计算机应用研究
Application Research of Computers
ISSN 1001-3695,CN 51-1196/TP

《计算机应用研究》网络首发论文

题目：基于语义分割动态特征点剔除的 SLAM 算法
作者：张恒，徐长春，刘艳丽，廖志芳
DOI：10.19734/j.issn.1001-3695.2021.09.0402
收稿日期：2021-09-26
网络首发日期：2021-12-14
引用格式：张恒，徐长春，刘艳丽，廖志芳．基于语义分割动态特征点剔除的 SLAM 算法[J/OL]．计算机应用研究.
https://doi.org/10.19734/j.issn.1001-3695.2021.09.0402

网络首发时间：2021-12-14 14:44:07
网络首发地址：https://kns.cnki.net/kcms/detail/51.1196.TP.20211213.1406.009.html

基于语义分割动态特征点剔除的 SLAM 算法 *

张 恒[1,2]，徐长春[1]，刘艳丽[1,2†]，廖志芳[3]

(1. 华东交通大学 信息工程学院，南昌 330013; 2. 上海电机学院 电子信息学院，上海 201306; 3. 中南大学 计算机学院，长沙 410083)

摘 要：针对动态物体容易干扰 SLAM 建图准确性的问题，提出了一种新的动态环境下的 RGB-D SLAM 框架，将深度学习中的神经网络与运动信息相结合。首先，算法使用 Mask R-CNN 网络检测可能生成动态对象掩模的潜在运动对象。其次，算法将光流方法和 Mask R-CNN 相结合进行全动态特征点的剔除。最后在 TUM RGB-D 数据集下的实验结果表明，该方法可以提高 SLAM 系统在动态环境下的位姿估计精度，比现有的 ORB-SLAM2 的表现效果更好。

关键词：同步定位与建图；特征点；动态环境；语义分割
中图分类号：TP242.6 doi: 10.19734/j.issn.1001-3695.2021.09.0402

Slam algorithm based on semantic segmentation and dynamic feature point elimination

Zhang Heng[1,2], Xu Changchun[1], Liu Yanli[1,2†], Liao Zhifang[3]

(1. *School of Information Engineering*, *East China Jiaotong University*, *Nanchang* 330013, *China*; 2. *School of Electronic Information*, *Shanghai Dianji University*, *Shanghai* 201306, *China*; 3. *School of Computer Science & Engineering*, *Central South University*, *Changsha* 410083, *China*)

Abstract: Aiming at the problem that dynamic objects tend to interfere with the accuracy of SLAM mapping, this paper proposed a new RGB-D SLAM framework for dynamic environments, which combined neural networks in deep learning with motion information. First, The algorithm used the Mask R-CNN network to detect potential moving objects that may generate dynamic object masks. Second, the algorithm combined the optical flow method and Mask R-CNN to remove full dynamic feature points. Finally, the experimental results under the TUM RGB-D dataset showed that this algorithm can improve the pose estimation accuracy of the SLAM system in dynamic environments and perform better than the existing ORB-SLAM2.

Key words: simultaneous localization and Mapping; feature points; dynamic environment; semantic segmentation

0 引言

同步定位与地图构建(simultaneous localization and mapping, SLAM)[1]是人在未知环境下实现自主定位与建图的核心技术，它旨在利用自身装载的传感器对自身位姿进行估计，并以增量的方式对周围环境动态构建实时地图。经过近 20 年的发展，SLAM 技术已经在自动驾驶，无人驾驶，虚拟现实，增强现实，无人机等领域发挥了重要作用。当前 SLAM 系统所使用的传感器主要有相机、激光雷达、惯性测量单元(Inertial Measurement Units，IMU)等。由于视觉传感器的成本较低，许多功能强大的 SLAM 系统都使用了视觉传感器，且效果相对较好。视觉传感器又可以分为单目相机、双目相机、RGB-D 相机以及事件相机等，均已被开源方案广泛适用，如 ORB-SLAM2[2]、LSD-SLAM[3]、SVO[4]等。视觉同步定位与地图构建(Visual SLAM,VSLAM)[5]因采用视觉传感器逐渐成为 SLAM 领域的热门研究方向之一。

视觉 SLAM 按照视觉里程计的计算方法不同可以分为直接法和特征点法[6]。直接法基于光度不变假设，它不依赖特征点的提取和匹配，直接通过两帧之间的像素灰度值构建光度误差(Photometric Error)来求解相机位姿。特征点法基于特征点的匹配，它通过最小化重投影误差(Reprojection Error)来计算相机位姿与地图点的位置。Engel 等[3]提出的 LSD-SLAM(Large Scale Direct Monocular SLAM)是直接法中比较完整的 SLAM 系统。该系统适用于大规模场景，能够构建大尺度的，全局一致性的环境地图。其后提出的 DSO(Direct Sparse Odometry)[7]稀疏直接法的视觉里程计，在准确性，稳定性和速度上优于 LSD-SLAM。Forster 等[4]提出的半直接法视觉里程计 SVO(Semi-direct Visual Odoemtry)结合了基于特征点的方法和直接跟踪光流方法的优点。其后提出的最新版本 SVO2.0[8]基于视觉惯性里程计，支持透视、鱼眼和双目相机，可以生成轻量的、全局一致性的环境地图。以上的工作[3,4,7,8]采用了直接法的视觉里程计方案。MonoSLAM[9]是第一个在单目相机上实时运行的视觉 SLAM 系统。它采用 EKF(Extended Kalman Filter)作为后端，在前端跟踪稀疏特征点，算法效率高，但其稀疏的特征点容易跟踪丢失。Klein 等[10]提出的 PTAM(Parallel Tracking And Mapping)是最早提出将跟踪和建图分开作为两个线程的一种 SLAM 算法，是一种基于关键帧的单目视觉 SLAM 算法。PTAM 采用 FAST(Features from Accelerated Segment Test)作为特征提取方法来实现跟踪和建图。MurArtal 等[2]提出的 ORB-SLAM2 可以在大规模场景下实现长期运行。其增加了对双目摄像机和 RGB-D 深度摄像机的支持，是基于特征点跟踪方法的 SLAM 的成功应用。Carlos 等在 ORB-SLAM2 基础上提出的 ORB-SLAM3[11]，增加了视觉惯性里程计、多地图融合等功

收稿日期：2021-09-26；**修回日期：**2021-11-17 **基金项目：**国家自然科学基金资助项目(61963017，61863013)；江西省科技创新杰出青年人才项目(20192BCBL23004)

作者简介：张恒(1979-)，男，湖北汉川人，教授，硕导，博士，主要研究方向为智能机器人、深度学习与计算机视觉；徐长春(1996-)，男，硕士研究生，主要研究方向为视觉 SLAM；刘艳丽(1979-)，女(通信作者)，教授，硕导，博士，主要研究方向为智能机器人、机器视觉(liuyl@sdju.edu.cn)；廖志芳(1968-)，女，教授，硕导，博士，主要研究方向为开源软件分析与研究.

能，支持单目、双目以及 RGB-D 相机，同时支持针孔相机和鱼眼相机模型的 SLAM 系统。以上的工作[2, 9~11]采用了特征点法的视觉里程计方案。然而上述方法大多是在静态环境下成功实现，不能实时检测和处理动态场景中的动态物体，在定位和建图过程中不可避免地会产生干扰。

由于传统基于特征点的方法很容易受到纹理缺失导致的特征点不足，相机运动过快导致的特征不匹配，以及光照突变导致状态估计失败等一系列问题。为了使系统适应动态环境，越来越多的目标检测和语义分割方法被引入到 SLAM 系统中。其中比较流行的目标检测和语义分割方法有 SegNet[12]、Mask R-CNN[13]、YOLOV3[14]等。深度学习的目标检测和语义分割具有更高的准确率，在 SLAM 系统中得到了广泛的应用。DS-SLAM[15]基于 ORB-SLAM2，将语义分割网络[11]与运动一致性检查相结合，以减少动态对象的影响。DynaSLAM[16]同样基于 ORB-SLAM2，通过添加动态目标检测功能，在单目、双目和 RGB-D 数据集的动态场景中具有强大的功能。它可以通过结合 Mask R-CNN 和多视图几何模型对动态场景进行改进。DDL-SLAM[17]增加了动态对象分割的功能，采用 DUNet[18]提供像素级的语义分割和多视图几何相结合的方法作为预处理阶段过滤掉与动态目标相关的数据。DP-SLAM[19]基于动态关键点检测的移动概率传播模型，结合了几何约束和语义分割的结果来跟踪贝叶斯概率估计框架中的动态关键点，从而过滤掉与移动对象相关联的关键点。OFM-SLAM[20]使用 Mask-RCNN 实例分割网络和光流方法检测动态特征点。RDS-SLAM[21]建立在 ORB-SLAM3 基础之上，添加了语义线程和基于语义的优化线程，以便在动态环境中实时进行可靠的跟踪和建图。使用移动概率来更新和传播语义信息，该概率被保存在地图中，并使用数据关联算法从跟踪中去除异常值。在遮挡了太多的背景特征而无法成功地从背景中跟踪时，DOE-SLAM[22]可以利用物体的特征和预测的物体运动来估计摄像机的姿态，从而跟踪运动对象的姿态。然而，在某些情况以上提出的工作会导致两个问题。首先，当动态物体占据了场景图像中很大比例的时候，直接去除与移动物体相关的所有特征会导致图像特征点数量的减少，从而导致轨迹丢失，SLAM 定位和建图的准确性就会受*到很大影响。其次，具有移动能力但处于静止状态的物体出现在图像中，虽然它们当前是静止状态的，比如停靠在路上的汽车，如果直接将这些汽车上的特征点去除，一些原始的有用信息就会丢失，也会导致定位和建图的不可靠。

为了确保该系统能够适应复杂室内环境下定位和建图的要求，本文提出了一种基于语义信息和几何信息的动态场景下的 SLAM 框架，所提出的方法致力于从以下两个方面改进系统：a)提出了一种基于 Mask R-CNN 的语义分割的 RGB-D SLAM 系统减少动态对象的影响；b)将 Mask-RCNN 分割的语义信息与光流法检测出的几何信息相结合提高了动态物体的识别准确率，这极大地提高了本文算法的姿态估计精度和鲁棒性。

1　总体框架

作为成熟的 SLAM 方案之一，ORB-SLAM2 系统方案受 PTAM 提出的跟踪过程和建图过程并行设计的启发，创新性地提出了三种线程模式：实时跟踪特征点线程、局部建图优化线程、回环检测线程。ORB-SLAM2 的三线程结果实现了非常好的跟踪和建图效果，并且可以保证轨迹和建图的全局一致性。

图 1 显示了系统的整体框架，在 ORB-SLAM2 系统的基础上增加了语义分割模块和运动目标检测模块。语义分割模块用于分割出具体的实例，包括动态物体和静态物体。运动目标检测模块首先对输入的每一帧图像进行对象检测，用于获取图像中的类别信息。跟踪线程首先提取 ORB 特征点[23]，特征点与对象类别信息相关联。根据特征点的类别和特征点从参考帧到当前帧的运动信息，结合上一帧地图点的动态信息，可以得到每个特征点的动态概率，剔除动态概率高的特征点。从关键点生成的地图点被赋予相应的动态概率，该概率将被传播到下一帧。剩下的部分类似于 ORB-SLAM2 的流程。进入模块判断当前帧是否为关键帧后，系统进入局部建图和闭环检测线程。

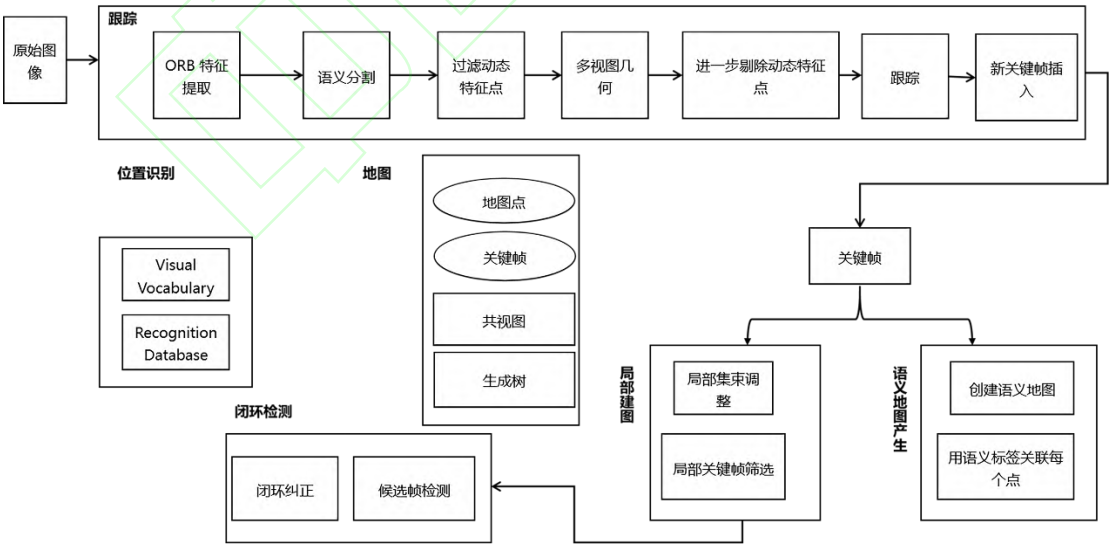


图 1　本文提出的算法框架

Fig. 1　The algorithmic framework proposed in this paper

1.1　问题描述

一般情况下，视觉 SLAM 问题可以用观测模型来描述：

$$\boldsymbol{z}_{k,j}=h(x_k,y_j)+\boldsymbol{v}_{k,j} \tag{1}$$

其中 x_k 表示 k 时刻相机在世界坐标中的位置，y_j 表示第 j 个路标点的位置。$\boldsymbol{z}_{k,j}$ 是相机拍摄图像中的像素坐标信息，对应于相机在 k 时刻观测到的路标。h 是非线性方程，$\boldsymbol{v}_{k,j}\sim\mathcal{N}(0,Q_{k,j})$ 是假设方差为零均值的高斯噪声。$Q_{k,j}$ 为观测方程的协方差矩阵。对于这种观测，误差项可以定义为

$$\boldsymbol{e}_{k,j}=\boldsymbol{z}_{k,j}-h(x_k,y_j) \tag{2}$$

然后，本文可以描述代价函数如下：

$$J(x)=\frac{1}{2}\sum_{k}^{m}\sum_{j}^{n}e_{k,j}(x)^T Q_{k,j}^{-1}e_{k,j}(x) \tag{3}$$

其中 x 表示所有摄像机姿态和地标位置。这是一个典型的非线性最小二乘问题。在静态环境下，通过迭代可以正确地解

决这一最小二乘问题。然而在动态环境中，移动路标会导致与观测值不匹配，从而导致错误的数据关联问题。为了解决动态环境下的问题，本系统需要从运动物体中去除特征点。

1.2　基于 Mask R-CNN 的运动目标检测

本文是在 PyTorch[24]基础上实现的 Mask R-CNN 语义分割。Mask R-CNN 不仅能够实现像素级的语义分割，还能够检测动态对象，如正在走路的人，行驶的汽车以及潜在可能运动的物体。Mask R-CNN 训练集采用 MS COCO 数据集[25]，共 80 种目标。实验是从这 80 个类中选择 20 个作为重点关注的对象，如“人”、“自行车”、“汽车”、“长凳”、“背包”、“猫”、“瓶子”、“椅子”、“沙发”、“床”、“电视”、“手提电话”、“书本”、“时钟”、“花瓶”等。因为考虑到该算法主要针对室内场景，本文的算法将人、椅子和显示器这三种类型的目标进行语义分割，将检测到的人作为动态目标，而将椅子和显示器作为静态目标，并将其语义信息添加到 SLAM 的建图中去。

目前最先进的基于特征点的 SLAM 算法是在姿态初始化时对两帧图像进行特征点对的匹配。然后，利用 RANSAC((Random Sample Consensus)算法[26]去除一些不匹配点对和动态点对。但是，当动态对象较多时，初始化位置并不准确。为了初始化一个相对鲁棒的摄像机姿态，本文使用 Mask R-CNN 分割结果。先通过 COCO 数据集中定义的类别将原始图像中的物体通过不同颜色进行标注，然后将获得的语义分割结果送到运动目标检测模块中，对物体进行进一步的剔除，以去掉动态目标中的特征点，只保留静态特征点。然后，通过匹配剩余的静态特征点来初始化摄像机姿态。如图 2 所示，Mask R-CNN 算法可以有效识别标注的各种静态和动态物体。

(a) 原始的 RGB 图像　　(b) 经过目标检测识别出来的图像　　(c) 对背景经过进一步的处理后得到的图像

图 2　使用 Mask R-CNN 进行语义分割的结果示例

Fig. 2　Example results of semantic segmentation using Mask R-CNN

2　语义分割

2.1　动态环境下的语义分割

近年来，随着深度学习在图像识别和语义分割中的成功应用，视觉 SLAM 中语义分割的研究逐渐受到重视。此外，机器人可以通过分割的语义信息增强对周围环境的理解，进而去除动态场景中的运动目标，这有助于提高 SLAM 系统的准确性和鲁棒性。因此，运动目标的语义分割是提高动态视觉 SLAM 系统性能的有效方法。图 3 是使用 DS-SLAM、DynaSLAM 和本文的方法进行语义分割的示例。

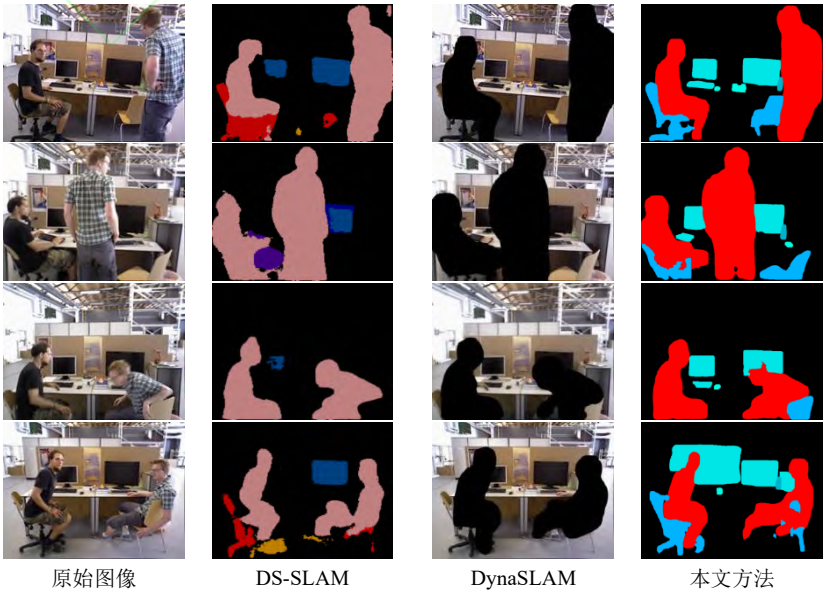
第一列是输入帧的原始图像。第二列是 DS-SLAM 下 SegNet 分割的语义图像。第三列是 DynaSLAM 下 Mask R-CNN 检测到的动态对象的语义分割结果。第四列显示了本文使用的 Mask R-CNN 方法检测到的动态对象的语义分割结果。

在第一行中，DS-SLAM 的语义分割效果相比 DynaSLAM 和本文提出的方法效果要差一些。DS-SLAM 没有对右边椅子进行分割，DynaSLAM 只分割了作为动态物体的人，而本文提出的方法对椅子和显示器的分割质量都有所提高。

在第二行中，可以看到在 DS-SLAM 的分割结果中，显示器只分割了一小部分，两边的椅子也没有被分割出来，而人体的一部分被错误地分割成了别的类别。与 DS-SLAM 技术相比，本文提出的方法提高了椅子和显示器的分割质量。

在第三行中，DS-SLAM 的分割结果中没有椅子，靠近人的显示器也没有完全被分割。与 DS-SLAM 相比，本文提出的方法对椅子和显示器的分割更加合理。

在第四行中，DS-SLAM 虽然分割出了部分椅子和显示器，但把背景也当作类别分割出来了。值得注意的是，DS-SLAM 和本文提出的方法对于这些小的物体，分割的结果也不是很好，因为它们很容易被划分到背景中。如第三行和第四行，右边人的头和身体被错误地分割成了显示器和椅子的一部分。但从总体来说，本文使用的语义分割方法能很好地适应各种情况。



原始图像　　DS-SLAM　　DynaSLAM　　本文方法

图 3　使用 DS-SLAM、DynaSLAM 和本文的方法进行语义分割的示例

Fig. 3　Example of semantic segmentation using DS-SLAM，dynaslam and the systems in this paper

2.2　基于光流法的动态特征点检测

利用稀疏金字塔光流(Lucas-Kanade，LK)[27]跟踪图像中的特征点，可以得到特征点之间的对应关系。特征点法需要描述子的计算和特征点的匹配过程，而光流法避开了计算和匹配描述子带来的时间，因此具有更好的实时性。

通过跟踪动态环境下每帧图像之间的光流信息，可以得到每帧图像之间的特征点匹配关系，从而恢复每帧图像之间的位置关系。基础矩阵 F 描述了这种相对变换关系，其解可以用 8 对匹配点来计算。

给定一对匹配点，它们的标准化坐标是 $p_1=[u_1,v_1,1]^T$ 和 $p_2=[u_2,v_2,1]^T$ 它们之间满足表达式：

$$s_1p_1=KP, s_2p_2=K(RP+t) \tag{4}$$

其中，这里 K 为相机的内参矩阵，R，t 为两个坐标系之间的相机运动。

根据对极几何约束，有如下公式：

$$(u_2,v_2,1)\begin{pmatrix}e_1 & e_2 & e_3\\ e_4 & e_5 & e_6\\ e_7 & e_8 & 1\end{pmatrix}\begin{pmatrix}u_1\\ v_1\\ 1\end{pmatrix}=0 \tag{5}$$

其中 F 就是中间的矩阵，上式可以进一步写成如下形式：

$$p_2^TFp_1=[u_2,v_2,1]F[u_1,v_1,1]^T=0 \tag{6}$$

利用基础矩阵 F，可以将前一帧中的关键点 p_1 投影到当前帧中，得到当前帧中的极线 l_2。通过计算光流跟踪得到的关键点到极线的距离，确定其是否为动态点：

$$l_2=[X,Y,Z]^T=Fp_1=F[u_1,v_1,1]^T \tag{7}$$

其中，X、Y、Z 为极线的法向量，F 为基础矩阵。

从光流跟踪的匹配特征点 p_2 到对应极线 l_2 的距离 D 可由下式计算：

$$D=\frac{|p_2^TFp_1|}{\sqrt{||X||^2+||Y||^2}}>\sigma \tag{8}$$

其中 σ 为阈值，如果满足上式，则 p_1 和 p_2 是动态特征点，然后将这些动态特征点从建图中剔除出去。

图 4 (a)为 ORB-SLAM2 的光流法检测结果，它并没有把动态物体剔除出去，如行人身上的特征点。图 4(b)为使用 YOLOv3 与光流法结合去除属于动态对象特征点的实验结果。跟踪时消除了落在行人身上的特征点，而只保留静态特征点。图 4(c)为本文提出的采用 Mask R-CNN 与光流法检测相结合的实验结果。黄色特征点表示这些点位于动态对象的边界外，但通过本文的算法将它们视为可靠的特征点。蓝色特征点表示这些点位于静态对象内，如图中可以看到的显示器和椅子，它们可以用于定位和建图。黄色特征点和蓝色特征点构成了本系统用于跟踪的静态特征点集。

(a) ORB-SLAM2 的光流法检测结果　　(b)YOLOv3 去除动态点的光流法　　(c)本文使用的方法

图 4　与光流法结合的各种 SLAM 系统的检测结果

Fig. 4　Detection results of various SLAM systems combined with the optical flow method

3　实验结果

3.1　TUM 数据集

本文在 TUM 数据集[28]上进行了实验。该数据集使用一个 RGB-D Kinect 摄像头，提供彩色和深度图像以及准确的真实轨迹，并包含不同室内环境中的 39 个序列。根据场景中是否有动态对象，本文将序列分为静态场景和动态场景。实验在 CPU 为 Intel Xeon E5-2689，GPU 为 GeForce GTX1070，内存为 64GB 的计算机上进行。

为了方便起见，本文用 fr3，half，w，s 来代表 freiburg3，halfsphere，walking，sitting 作为序列的名称。从 TUM RGB-D 数据集中选取了 8 组序列，将所提出的系统与 ORB-SLAM2、DS-SLAM 和 DynaSLAM 进行比较。使用绝对轨迹误差(Absolute Trajectory Error，ATE)和相对位姿误差(Relative Pose Error，RPE)来进行定量评估。ATE 是估计位姿与实际位姿之间的直接差值，可以非常直观地反映算法精度和轨迹全局一致性。RPE 包含相对平移误差和相对旋转误差，直接测量里程计的误差。

3.2　定量评估

本文给出了绝对轨迹误差的均方根误差(Root Mean Square Error，RMSE)和标准差(Standard Deviation，S.D.)的值，均方根误差(RMSE)描述了所估计的值与真实值之间的偏差，因此其值越小，代表所估计的轨迹越接近真实值。标准差(S.D.)反映了系统轨迹估计的离散程度。以上两个指标相结合能更好地证明系统的鲁棒性和稳定性。为了更好地反映出本文算法的性能，本文提出将 ORB-SLAM2 与该系统作对比。如表 1~4 所示，RMSE 和 S.D.的值计算公式如下：

$$\sigma_{RMSE}=\left(1-\frac{\alpha}{\beta}\right)\times 100\% \tag{9}$$

$$\sigma_{S.D.}=\left(1-\frac{\gamma}{\mu}\right)\times 100\% \tag{10}$$

其中 σ_{RMSE} 表示本文算法的 RMSE 值的改进，α 表示本文算法的 RMSE 值，β 表示 ORB-SLAM2 的 RMSE 值。$\sigma_{S.D.}$ 表示本文算法的 S.D.值的改进，γ 表示本文算法的 S.D.值，μ 表示 ORB-SLAM2 的 RMSE 值。

本文算法与 ORB-SLAM2、DS-SLAM、DynaSLAM 算法的比较结果如表 1~4 所示。对于高动态序列，本文算法的绝对轨迹误差 ATE 的均方根误差 RMSE 和标准差 S.D.在 fr3/w/xyz 序列下分别为 98.92%和 99.07%。该对比实验表明本文算法在高动态环境下具有良好的性能。对于低动态序列，本文算法的绝对轨迹误差 ATE 的均方根误差 RMSE 和标准差 S.D.在 fr3/s/rpy 下分别仅为 11.64%和 34.35%，在 fr3/s/half 下分别仅为 21.61%和 2.38%。原因是在低动态序列中，大多数对象是静态的，物体运动缓慢，运动物体在环境中占的比例小。ORB-SLAM2 在静态环境下可以获得良好的效果，因此在低动态序列下很难提高性能。而且在低动态环境中可以使用对象上的特征点，并且它们不会影响跟踪性能，所以在这种情况下，本文算法的改进并不明显。与其他两种动态环境下的 SLAM 方法相比，本文算法优于 DS-SLAM，并且大多数序列的性能都优于 DynaSLAM。

表 1　绝对轨迹误差的对比结果

Tab. 1　Comparison results of absolute trajectory error (ATE)

序列	ORB-SLAM2		DS-SLAM		DynaSLAM		本文方法	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/xyz	0.5560	0.3009	0.1631	0.0596	0.0184	0.0087	**0.0060**	**0.0028**
fr3/w/rpy	0.8790	0.4035	0.1856	0.0972	0.0425	0.0282	**0.0306**	**0.0164**
fr3/w/static	0.4124	0.0907	0.0064	0.0027	0.0064	0.0031	**0.0025**	**0.0011**
fr3/w/half	0.7270	0.3905	0.0328	0.0169	**0.0250**	**0.0103**	0.0327	0.0217
fr3/s/xyz	0.0127	0.0062	0.0185	0.0118	0.0143	0.0065	**0.0053**	**0.0021**
fr3/s/rpy	0.0378	0.0230	**0.0270**	**0.0153**	0.0865	0.0516	0.0334	0.0151
fr3/s/static	0.0125	0.0043	0.0100	0.0043	0.0085	0.0051	0.0023	0.0011
fr3/s/half	0.0398	0.0210	**0.0141**	**0.0057**	0.0200	0.0081	0.0312	0.0205

表 2　相对轨迹误差平移部分的对比结果

Tab. 2　Comparison results of relative pose error(RPE)

序列	ORB-SLAM2		DS-SLAM		DynaSLAM		本文方法	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/xyz	0.0477	0.0305	0.0368	0.0269	0.0196	0.0121	**0.0140**	**0.0060**
fr3/w/rpy	0.0452	0.0332	0.0767	0.0622	0.0433	0.0342	**0.0098**	**0.0071**
fr3/w/static	0.0281	0.0225	0.0076	0.0038	0.0074	0.0040	**0.0014**	**0.0008**
fr3/w/half	0.0360	0.0222	0.0249	0.0138	0.0179	0.0102	**0.0164**	**0.0094**
fr3/s/xyz	0.0184	0.0099	0.0242	0.0160	0.0155	0.0079	**0.0096**	**0.0050**
fr3/s/rpy	0.0318	0.0252	0.0237	0.0136	0.0412	0.0269	**0.0038**	**0.0018**
fr3/s/static	0.0168	0.0061	0.0168	0.0078	0.0138	0.0080	**0.0014**	**0.0008**
fr3/s/half	0.0189	0.0144	0.0141	0.0079	0.0192	0.0107	**0.0135**	**0.0067**

表 3　相对轨迹误差旋转部分的对比结果

Tab. 3　Comparison results of the rotation part of RPE

序列	ORB-SLAM2		DS-SLAM		DynaSLAM		本文方法	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/xyz	1.006	0.6331	0.8633	0.6026	**0.5946**	**0.3494**	0.9463	0.6200
fr3/w/rpy	**0.9756**	**0.6785**	1.5435	1.1576	1.0239	0.7642	1.0430	0.7123
fr3/w/static	0.5416	0.4012	0.2258	**0.0821**	**0.2126**	0.1217	0.4099	0.2338
fr3/w/half	0.8473	0.4837	0.6355	0.3386	0.5735	0.3285	0.7664	0.4783
fr3/s/xyz	0.5631	0.3270	0.5814	0.3277	0.5416	0.3406	0.5422	**0.2925**
fr3/s/rpy	0.8356	0.5747	0.7285	0.3595	0.7210	0.4064	**0.6038**	**0.3441**
fr3/s/static	0.5384	0.2233	0.4766	0.2246	**0.4146**	0.2213	0.5315	0.1976
fr3/s/half	0.5571	0.3398	0.5400	**0.2656**	0.5904	0.3176	**0.4778**	0.2661

表 4　绝对轨迹误差和相对位姿误差的改进结果

Tab. 4　Improvement results of ATE and RPE

序列	改进值 1		改进值 2		改进值 3	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/xyz	**98.92%**	**99.07%**	70.65%	80.33%	5.93%	2.07%
fr3/w/rpy	**96.52%**	**95.94%**	78.32%	78.61%	-6.91%	4.98%
fr3/w/static	**93.94%**	**98.79%**	**95.02%**	**96.44%**	24.32%	41.72%
fr3/w/half	**95.5%**	**94.44%**	54.44%	57.66%	9.55%	1.12%
fr3/s/xyz	58.27%	66.13%	47.83%	49.49%	3.71%	10.55%
fr3/s/rpy	11.64%	34.35%	88.05%	92.86%	27.74%	40.13%
fr3/s/static	81.6%	74.42%	**91.67%**	**86.89%**	1.28%	11.51%
fr3/s/half	21.61%	2.38%	28.57%	53.47%	14.23%	21.69%

从表 2 可以看出，本文算法在低动态场景和高动态场景下比原始的 ORB-SLAM2 都有了很大的改进。在 fr3/w/xyz，fr3/w/rpy 等 8 个序列上，本文算法的结果相对更好。在 fr3/w/half 序列上，本文算法相对于 DS-SLAM 和 Dyna-SLAM 的结果非常接近。DS-SLAM 的 S.D.值取得了较好的结果。从表 3 可以看出，本文算法在 fr3/w/half 和 fr3/w/xyz 序列上得到了最好的结果。DynaSLAM 在 fr3/w/rpy 序列上取得了更好的结果，但本文算法相对优于 DS-SLAM 和 ORB-SLAM2。值得注意的是，ORB-SLAM2、DS-SLAM 和 DynaSLAM 在 fr3/w/rpy 序列上的 RMSE 值没有明显改善。在 fr3/w/static 序列上，DynaSLAM 的 RMSE 值和本文算法的 S.D.值分别得到了更好的结果，并且本文算法的 RMSE 值优于 DS-SLAM 的 RMSE 值。事实上，这三个系统的结果非常接近。本文算法的 RMSE 值和 DS-SLAM 的 S.D.值在 fr3/w/xyz 上分别取得了较好的结果。从表 4 可以看出，本文算法在高动态场景下的改进相较于 ORB-SLAM2 有了很大的提升，但在低动态环境下改进效果不是那么明显。

与原来的 ORB-SLAM2 系统相比，本文算法可以大大提升高动态序列的精度。具体来说，对于低动态序列，平均可以达到 20%以上的改进。对于高动态场景，改进更加明显，可以达到 90%以上。结果表明，该方法可以进一步消除动态目标的干扰，从而减少优化过程中的位姿误差。

3.3　定性评估

为了更进一步地评估系统，选取了两个有代表性的序列与 DS-SLAM，DynaSLAM 和本文算法作比较。其中 fr3/w/xyz 是高动态环境下的序列，fr3/s/half 是低动态环境下的序列。蓝色的实线表示 DS-SLAM 系统估计的轨迹，绿色的实线表示 DynaSLAM 系统估计的轨迹，红色的实线表示本文算法估计的轨迹。黑色的虚线表示相机的真实轨迹。本文的轨迹图是用 evo[29]工具画出来的。该软件包可以用于评估和比较 SLAM 算法的轨迹误差，包括绝对轨迹误差和相对位姿误差。从图 5 和 6 可以看出，在 fr3/w/xyz 序列下 DS-SLAM 系统估计的轨迹相比真实轨迹有很大的漂移。而在 fr3/s/half 序列下 DS-SLAM 和 DynaSLAM 系统估计的轨迹与本文算法估计的轨迹与真实轨迹几乎重合。说明 DS-SLAM 和 DynaSLAM 系统在低动态环境下运行情况良好，而在高动态环境下很容易造成轨迹丢失。本文算法在高动态环境和低动态环境下很好地克服了 DS-SLAM 和 DynaSLAM 系统的弊端。

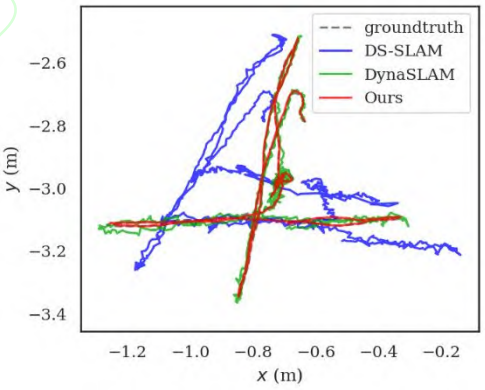


图 5　在 fr3/w/xyz 序列下估计轨迹和真实轨迹比较结果

Fig. 5　Comparison result of estimated trajectory and real trajectory under fr3/w/xyz sequence

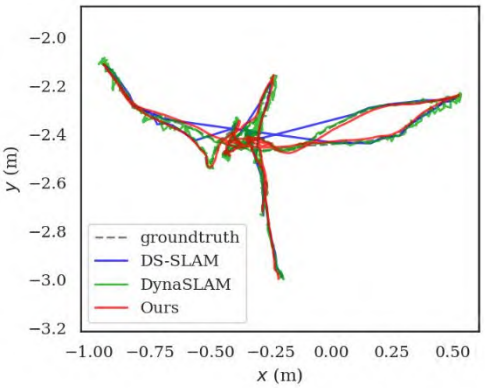


图 6　在 fr3/s/half 序列下估计轨迹和真实轨迹比较结果

Fig. 6　Comparison result of estimated trajectory and real trajectory under fr3/s/half sequence

图 7~9 是 DS-SLAM，DynaSLAM 和本文算法在 fr3/w/xyz 序列下的绝对轨迹误差曲线图。在 fr3/w/xyz 序列下 DS-SLAM 系统与真实轨迹有很大的差别，而本文算法预测的轨迹与真实轨迹几乎保持一致。这是因为 DS-SLAM 估计的轨迹由于动态物体不移动或移动缓慢，因此和真实的轨迹比较有很大的差异。图 10~12 是 DS-SLAM，DynaSLAM 和本文算法在 fr3/s/half 序列下的绝对轨迹误差曲线图。在 fr3/s/half 序列下 DS-SLAM 系统与真实轨迹的误差比较大，DynaSLAM 系统与真实轨迹的误差相对较小，本文算法预测的轨迹与真实轨迹的误差也很小。说明在低动态环境下两个系统的误差很相似，但是本文算法预测的轨迹更接近真实的轨迹。

最后，DS-SLAM 只将人作为分割的动态对象，而本文算法预先定义了 20 个潜在动态或可移动的物体，并在 RGB-D 数据集中进行了评估。本文算法更适用于各种复杂的场景。DynaSLAM 将分割后的内容直接视为动态对象，并且只在 RGB-D 情况下使用多视图几何提取动态特征点。本文算法添加了运动目标检测模块可以避免在静态掩模上丢弃过多的特征点，能够解决剩余静态特征点太少的问题。因此，本文算法比直接去除掩模中所有特征点的方法具有更好的鲁棒性。DS-SLAM 和 DynaSLAM 在进行语义分割时计算量会比较大，容易降低 SLAM 系统运行效率，导致对动态特征点跟踪失败。本文算法在语义分割的过程中提高了算法分割的效率，计算量相对较小，能够很好地跟踪动态特征点。但与较为先进的 SLAM 系统相比，本文算法在实时性方面仍存在差距。因此，下一步的研究方向是进一步优化语义分割网络，进一步提高系统的实时性。上述定性结果表明，本文算法在鲁棒性和准确性方面有显著提高，特别是在高动态环境下的序列中。

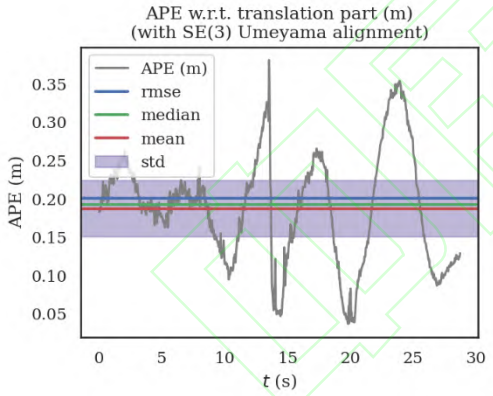


图 7　在 fr3/w/xyz 序列下 DS-SLAM 系统的绝对轨迹误差曲线

Fig. 7　The absolute trajectory error curve of the DS-SLAM system under fr3/w/xyz sequence

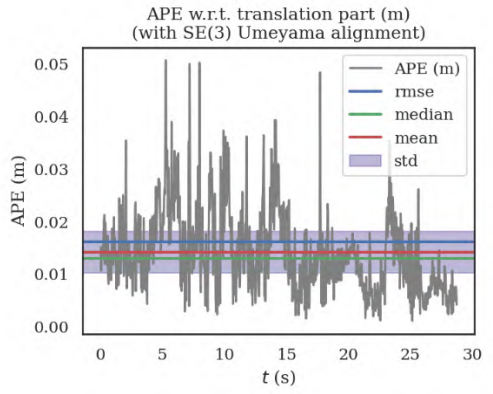


图 8　在 fr3/w/xyz 序列下 DynaSLAM 系统的绝对轨迹误差曲线

Fig. 8　The absolute trajectory error curve of the DynaSLAM system under fr3/w/xyz sequence

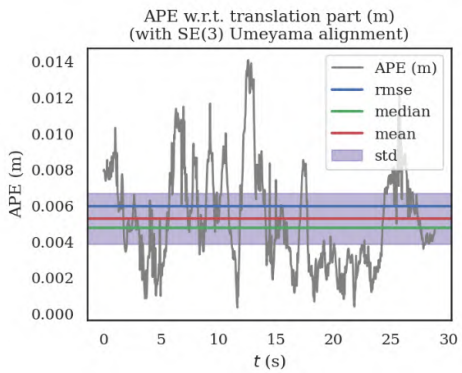


图 9　在 fr3/w/xyz 序列下本文系统的绝对轨迹误差曲线

Fig. 9　The absolute trajectory error curve of the system in this paper under fr3/w/xyz sequence

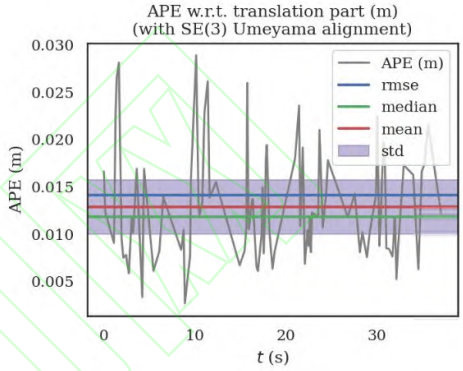


图 10　在 fr3/s/half 序列下 DS-SLAM 系统的绝对轨迹误差曲线

Fig. 10　The absolute trajectory error curve of the DS-SLAM system under fr3/s/half sequence

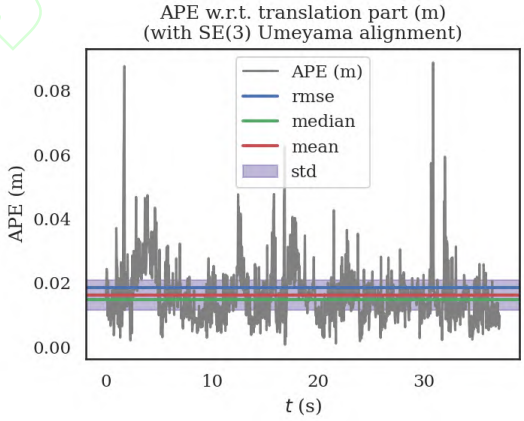


图 11　在 fr3/s/half 序列下 DynaSLAM 系统的绝对轨迹误差曲线

Fig. 11　The absolute trajectory error curve of the DynaSLAM system under fr3/s/half sequence

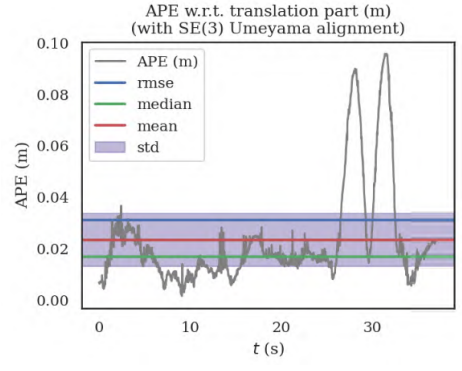


图 12　在 fr3/s/half 序列下本文系统的绝对轨迹误差曲线

Fig. 12　The absolute trajectory error curve of the system in this paper under fr3/s/half sequence

4 实验结果

本文提出了一种基于 ORB-SLAM2 的 Mask R-CNN 动态物体剔除方法。通过光流和语义分割过滤特征点，检测和消除动态特征点，利用稳定的静态特征点进行动态场景下的运动估计，完成语义地图的构建。本文使用公开的 TUM 数据集以及搭建的实验平台对比了 ORB-SLAM2、DS-SLAM 和 DynaSLAM 三个主流算法的位姿估计精度。评估结果表明，该系统在高动态场景下精度和速度方面都优于现有的方法。实验结果表明，本文提出的算法在动态环境下具有可靠的优越性、准确性和鲁棒性。然而，该算法在某些大规模的室外场景下很容易跟踪丢失，仅适用于室内场景，未来可考虑改进本文的语义分割网络以适应各种复杂多变的情况，更好地实现机器人的路径规划与导航。

参考文献：

[1] 刘浩敏, 章国锋, 鲍虎军. 基于单目视觉的同时定位与地图构建方法综述 [J]. 计算机辅助设计与图形学学报, 2016, 28 (6): 855-868. (Liu Haomin, Zhang Guofeng, Bao Hujun. Overview of simultaneous localization and mapping methods based on monocular vision [J]. Journal of Computer Aided Design and Graphics, 2016, 28 (6): 855-868.)

[2] Mur-Artal R, Tardos J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras [J]. IEEE Trans on Robotics, 2017, 33 (5): 1255-1262.

[3] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM [C]// Proc of European Conference on Computer Vision. Springer, Cham, 2014: 834-849.

[4] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry [C]// Proc of IEEE International Conference on Robotics and Automation (ICRA) . IEEE, 2014: 15-22.

[5] 吴凡, 宗艳桃, 汤霞清. 视觉 SLAM 的研究现状与展望 [J]. 计算机应用研究, 2020, 37 (8): 2248-2254. (Wu Fan, Zong Yantao, Tang Xiaqing. Research status and prospects of visual SLAM [J]. Application Research of Computers, 2020, 37 (8): 2248-2254.)

[6] 谷晓琳, 杨敏, 张燚, 等. 一种基于半直接视觉里程计的 RGB-D SLAM 算法 [J]. 机器人, 2020, 42 (1): 39-48. (Gu Xiaolin, Yang Min, Zhang Yi, *et al.* An RGB-D SLAM algorithm based on semi-direct visual odometry [J]. Robot, 2020, 42 (1): 39-48.)

[7] Wang R, Schworer M, Cremers D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras [C]// Proc of IEEE International Conference on Computer Vision. 2017: 3903-3911.

[8] Forster C, Zhang Z, Gassner M, *et al.* SVO: Semidirect visual odometry for monocular and multicamera systems [J]. IEEE Trans on Robotics, 2016, 33 (2): 249-265.

[9] Davison A J, Reid I D, Molton N D, *et al.* MonoSLAM: Real-time single camera SLAM [J]. IEEE Trans on Pattern Analysis and Machine Intelligence. 2007, 29 (6): 1052-1067.

[10] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces [C]// Proc of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, 2007: 225-234.

[11] 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 Trans on Robotics, 2021.

[12] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence. 2017, 39 (12): 2481-2495.

[13] He K, Gkioxari G, Dollár P, *et al.* Mask r-cnn [C]// Proc of IEEE International Conference on Computer Vision. 2017: 2961-2969.

[14] 邹斌, 林思阳, 尹智帅. 基于 YOLOv3 和视觉 SLAM 的语义地图构建 [J]. 激光与光电子学进展, 2020, 57 (20): 201012. (Zou Bin, Lin Siyang, Yin Zhishuai. Semantic map construction based on YOLOv3 and visual SLAM [J]. Progress in Laser and Optoelectronics, 2020, 57 (20): 201012.)

[15] Yu C, Liu Z, Liu X J, *et al.* DS-SLAM: A semantic visual SLAM towards dynamic environments [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) . IEEE, 2018: 1168-1174.

[16] 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.

[17] Ai Y, Rui T, Lu M, *et al.* DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with deep learning [J]. IEEE Access. 2020, 8: 162335-162342.

[18] Jin Q, Meng Z, Pham T D, *et al.* DUNet: A deformable network for retinal vessel segmentation [J]. Knowledge-Based Systems. 2019, 178: 149-162.

[19] Li A, Wang J, Xu M, *et al.* DP-SLAM: A visual SLAM with moving probability towards dynamic environments [J]. Information Sciences. 2021, 556: 128-142.

[20] Zhao X, Zuo T, Hu X. OFM-SLAM: A Visual Semantic SLAM for Dynamic Indoor Environments [J]. Mathematical Problems in Engineering. 2021, 2021: 1-16.

[21] Liu Y, Miura J. RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods [J]. IEEE Access, 2021, 9: 23772-23785.

[22] Hu X, Lang J. DOE-SLAM: Dynamic Object Enhanced Visual SLAM [J]. Sensors, 2021, 21 (9): 3091.

[23] Rublee E, Rabaud V, Konolige K, *et al.* ORB: An efficient alternative to SIFT or SURF [C]// Proc of International Conference on Computer Vision. Ieee, 2011: 2564-2571.

[24] Paszke A, Gross S, Massa F, *et al.* Pytorch: An imperative style, high-performance deep learning library [J]. Advances in Neural Information Processing Systems. 2019, 32: 8026-8037.

[25] Lin T Y, Maire M, Belongie S, *et al.* Microsoft coco: Common objects in context [C]// Proc of European Conference on Computer Vision. Springer, Cham, 2014: 740-755.

[26] Chum O, Matas J, Kittler J. Locally optimized RANSAC [C]// Proc of Joint Pattern Recognition Symposium. Springer, Berlin, Heidelberg, 2003: 236-243.

[27] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision [C]. Proc of Vancouver, British Columbia, 1981.

[28] Sturm J, Engelhard N, Endres F, *et al.* A benchmark for the evaluation of RGB-D SLAM systems [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 573-580.

[29] Grupp M. evo: Python package for the evaluation of odometry and SLAM [J]. 2017. https://github. com/MichaelGrupp/evo