基于深度学习的实时同步定位与建图算法研究

石征锦[1] 王晟霖[1] 武 晨[1] 卜春光[2,3] 范晓亮[2,3]
SHI Zhengjin WANG Shenglin WU Chen BU Chunguang FAN Xiaoliang

摘 要 传统的视觉同步定位与建图（simultaneous localization and mapping，SLAM）算法大多数建立在假设场景是静态的基础之上，这种假设限制了视觉 SLAM 在现实场景的应用。针对传统 SLAM 算法在动态环境下定位精度低、鲁棒性差的问题，提出了一种实时动态视觉 SLAM 算法。首先所提出的算法以 ORB-SLAM3 为基础，新增了一个语义线程，该线程与其他线程并行运行，可以避免语义线程运行较慢而影响跟踪线程的运行。然后使用移动概率更新和传播语义信息，将其保存在地图中，并且使用数据关联算法从跟踪中去除动态点。最后使用公共 TUM 数据集来评估，证明了所提出的算法在动态环境下的鲁棒性和实时性优于现有的算法。

关键词 视觉同步定位与建图；动态环境；语义分割；实时性

doi：10.3969/j.issn.1672-9528.2024.01.044

0 引言

同步定位与建图（simultaneous localizat-ion and mapping，SLAM）[1] 是一项利用传感器（如单目相机、双目相机和 RGB-D 相机）在未知环境中估计自身位姿并在定位的基础上重建地图的技术，其广泛应用于增强现实（AR）、移动机器人和自动驾驶等领域。视觉 SLAM（vSLAM）[2] 使用相机作为主要传感器，在相同条件下可以获取更丰富的信息。然而传统的视觉 SLAM，例如 ORB-SLAM2[3]，对于刚性场景的假设限制了视觉 SLAM 在现实场景中的应用。现实中的动态物体如人群等，会在 SLAM 过程中造成错误的特征匹配，从而影响定位精度。动态环境是视觉 SLAM 在现实应用中的一项重大挑战。通常而言，视觉 SLAM 有两项重要的指标：跟踪的鲁棒性和实时性。如何实时地在未知场景下检测动态对象并且避免其对 SLAM 过程的不良影响是实现视觉 SLAM 在现实环境中应用亟待解决的问题。

目前，动态视觉 SLAM 的解决方案可以分为两类：基于几何 [4] 方法和基于深度学习 [5] 的方法。Dai 等人 [6] 通过地图点之间的相关性来区分特征点的动静状态；Li 等人 [7] 通过对关键帧边缘点使用静态加权的方法来减少动态物体的影响；Zou 等人 [8] 通过将地图特征投影到当前帧来计算重投影误差，通过计算结果判断特征点是否为动态。以上方式均采用几何方式判断特征点是否为动态，这些方式不能检测潜在的运动物体，例如坐着的人或停靠的车等。这类物体的特征对于跟踪同样是不可靠的，需要将其从跟踪线程中删除。

随着深度学习技术的发展，部分研究者尝试用深度学习解决动态 SLAM 问题。基于深度学习的方法可以使用目标检测或语义分割方法来获取潜在动态物体的边界框或像素级别的掩膜。潜在的移动物体可以被检测到并移除，然后利用静态特征点建图。DS-SLAM[9] 将语义分割（SegNet[10]）与运动一致性方式相结合的方式来减少动态对象的影响。DS-SLAM 假设人身上的特征点最有可能动态的，如果一个人被确定是静止的，那么此人身上的特征点也可用作位姿估计。

Bescos 等人 [11] 提出了 Dyna-SLAM，将语义分割和多视图集合结合，利用 MASK R-CNN[12] 对潜在动态目标（如坐着的人、停着的车）进行分割，同时使用静态地图点绘制被动态物体遮挡的部分，在动态场景中具有鲁棒性，但缺点是耗时严重、实时性差。Zhong 等人 [13] 提出了 Detect-SLAM，该方法只在关键帧的彩色图像使用 SSD 网络 [14] 检测动态物体，并通过运动概率表示特征点的动态概率。DS-SLAM、Dyna-SLAM 和 Detect-SLAM 是近年来视觉 SLAM 在动态环境下有效的解决方法。然而，以上三种方法需要先检测或分割物体，然后在跟踪中去除动态点，跟踪线程必须先等待语义分割结果。因此，这些方式的速度受到语义分割速度的限制，例如 Dyna-SLAM 使用的 MASK R-CNN 分割一张图像大约需要 200 ms，这限制了整个系统的实时性。

1. 沈阳理工大学自动化与电气工程学院 辽宁沈阳 110159
2. 中国科学院沈阳自动化研究所机器人学国家重点实验室 辽宁沈阳 110016
3. 中国科学院机器人与智能制造创新研究院 辽宁沈阳 110016

[基金项目] 国家重点研发计划（2022YFB4703605）

考虑到视觉 SLAM 对实时性的要求较高，提出了一种基于语义分割的实时视觉 SLAM 算法，本文的主要贡献如下。

（1）对 ORB-SLAM3[15] 进行改进，添加语义分割线程，该线程与跟踪线程并行，不会阻塞跟踪线程进行，在保证实时性的同时，有效减少动态物体对 SLAM 位姿估计的影响。

（2）提出一种分割方式，只在关键帧上进行分割，提高系统的运行效率。

（3）以移动概率的方式表示特征点是否为动态，通过语义信息更新移动概率。

（4）使用 TUM 数据集对提出的算法进行测试，并与类似的算法进行比较，证明了算法的实时性和鲁棒性。

1 系统框架

系统在 ORB-SLAM3 的框架上新增了一个语义线程，该语义线程与跟踪线程并行，并且在跟踪线程中做出一些改动。如图 1 所示，首先每一帧通过跟踪线程，在跟踪最新一帧后，通过局部地图跟踪进行优化。其次，进行关键帧的选择，选择关键帧用于局部建图以及语义线程。再次，语义线程对关键帧进行分割，并将语义信息保存在地图集中。然后，语义标签生成先验动态图片的掩膜。最后语义信息用以更新关键帧中的地图点的移动概率。系统的其余部分与 ORB-SLAM3 相同。

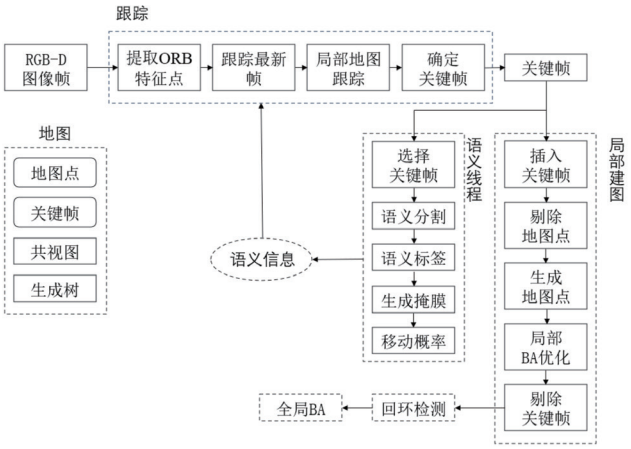


图 1 算法流程图

1.1 语义线程

语义线程负责生成于语义信息并保存到地图中。语义线程的运行过程为：首先，从关键帧列表中选择语义关键帧，关键帧由 ORB-SLAM3 选择；其次，从语义模型中请求语义标签并返回；然后，利用语义标签计算先验动态对象的掩膜；最后，更新存储在地图集中的地图点的移动概率。

1.2 语义关键帧选择

语义关键帧的选择是为语义线程请求语义标签提供语义关键帧。语义线程不会影响跟踪线程的运行速度，但跟踪线程的当前帧与最新的语义信息在时间上存在一定的延迟，这是由于语义分割模型的运行速度耗时更长导致的。如果按照顺序分割每个关键帧，当前帧可能无法获得最新的语义信息。为了评估这种延迟，将其称为语义延迟，即当前帧与包含最新语义信息的帧的距离，计算公式为：

$$d(F_t) = FrameID(F_t) - FrameID(KF_t) \quad (1)$$

式中，F_t 表示当前帧，$d(F_t)$ 表示当前帧的语义延迟，KF_t 表示包含最新语义信息的帧，即语义关键帧。

如图 2 所示，显示了不同关键帧选择方法的语义延迟，在此先假设语义分割耗时 t=10。图 2（a）表示按顺序选择语义关键帧，在使用比较耗时的语义分割方法时，语义延迟会逐渐增加。

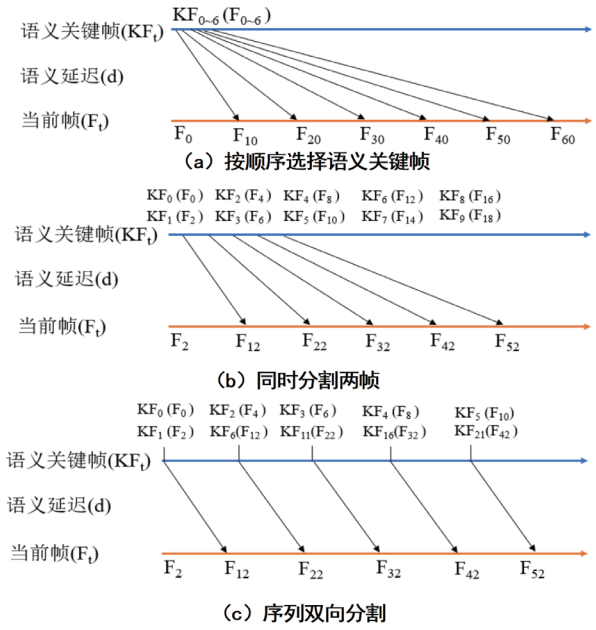


图 2 不同的关键帧选择策略产生的语义延迟

如图 2（a），首先按顺序选择 F_0 为语义关键帧。当 t=10 时，语义模型完成了对 KF_0 的分割，根据式（1）计算出当前帧 F_{10} 的语义延迟 $d(F_{10})$=10。然后，语义模型开始分割 $KF_1(F_1)$，当 t=20 时，语义延迟 d=18，同样地，在 t=60 时，语义延迟 d=54。也就是说，当前帧距离最新的语义信息相差 54 帧，系统无法获得最新的语义信息。

为了减少语义延迟，尝试了一种减少语义延迟的策略，即同时分割两帧图像。如图 2（b）所示，同时对 $KF_0(F_0)$ 和 $KF_1(F_2)$ 进行分割，则 t=12 时，语义延迟 d=10，当 t=22 时，语义延迟 d=16。同样地，t=52 时，d=34，语义延迟仍然呈线性增长。

为了避免语义延迟线性增长的问题，选择了双向分割的方法，即不按照顺序选择语义关键帧，而使用序列前后两帧进行分割。如图 2（c）所示，在第一个序列（从 F_0 到

F_{12}），选择 F_0 和 F_2 作为语义关键帧，t=12 时分割结束，系统请求语义标签，语义延迟 d=10，选择序列前后两帧作为新的语义关键帧，即 F_4 与 F_{12}，t=22 时分割结束，语义延迟 d=10，选择新的关键帧 F_6 与 F_{22}，继续进入下一轮。后续语义延迟同样为 10。

如图 3 所示为上述三种分割方式的语义延迟，在第三种方式的情况下，语义延迟成为一个常量。利用此方式，跟踪线程可以尽可能地使用最新的语义信息。然而在实际情况中，语义延迟并不一定是 10，具体的延迟取决于语义分割模型的速度，关键帧的选择也受到语义分割速度的影响。

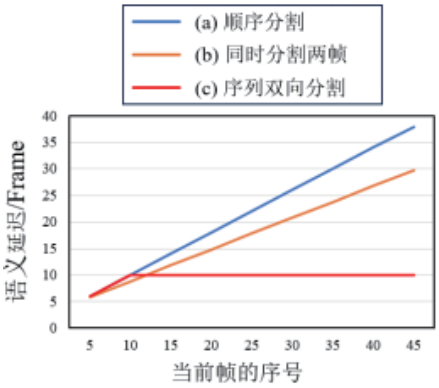


图 3 三种分割方式的语义延迟

1.3 语义分割

选择 SegNet 作为分割模型。SegNet 是 2016 年由 Cambridge 提出的开源图像语义分割深度学习网络，该网络基于 caffe 框架，能够提供像素级的语义分割结果。使用了 PASCAL VOC[16] 2012 数据集对分割模型训练，模型提供了 20 个分类。

1.4 语义分割掩膜

首先将所有实例分割结果生成的二值图合并成一个掩膜（Mask）图像，以生成人的掩膜图像。然后利用掩膜计算特征点的移动概率。如图 4 所示为实例分割以及生成的掩膜。

(a) 被分割的图像　　(b) 分割结果

图 4 SegNet 分割生成掩膜

1.5 移动概率

为了避免跟踪线程等待语义信息，语义线程被设计到一个单独的线程，与跟踪线程并行。使用移动概率将语义信息传递到跟踪线程，移动概率被用于识别移动物体的特征点，动态的特征点会被移除。

将移动概率定义为每个特征点 i 在 t 时刻的移动概率 $p(m_{i,t})$。如果特征点的移动概率接近于 1，那么该特征点就有可能是动态的，移动概率越接近于 0 就越可能是静态点。每个特征点被分成动态（m=D, *dynamic*）和静态（m=S, *Static*）两种状态。初始状态的移动概率被设置为 m_0=0.5。

利用设置的阈值 θ_d 和 θ_s 来判断特征点是否为动态，如公式（2）所示，根据实验的结果阈值分别被设置为 θ_d=0.6，θ_s=0.4。

$$状态(m_{i,t})=\begin{cases}静态 & p(m)<\theta_s\\ 动态 & p(m)>\theta_d\\ 未知 & 其他\end{cases} \quad (2)$$

2 实验结果

使用了室内 TUM[17] 数据集评估跟踪的精度。TUM RGBD 数据集是评估 RGBD SLAM 的常用数据集，包含了 39 个序列，每个序列包含了彩色图像，深度图像以及真实的相机位姿。选择的数据集序列为 walking_xyz、walking_rpy、walkiing_halfs-phere 和 walking_static。在这些序列中，两人走过一间办公室，相机分别按 xyz 轴移动、沿 rpy 轴运动、在一个直径为 1 m 的半球上移动以及基本固定不动。这是为了评估 SLAM 系统在动态环境下的鲁棒性。

评估 SLAM 系统定位精度使用了两个主要的测量指标：绝对轨迹误差（ATE）和相对位姿误差（RPE）。这两个数值非常适合用来测试视觉 SLAM 的性能。比较了 ATE 和 RPE 的均方根误差（RMSE）和标注差（S.D.）。

测试的结果与目前主流的开源视觉 SLAM 算法进行对比，对于其他用于对比的算法，尽可能地使用原论文中的实验结果。

2.1 精度对比实验

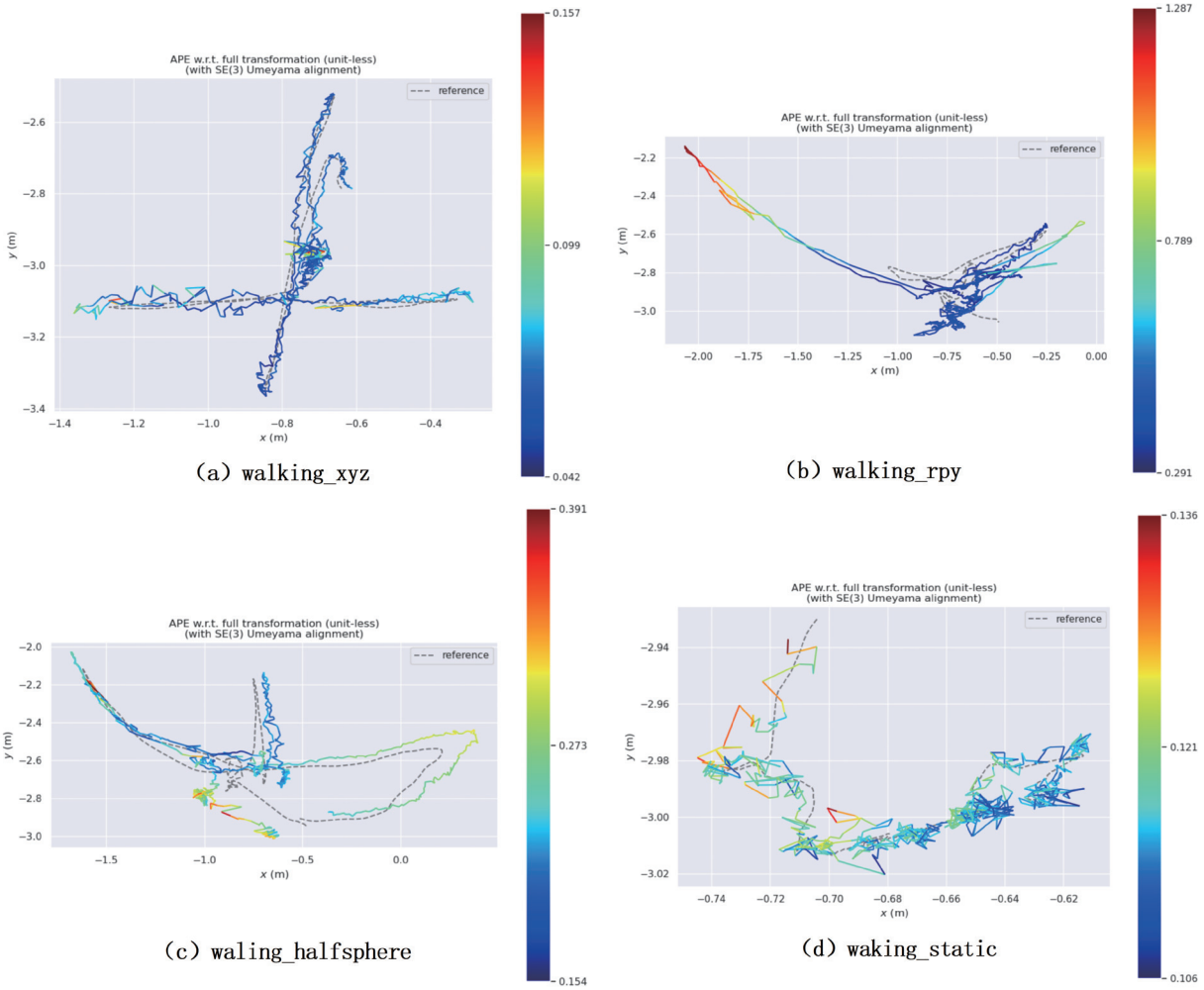
表 1 和表 2 显示了同类的开源 SLAM 系统绝对轨迹误差和相对位姿误差，包括了 ORB-SLAM3（不含语义信息）、Dyna-SLAM 和 DS-SLAM。从实验结果可以看出，在跟踪精度上，本文的方法明显优于没有使用语义分割的 ORB-SLAM3，与 DS-SLAM 比较接近。图 5 展示了提出的方法在各个数据集序列下估计的轨迹与真实轨迹的差别，其中虚线为相机真实的轨迹，实线为估计的轨迹，实线的颜色表示估计的轨迹与真实轨迹误差的大小，越接近红色表示误差越大，越接近蓝色表示误差越小。

表 1 与其他开源 SLAM 系统的绝对轨迹误差（ATE）的对比

数据集序列	ORB-SLAM3		Dyna-SLAM		DS-SLAM		文中算法	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
walking_xyz	0.917 8	0.485 9	0.016 4	0.008 6	0.024 7	0.016 1	0.069 8	0.015 2
walking_rpy	1.019 7	0.512 2	0.035 4	0.019 0	0.444 2	0.235 0	0.471 1	0.172 7
walking_half	0.657 2	0.312 4	0.029 6	0.015 7	0.030 3	0.015 9	0.255 0	0.038 5
walking_static	0.361 4	0.152 2	0.006 8	0.003 2	0.006 5	0.003 3	0.127 5	0.003 1

表 2 与其他开源 SLAM 系统的相对轨迹误差（RPE）的对比

数据集序列	ORB-SLAM3		Dyna-SLAM		DS-SLAM		文中算法	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
walking_xyz	0.425 8	0.306 3	0.021 7	0.011 9	0.033 3	0.022 9	0.021 8	0.012 4
walking_rpy	0.436 8	0.436 8	0.044 8	0.026 2	0.150 3	0.116 8	0.048 5	0.037 9
walking_half	0.326 2	0.326 2	0.028 4	0.014 9	0.029 7	0.015 2	0.028 0	0.017 1
walking_static	0.326 2	0.326 2	0.012 6	0.006 7	0.007 8	0.003 8	0.010 5	0.005 2



(a) walking_xyz　(b) walking_rpy

(c) waling_halfsphere　(d) waking_static

图 5 在不同数据集序列下估计的轨迹与真实轨迹

2.2 性能对比实验

运行的速度是视觉 SLAM 性能另一个重要指标。如表 3 所示为提出的方法与同类 SLAM 系统运行速度的对比，从结果可以看出，提出的方法运行速度明显优于 Dyna-SLAM 与 DM-SLAM[18]，与 DS-SLAM 的运行速度相当。Dyna-SLAM 使用了 Mask R-CNN 和多视图几何的方法取得了最高的精度，然而这种方法需要大量的时间，因此 Dyna-SLAM 的运行速度较慢。综合实验结果，本文提出的方法在使用了更少的计算资源时实现了类似的性能。

表 3 在 TUM 数据集上与其他系统运行速度对比

方法	GPU	分割方法	分割每帧的时间 /ms	跟踪每帧的时间 /ms
ORB-SLAM3	—	—	—	22 ～ 30
Dyna-SLAM	Tesla M40	Mask R-CNN	195.00	>300
DS-SLAM	P4000	SegNet	37.57	59.40
DM-SLAM	RTX 1080 Ti	Mask R-CNN	201.02	>201
本文算法	P1000	SegNet	30.00	50 ～ 70

3 结论

本文基于现有的动态视觉 SLAM 在动态环境下的实时性差的问题，提出了一种使用独立语义线程的 SLAM 算法。在 TUM 数据集上的实验结果表明，文中的方法提高了运行的速度，在保证鲁棒性的情况下，获得了更好的实时性。

参考文献：

[1] 吴建清 , 宋修广 . 同步定位与建图发展综述 [J]. 山东大学学报（工学版）,2021,52(5):16-31.

[2] 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.

[3] 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.

[4] 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.

[5] 李小倩 , 何伟 , 朱世强 , 等 . 基于环境语义信息的同步定位与地图构建方法综述 [J]. 工程科学学报 ,2021,43(6):754-767.

[6] 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.

[7] LI SL, LEE D. RGB-D SLAM in dynamic environments using static point weighting[J]. IEEE robotics and automation letters. 2017,2(4): 2263-2270.

[8] ZOU D, TAN P. CoSLAM: collaborative visual SLAM in dynamic environments[J]. IEEE transactions on pattern analysis and machine intelligence,2013,35(2):354-366.

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

[10] 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.

[11] BESCOS B, FACIL J M, CIVERA J, et al. Dyna-SLAM: tracking, mapping, and inpainting in dynamic scenes [J]. IEEE robotics and automation letters. 2018, 3(4): 4076-4083.

[12] EH K,GKIOXARI G,DOLLAR P,et al. Mask R-CNN[J]. IEEE transctions on pattern anaysis and machine intelligence, 2017,42(2): 2980–2988.

[13] ZHONG F W, WANG S, ZHANG Z Q, et al. Detect-SLAM: making object detection and SLAM mutually beneficial[C]// IEEE Winter Conference on Applications of Computer Vision(WACV).Piscataway,IEEE, 2018:1001-1010.

[14] LIU W, D Anguelov, D Erhan, et al. SSD: single shot multibox detector[C]//In European conference on computer vision. Piscataway,IEEE,2016:21–37.

[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 transactio-ns on Robotics, 2021,37(6):1874-1890.

[16] EVERINGHAM M,VAN G L.WILLIAMS C, et al. The pascal visual object classes (VOC) challenge international journal of computer vision[J].International journal of computer vision, 2010,88(2):303-338.

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

[18] CHENG J, WANG Z, ZHOU H,et al.DM-SLAM: A feature-based SLAM system for rigid dynamic scenes[J]. ISPRS international journal of geo-information, 2020,9(4):1-18.

【作者简介】

石征锦 （1963—），男，教授，研究方向：先进控制理论及应用。

王晟霖（1996—），男，硕士研究生，研究方向：复杂系统综合自动化技术。

武晨（1998—），女，硕士研究生，研究方向：检测技术与自动化装置。

卜春光（1971—），通信作者，男，副研究员，研究方向：自主移动与操作机器人。

（收稿日期：2023-10-15）