提高系统的速度。

参考文献(References):

[1] GAO X. Fourteen lectures on visual SLAM: From theory to practice[M]. Beijing: Electronic Industry Press, 2019: 152-153.

[2] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: real-time single camera SLAM [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007, 29(6): 1052-1067.

[3] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]//Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Piscataway: IEEE Press, 2007: 225-234.

[4] CAMPOS C, ELVIRA R, RODRIGUEZ 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.

[5] KERL C, STURM J, CREMERS D. Dense visual SLAM for RGB-D cameras [C]//Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2013: 2100-2106.

[6] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: large-scale direct monocular SLAM[C]//European Conference on Computer Vision. Cham: Springer, 2014: 834-849.

[7] 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.

[8] 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.

[9] HE K, GKIOXARI G, DOLLAR P, et al. Mask R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision. Piscataway: IEEE Press, 2017: 2961-2969.

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

[11] YU C, LIU Z, LIU X J, et al. DS-SLAM: a semantic visual SLAM towards dynamic environments[C]//Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 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] LIU Y, MIURA J. RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods [J]. IEEE Access, 2021, 9: 23772-23785.

[14] ZHANG H, HUANG R, YUAN L. Robust indoor visual-inertial SLAM with pedestrian detection[C]//Proceedings of the IEEE International Conference on Robotics and Biomimetics. Piscataway: IEEE Press, 2021: 802-807.

[15] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway: IEEE Press, 2012: 573-580.

责任编辑:李翠薇