动态场景下基于 YOLOv5 和几何约束的视觉 SLAM 算法

王鸿宇[1]，吴岳忠[1,2*]，陈玲姣[1]，陈茜[1]
（1.湖南工业大学 轨道交通学院，湖南 株洲 412007；
2.湖南省智能信息感知及处理技术重点实验室，湖南 株洲 412007）

摘要：目的 移动智能体在执行同步定位与地图构建（Simultaneous Localization and Mapping，SLAM）的复杂任务时，动态物体的干扰会导致特征点间的关联减弱，系统定位精度下降，为此提出一种面向室内动态场景下基于 YOLOv5 和几何约束的视觉 SLAM 算法。**方法** 首先，以 YOLOv5s 为基础，将原有的 CSPDarknet 主干网络替换成轻量级的 MobileNetV3 网络，可以减少参数、加快运行速度，同时与 ORB-SLAM2 系统相结合，在提取 ORB 特征点的同时获取语义信息，并剔除先验的动态特征点。然后，结合光流法和对极几何约束对可能残存的动态特征点进一步剔除。最后，仅用静态特征点对相机位姿进行估计。**结果** 在 TUM 数据集上的实验结果表明，与 ORB-SLAM2 相比，在高动态序列下的 ATE 和 RPE 都减少了 90%以上，与 DS-SLAM、Dyna-SLAM 同类型系统相比，在保证定位精度和鲁棒性的同时，跟踪线程中处理一帧图像平均只需 28.26 ms。**结论** 该算法能够有效降低动态物体对实时 SLAM 过程造成的干扰，为实现更加智能化、自动化的包装流程提供了可能。

关键词：视觉 SLAM；动态场景；目标检测；光流法；对极几何约束
中图分类号：TB486；TP242　　**文献标志码：**A　　**文章编号：**1001-3563(2024)03-0208-10
DOI：10.19554/j.cnki.1001-3563.2024.03.024

Visual SLAM Algorithm Based on YOLOv5 and Geometric Constraints in Dynamic Scenes

WANG Hongyu[1], *WU Yuezhong*[1,2*], *CHEN Lingjiao*[1], *CHEN Xi*[1]
(1. School of Railway Transportation, Hunan University of Technology, Hunan Zhuzhou 412007, China;
2.Key Laboratory for Intelligent Information Perception and Processing Technology, Hunan Zhuzhou 412007, China)

ABSTRACT: When mobile intelligence agent performs the complex task of Simultaneous Localization And Mapping (SLAM), the interference of dynamic objects will weaken the correlation between feature points and the degradation of the system's localization accuracy. In this regard, the work aims to propose a visual SLAM algorithm based on YOLOv5 and geometric constraints for indoor dynamic scenes. First, based on YOLOv5s, the original CSPDarknet backbone network was replaced by a lightweight MobileNetV3 network, which could reduce parameters and speed up operation, and at the same time, it was combined with the ORB-SLAM2 system to obtain semantic information and eliminate a priori dynamic feature points while extracting ORB feature points. Then, the possible residual dynamic feature points were further culled by combining the optical flow method and epipolar geometric constraints. Finally, only static feature points were used for camera position estimation. Experimental results on the TUM data set showed that both ATE and RPE were

收稿日期：2023-10-25
基金项目：国家重点研发计划项目（2022YFE010300）；湖南省自然科学基金项目（2021JJ50050，2022JJ50051，2023JJ30217）；湖南省教育厅科学研究项目（22A0422，21A0350，21B0547，21C0430）；中国高校产学研创新基金重点项目（2022IT052）；湖南省研究生创新基金项目（CX20220835）
*通信作者

reduced by more than 90% on average under high dynamic sequences compared with ORB-SLAM2, and the processing of one frame in the tracking thread took only 28.26 ms on average compared with the same type of systems of DS-SLAM and Dyna-SLAM, while guaranteeing localization accuracy and robustness. The algorithm can effectively reduce the interference caused by dynamic objects to the real-time SLAM process. It provides a possibility for more intelligent and automatic packaging process.

KEY WORDS: visual SLAM; dynamic scene; target detection; optical flow method; epipolar geometric constraints

视觉 SLAM（Simultaneous Localization and Mapping）技术在包装行业的应用日益广泛，其通过分析相机捕获的图像信息，实现对环境的实时感知和三维重建。例如，在自动化包装生产线上，运用视觉 SLAM 技术能够使机器人精准辨认并定位包装盒，从而精确执行抓取、搬运和摆放的任务。这不仅大幅提高了包装的效率，同时确保了作业的精准度，降低了对人工的依赖。视觉 SLAM 在应对复杂和动态变化的包装环境时展现出其独特优势，能够实时更新地图信息以适应环境变化，确保定位精度。例如，在处理不同大小和形状商品的包装任务时，视觉 SLAM 能够协助机器人迅速适应新任务，避免了重新设定程序的繁琐过程。

随着计算机图形算力的不断提高，大量优秀的视觉 SLAM 算法被相继提出[1]，主要分为如 PTAM[2]（Parallel Tracking and Mapping）、ORB-SLAM2[3]的特征点法和 DSO[4]（Direct Sparse Odometry）、D3VO[5]（Deep Direct Dense Visual Odometry）的直接法 2 类。然而，上述视觉 SLAM 算法均假设在静态环境下运行，但在实际场景下会出现实时运动的行人和车辆（高动态物体）和临时移动的物品（低动态物体），致使假设并不成立。

针对上述问题，国内外学者的方法主要分为 2 种。一种是基于几何的传统算法，Dai 等[6]使用 Delaunay 三角剖分来为地图中的点构建图状结构，以识别它们的邻近性。接着移除在多个关键帧中有差异的边缘，最终实现动态物和静态背景的区分。张有全等[7]提出了一种紧耦合的视觉惯性 SLAM 系统，该系统采用了直接法和共视图优化。在跟踪线程中，结合 IMU 信息和基于稀疏图像对齐的直接法来进行初始位姿的估计。虽然依靠几何算法能在一定程度上提高定位精度，但当场景中存在较多弱纹理区域、光照强度变化时，系统性能就会明显下降，甚至失效。另一种是随着深度学习在计算机视觉领域中不断取得突破性的成果，基于目标检测、语义分割等深度学习方法对特征点做先验语义信息标注，与传统视觉 SLAM 结合以剔除动态物体的特征点[8]。Zhong 等[9]提出了基于目标检测网络 SSD[10]的 Detect-SLAM 系统，只检测 ORB-SLAM 关键帧中的运动对象，利用语义信息并通过一种实时传播关键点运动概率的方法剔除动态特征点。Yu 等[11]提出了 DS-SLAM 系统，该系统在 ORB-SLAM2 的基础上结合 SegNet[12]实时语义分割网络与运动一致性检测，可以有效降低动态环境中行人的影响。同样，在 ORB-SLAM2 基础上 Bescos 等[13]提出了 DynaSLAM 系统，通过神经网络 Mask R-CNN[14]分割具有先验信息的动态目标，再利用多视图几何分割潜在的动态特征点，但是算力和功耗要求高，实时性较差。Yan 等[15]提出一种结合几何信息和语义信息的 DGS-SLAM 系统，设计了一种语义关键帧选择策略对 ORB 关键帧与语义关键帧区分，同时融合几何与语义（YOLACT）先验残差对运动目标进行检测。

本文在 ORB-SLAM2 的基础框架上进行改进，主要改进和创新如下。

1）添加了一条新的动态目标检测线程，用轻量级的 MobileNetV3 网络替换 YOLOv5s 原有的主干网络，并将目标检测提取到的场景语义信息和 ORB 特征点相结合获取图像信息，利用目标检测算法预测先验的动态区域并剔除其中的动态特征点。

2）提出一种动态特征点剔除策略，首先剔除被先验语义信息标注的动态特征点，然后，结合光流法和对极几何约束对可能残存的动态特征点进一步剔除。

1　系统概述

1.1　系统框架

ORB-SLAM2 系统的良好性能是基于静态场景的假设，实际场景中，动态物体会直接影响特征点间的关联匹配，导致建图时出现“鬼影”等现象。因此，本文在原有 ORB-SLAM2 系统的视觉里程计中，添加了目标检测线程和剔除动态特征点模块。改进后的系统框架如图 1 所示，利用轻量级的目标检测网络 YOLOv5s-MobileNetV3 检测图像序列中的运动目标并获取语义信息，在视觉里程计中提取到 ORB 特征点的同时根据语义信息剔除先验的动态特征点。由于环境中可能会存在潜在的动态物体以及运动模糊的对象，而出现被目标检测网络漏检的情况。因此，在完成相邻 2 帧的待定静态特征点匹配后，利用 RANSAC 算法得到两图像间的基础矩阵。然后结合对极几何约束和金字塔 LK 光流法对可能残存的动态特征点进一步剔除[16]。最后，仅用剩余的静态特征点对相机位姿进行估计。

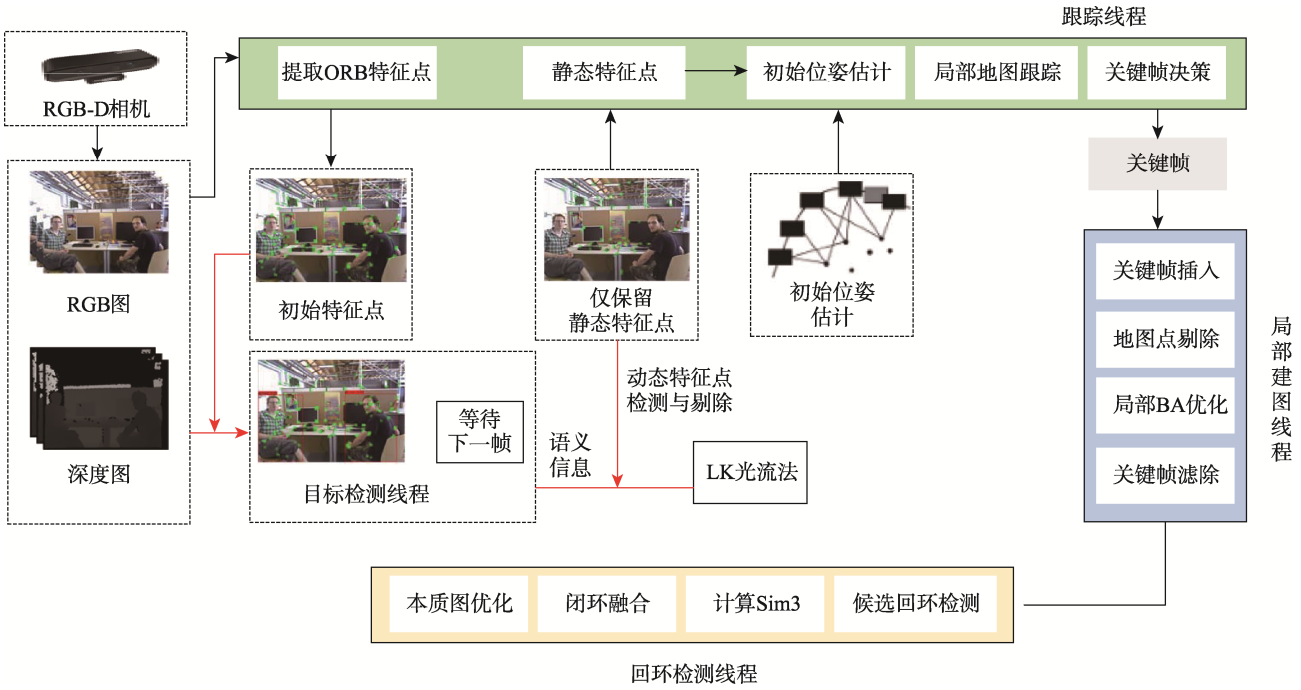


图 1　改进后的系统框架
Fig.1 Improved system frame

1.2　轻量级目标检测网络 YOLOv5s-MobileNetV3

YOLOv5 是当前较为先进的单阶段目标检测算法之一。相较于 YOLOv4 拥有更好的增强方式，相较于当前最新的 YOLOv7 和 YOLOv8，训练和推理速度也要快很多，其具有较低的内存占用和泛化能力强的优势。这使得 YOLOv5 在移动设备或者资源受限的系统中更具优势。因此，选择 YOLOv5 作为本文的目标检测网络是合理的选择。YOLOv5s 的模型参数量约为 7.5 M，是 YOLOv5 系列当中模型最小、运行速度最快的网络，在 NVIDIA V100 GPU 上，平均每张图片的推理时间仅为 0.002 s，满足实时处理的标准[17]。

MobileNet 神经网络结合了逐深度卷积（Depthwise，DW）和逐点卷积（Pointwise，PW），形成了深度可分离卷积（Depthwise Separable Convolution，DSC），使能够构建并训练小型网络模型。相较于常规卷积，它在确保准确性的前提下，运算时间缩短约 11%，模型参数也减少约 14%。MobileNetV3[18]由 Google 在 2019 年提出，与 V1 和 V2 相比，其在精度、推理速度、模型结构等方面上都得到有效提升，网络结构如图 2 所示。考虑到本文更倾向于轻量化卷积网络模型，因此将 YOLOv5s 的 Backbone 用 MobileNetV3-Small 替代，并调整了各层的特征图使其对齐。YOLOv5s 的 Neck 需要 3 种尺度的特征图，它们分别来自 3-P3/8、8-P4/16 和 11-P5/32，如图 3 所示。

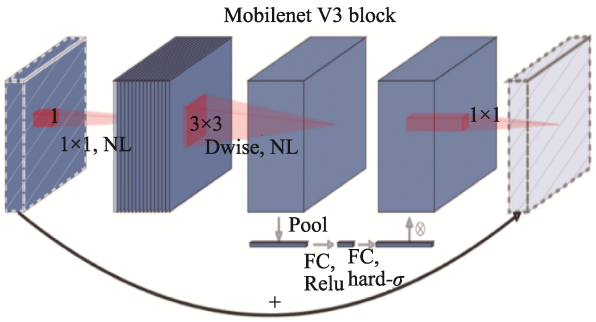


图 2　MobileNetV3 的网络结构
Fig.2 Network structure of MobileNetV3

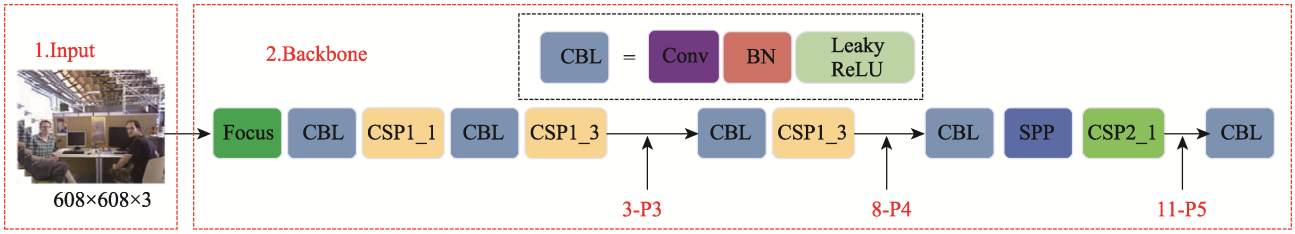


图 3　对 Backbone 的改进
Fig.3 Improvements to Backbone

1.3　动态特征点剔除策略

1.3.1　先验动态区域特征点剔除

在经过目标检测线程确定动态特征点后，先剔除动态物体范围内的特征点，具体剔除过程：假设跟踪线程中提取到第 K 帧图像的特征点集合 A_K，如式（1）所示。

$$A_K=\{p_1, p_2, p_3, \cdots, p_i\} \tag{1}$$

然后基于先验知识确定动态特征点的集合 D_K，如式（2）所示。

$$D_K=\{d_1, d_2, d_3, \cdots, d_i\} \tag{2}$$

如果满足以式（3）条件，则将 P_i 标记为动态特征点，并将其从集合 A_K 中剔除，剩下的特征点视为准静态特征点，它们集合记为 S_K，如式（4）所示。

$$p_i \in D_K(i=1, 2, 3, \ldots, n) \tag{3}$$

$$D_K \cup S_K = A_K \tag{4}$$

实际环境中还可能存在以下特殊情况：除了提前视为动态物体的“人、猫、狗”，还可能忽略“随手拿起翻阅的书本”和“使用电脑需要移动的鼠标”等潜在动态物体，只有在其他外在因素作用下才会被视为动态物体；现实中的动态物体往往是不规则的形状，而目标检测网络的预测框则是规则的矩形区域，这样会使动态物体周围原本的静态特征点被暴力剔除，甚至导致后续特征点匹配时出现跟踪丢失。因此，还需要结合对极几何约束和 LK 光流法再次进行剔除，在剔除动态特征点的前提下，尽可能保留较多的静态特征点。

1.3.2　对极几何约束

在基于目标检测线程剔除动态特征点后，需要对当前帧和参考帧的准静态特征点进行特征匹配，以计算两特征点之间的汉明距离得到匹配点对，取升序排列前 1/3 的点对通过 RANSAC 算法求解出基础矩阵 $\boldsymbol{F}$。具体计算过程如下：如图 4 所示，K_1 和 K_2 为来自同一相机的连续 2 帧图像，p_1 和 p_2 则是一对特征匹配点，O_1 和 O_2 为是相机的中心，p 点是 O_1p_1 和 O_2p_2 的交点，p_1e_1 和 p_2e_2 则是各帧极线，极线求解过程如式（5）~（7）所示。

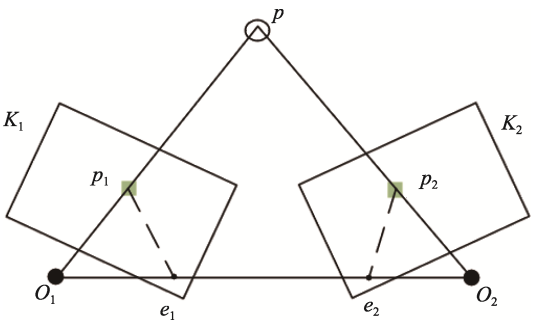


图 4　对极几何约束示意图
Fig.4 Schematic diagram of epipolar geometric constraints

$$p_1 = F[x_1, y_1, 1]^{\mathrm{T}} \tag{5}$$

$$p_2 = F[x_2, y_2, 1]^{\mathrm{T}} \tag{6}$$

$$p_2e_2 = \begin{bmatrix}\boldsymbol{A}\\ \boldsymbol{B}\\ \boldsymbol{C}\end{bmatrix} = Fp_1 = F\begin{bmatrix}x_1\\ y_1\\ 1\end{bmatrix} \tag{7}$$

式中：$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 为线向量。点 p_2 与对应极线之间的距离 d 如式（8）所示。

$$d = \frac{(p_2)^{\mathrm{T}} Fp_1}{\sqrt{\boldsymbol{A}^2+\boldsymbol{B}^2+\boldsymbol{C}^2}} \tag{8}$$

将准静态特征点到极线的距离 d 与设定好的阈值范围 d_ε 比较。如果超出阈值范围，则被视为动态特征点被剔除，本文实验部分 d_ε 取[0.3, 0.7]。若出现物体的运动方向与相机平行，也能满足对极几何约束的条件，因此本文选择 LK 光流法对动态特征点做进一步剔除。

1.3.3　LK 光流法

光流法基于图像序列中像素在时间上的变化和相邻帧的关系来分析前后帧间的匹配，从而获取物体的运动数据。这里选择计算量较稠密光流少的 LK 光流法，如图 5 所示。该方法基于 3 个假设：（1）灰度不变，图像中相同位置的像素灰度在每帧图像中不改变；（2）小运动，相邻帧之间的位移是较小的；（3）空间一致，相邻像素具有相似的运动[19]。从假设（1）可知在灰度不变时，有：

$$I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) = I(x, y, t) \tag{9}$$

式中：t 和 $t+\mathrm{d}t$ 为相邻帧的对应时间；$I(x, y, t)$ 和 $I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t)$是像素点在相邻帧中的位置。从假设（2）对式（9）左侧进行泰勒级数展开，保留一阶项，得到：

$$I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) \approx I(x, y, t)+\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial t}\mathrm{d}t \tag{10}$$

将式（9）和（10）相加，两边同时除以 $\mathrm{d}t$，得到：

$$\frac{\partial I}{\partial x}\mathrm{d}x+\frac{\partial I}{\partial y}\mathrm{d}y+\frac{\partial I}{\partial t}\mathrm{d}t = 0 \tag{11}$$

式中：$\mathrm{d}x/\mathrm{d}t$ 和 $\mathrm{d}y/\mathrm{d}t$ 分别为像素在 x 和 y 轴上的运动速度，记为 u、v。同时$\partial I/\partial x$ 和$\partial I/\partial y$ 则是图像在该点 x 和 y 方向的梯度，记为 I_x、I_y。图像灰度随时间的变化量记为 I_t，有：

$$\begin{bmatrix}I_x I_y\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix} = -I_t \tag{12}$$

由于计算的是像素的运动，所以引入额外的约束来计算 u、v。根据假设（3），选择 3×3 窗口内 9 个具有同样的运动的像素点，有：

$$\boldsymbol{AV} = \boldsymbol{b} \tag{13}$$

其中，$\boldsymbol{A}=\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{x9} & I_{y9} \end{bmatrix}$，$\boldsymbol{V}=\begin{bmatrix} V_x \\ V_y \end{bmatrix}$，$\boldsymbol{b}=\begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{t9} \end{bmatrix}$，用最小二乘法求解的结果如下：

$$\boldsymbol{V}=\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \tag{14}$$

通过 LK 光流法求得剩余特征点的光流大小后，计算其平均值与标准差，通过式（16）~（17）判断特征点是否为动态点。

$$\left|L_i-L_{\mathrm{avg}}\right|>2L_{\mathrm{std}} \tag{16}$$

或

$$\left|L_i-L_{\mathrm{avg}}\right|>L_{\mathrm{thr1}}\left(L_{\mathrm{std}}<L_{\mathrm{thr2}}\right) \tag{17}$$

式中：L_i 为第 i 个特征点的光流大小；L_{avg}、L_{std} 分别为各特征点光流大小的平均值和标准差；L_{thr1} 和 L_{thr2} 为设定的阈值。当 L_i 满足上述关系时，则认为是动态特征点并剔除。

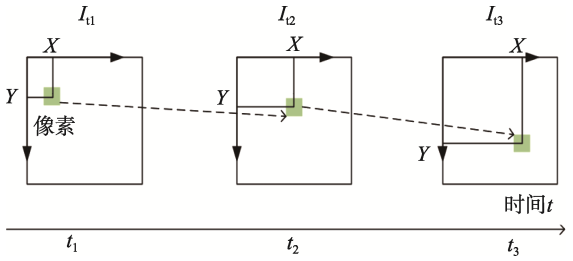


图 5　LK 光流法示意图
Fig.5 Schematic diagram of LK optical flow method

2　实验结果与分析

2.1　轻量化目标检测算法实验

本实验选取 COCO 数据集对改进后的轻量化目标检测网络 YOLOv5s 进行验证。COCO 数据集涵盖了 80 个目标类别，考虑到本文算法应用场景主要是室内动态环境，因此针对性地将“人、猫、狗”类别的数据集进行实验验证，其余类别暂定是静态对象。经过训练后，轻量化的目标检测网络 YOLOv5s-MobileNetV3 与其余的 YOLOv5 不同版本在 CPU 上的验证结果如表 1 所示。其中，$A_{\mathrm{mAP@0.5}}$ 指在阈值 0.5 处设置的 IOU（交并比）的平均精度均值（Mean Average Precision, mAP）。因为在目标检测中，当预测框与真实框的 IOU 值大于 0.5 时，通常认为检测结果是正确的，mAP 值越高，说明模型的性能越好。

从表 1 可以看出，与原有的 YOLO5s 算法对比，在替换原有的主干网络后，算力要求降低至 6.2GFLPOS，模型参数量降低至 3.8M。虽然 MAP 降低了 2.3%，但是检测速度提高了 27 帧/s。在牺牲了一小部分精度的同时，提高了运行速度。因此，可以满足轻量化目标检测算法在资源受限的设备上对精度和实时性的要求。

2.2　TUM 数据集

本文采用公开数据集 TUM RGB-D 对改进后 ORB-SLAM2 算法做整体性能评估。它包含在室内场景中采集到的 RGB 图像、深度图像及位姿真实值等内容，常被学者用来做 SLAM 算法的评估。选取 TUM 数据集中 5 个不同序列进行测试，分别是 fr3_walking_xyz、fr3_walking_static、fr3_walking_rpy、fr3_walking_halfsphere 和 fr3_sitting_static。其中，前 4 组的 fr3_walking 属于高动态场景，表示 2 个人在桌子周围进行走动并伴随着坐立之间的姿态转换；第 5 组的 fr3_ sitting 表示 2 个人坐在桌子前，桌上的物品和肢体会有轻微移动。xyz、static、rpy、halfsphere 则代表着相机做不同方向的运动。

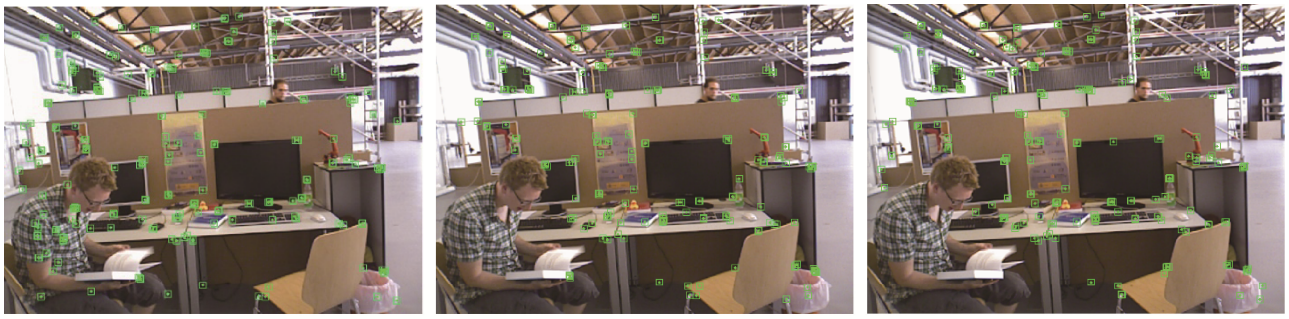
2.3　动态特征点剔除效果

图 6 中图像来源于 fr3_walking_halfsphere 高动态场景序列，此时桌前的人在拿着书本做小范围运动，人和书本均是运动状态。图 6a 是 ORB-SLAM2 算法对图像提取特征点的结果，可以看出此时静态和动态区域的特征点都被提取。图 6b 是仅通过目标检测算法得到先验语义信息，并剔除动态特征点的结果，可以看出人的动态特征点基本被剔除，但书本属于潜在的动态物体而其特征点并未被完全剔除。图 6c 是基于目标检测网络、对极几何约束和光流法共同作用的结果，可以看出此时书本上的剩余动态特征点也已经被完全剔除，且人周围原本的静态特征点并未被暴力剔除，剔除动态特征点的同时保留了更多的静态特征点，用于后续初始位姿估计。

表 1　轻量化目标检测算法在 CPU 上验证
Tab.1 Lightweight object detection algorithm verified on the CPU

网络模型	算力要求/GFLPOS	模型参数量/10^6	帧率/（帧·s^{-1}）	$A_{\mathrm{mAP@0.5}}$/%
YOLOv5s-MobileNetV3	6.2	3.8	106	66.4
YOLOv5s	17.4	7.4	79	68.7
YOLOv5m	49.2	21.1	52	72.4
YOLOv5l	109.6	47.5	33	75.6
YOLOv5x	206.3	89.1	20	77.1

注：一个 GFLOPS（gigaFLOPS）等于每秒 10 亿次的浮点运算。



a ORG-SLAM2　　b Ours(1)　　c Ours(2)

图 6　动态特征点剔除效果对比
Fig.6 Comparison of dynamic feature point elimination effect

2.4　改进后 ORB-SLAM2 算法性能评估

2.4.1　误差结果对比

选取绝对轨迹误差（Absolute Trajectory Error，ATE）以及相对位姿误差（Relative PoseError，RPE）为算法性能评价指标。ATE 是相机位姿的估计值和真实值的直接差值，能直观地反映出算法精度和轨迹全局一致性。RPE 是计算 2 个相同时间戳上相机位姿的估计值和真实值之间的差异，反映出连续位姿上的累积误差信息，包括平移和旋转两部分[20]。其中，以均方根误差 RMSE、平均值 Mean 以及标准差 STD 这 3 个参数对 ATE 和 RPE 这 2 个指标进行统计，RMSE 以及 STD 能够反映系统的鲁棒性，Mean 则能够反映位姿估计的精度。为了直观地看出改进后的算法性能提升，在同一组数据集序列将 ORB-SLAM2 与本文算法做对比。这里将提升效率（Improvements）定义为 I，其公式见式（18）。

$$I=\frac{B-A}{B}\times 100\% \tag{18}$$

式中：A 为改进之后（After）的算法，即本文算法结果；B 为改进之前（Before）的算法，即 ORB-SLAM2 算法结果。表 2、表 3 和表 4 分别是改进前后算法的 ATE、RPE（平移部分）和 RPE（旋转部分）的运算结果。

表 2　ATE 实验结果对比
Tab.2 Comparison of ATE experimental results

序列	ORB-SLAM2			Ours			I/%		
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD
Walking_xyz	0.651 3	0.587 6	0.281 1	0.018 3	0.015 6	0.009 6	97.17	97.34	96.58
Walking_ static	0.385 3	0.349 4	0.162 5	0.008 5	0.007 4	0.004 3	97.79	97.88	97.35
Walking_rpy	0.823 6	0.720 6	0.398 8	0.028 1	0.022 4	0.016 9	96.58	96.89	95.76
Walking_half	0.349 6	0.293 6	0.189 8	0.028 3	0.023 5	0.015 6	91.90	91.99	91.78
Sitting_static	0.008 2	0.007 1	0.004 0	0.005 6	0.004 9	0.002 6	31.70	30.98	35.00

表 3　RPE 平移部分实验结果对比
Tab.3 Comparison of experimental results of RPE translation part

序列	ORB-SLAM2			Ours			I/%		
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD
Walking_xyz	0.435 1	0.309 1	0.306 3	0.022 1	0.019 4	0.010 6	94.92	93.72	96.53
Walking_ static	0.200 6	0.087 1	0.180 7	0.011 2	0.009 0	0.006 6	94.41	89.66	96.34
Walking_rpy	0.392 8	0.270 9	0.284 4	0.042 3	0.034 9	0.023 9	89.23	87.11	91.59
Walking_half	0.292 5	0.171 9	0.236 7	0.032 4	0.027 0	0.017 7	88.92	84.29	92.52
Sitting_static	0.009 7	0.008 5	0.004 6	0.006 8	0.006 0	0.003 2	29.89	29.41	30.43

表 4 RPE 旋转部分实验结果对比

Tab.4 Comparison of experimental results of RPE rotation part

序列	ORB-SLAM2			Ours			*I*/%		
	RMSE	Mean	STD	RMSE	Mean	STD	RMSE	Mean	STD
Walking_xyz	8.437 3	6.137 9	5.789 2	0.630 5	0.486 5	0.401 0	92.52	92.07	93.07
Walking_ static	3.553 3	1.602 4	3.171 5	0.276 9	0.240 7	0.136 8	92.20	84.97	95.68
Walking_rpy	7.585 1	5.266 1	5.506 9	0.910 8	0.759 8	0.502 2	87.99	85.43	90.88
Walking_half	5.946 6	3.625 0	4.714 0	0.776 9	0.682 0	0.371 9	86.93	81.18	92.11
Sitting_static	0.288 7	0.261 3	0.122 9	0.262 2	0.235 2	0.115 8	9.17	9.98	5.77

从表 2 的对比结果可知，在 ATE 方面，4 组高动态序列的 RMSE、Mean 以及 STD 相较于改进之前的 ORB-SLAM2 算法均有大幅度的提升，误差可减小 90%以上。尤其是在 Walking_ static 序列，RMSE、Mean 和 STD 最高分别提升了 97.79%、97.88%和 97.35%。实验结果表明，在高动态场景下的本文算法可以显著提高视觉 SLAM 系统的定位精度和鲁棒性。在低动态序列 Sitting_static 中，各项指标虽也有一定的提升，但相较于高动态序列的提升并不明显。这是由于该序列中的动态物体较少，且原 ORB-SLAM2 算法在低动态场景下本身就有着良好的表现。从表 3 和表 4 的对比结果可以看出，在 RPE 方面的结论与上述类似：本文算法在高动态序列中提升效果明显，在低动态序列中有一定提升但并不明显。

图 7 和图 8 分别是 ORB-SLAM2 与本文算法 ATE 对比和 RPE 对比，图 7a~c 是 ORB-SLAM2 的结果，图 7d~f 是本文算法。图 7 中，黑色代表真实的相机轨迹，蓝色是预测估计的路径。红线则表示两者的误差，线越短，精确度越高。图 8 中，误差波动越小则代表系统越稳定。显然，在室内动态场景下，本文算法在定位精度和鲁棒性上都超越了原有 ORB-SLAM2 算法。

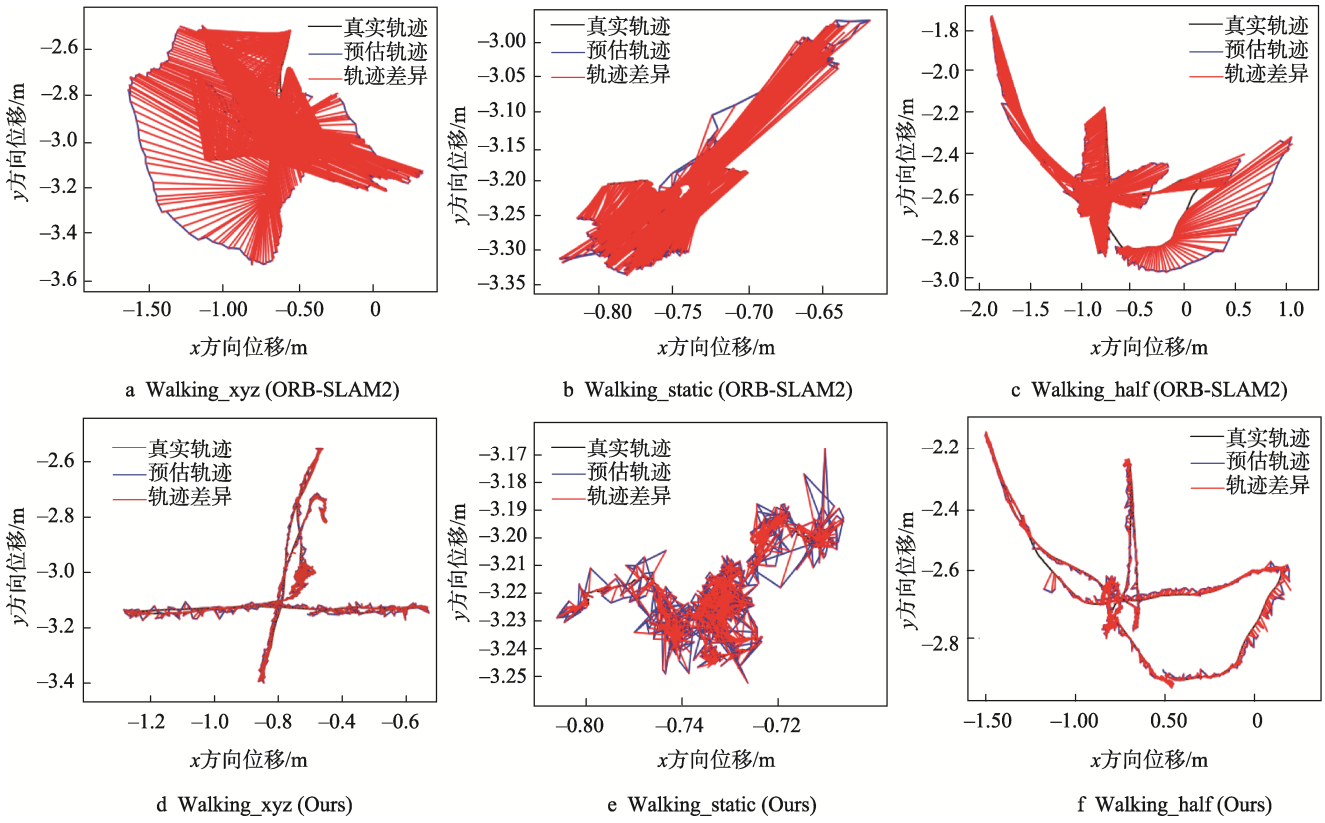


图 7 ORB-SLAM2 与本文算法的 ATE 对比

Fig.7 ATE comparison graph between ORB-SLAM2 and the algorithm proposed in the work

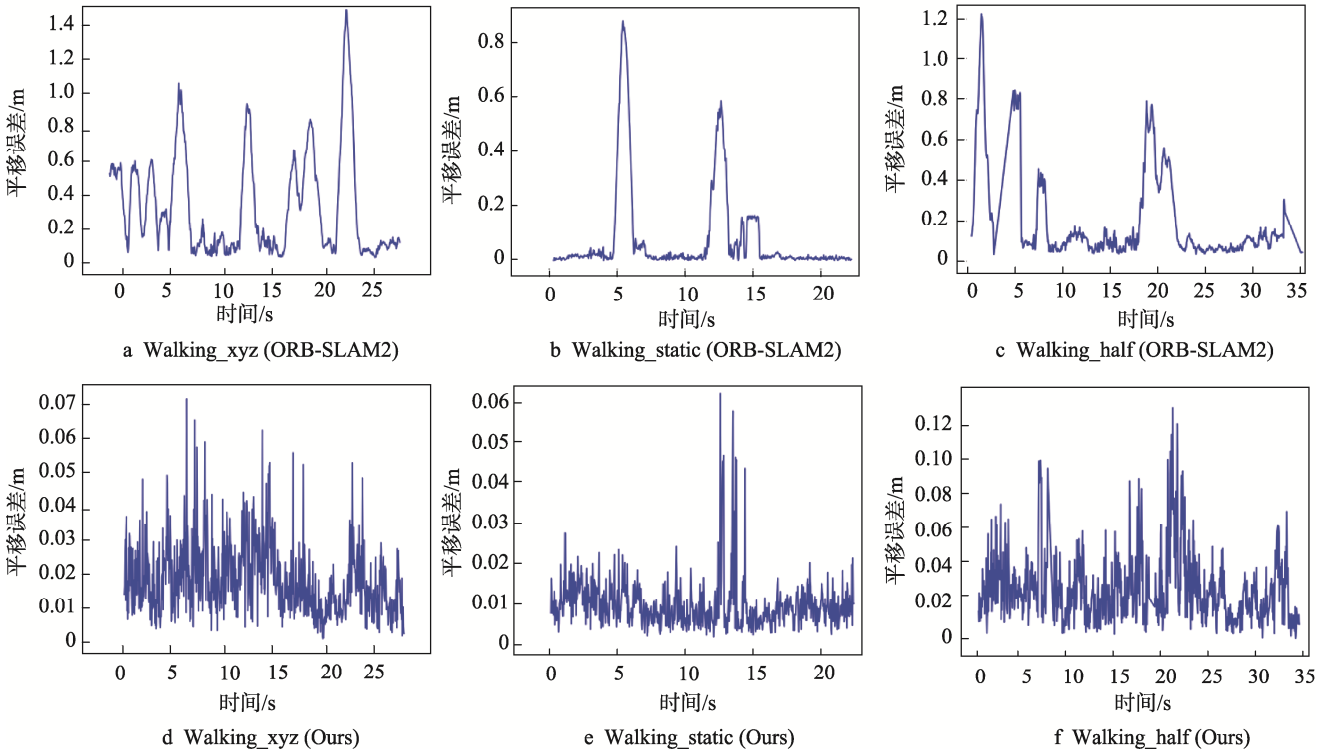


图 8　ORB-SLAM2 与本文算法的 RPE 对比

Fig.8 RPE comparison between ORB-SLAM2 and the algorithm proposed in the work

2.4.2　与同类型其他算法对比

为了进一步验证改进算法的性能，将其与其他同类型的优秀算法进行比较。其中，DS-SLAM 和 DynaSLAM 是在 ORB-SLAM2 的基础上改进，通过语义分割网络提取动态场景的语义信息；DVO-SLAM 是利用相邻图像之间的像素光度误差来进行相机运动估计和场景重建。同类型算法绝对轨迹误差对比结果如表 5 所示。从 5 组动态场景序列的实验可知，在以下列出的算法中，Dyna-SLAM、DS-SLAM 和本文算法定位精度较高。本文算法在精度上取得至少次优的结果，有着良好的表现。

表 6 列出了 3 种算法处理一帧图片所需的时间。其中，DS-SLAM 和 DynaSLAM 采用以逐像素的方式对动态对象进行检测比较耗时，而本文采用轻量级目标检测网络，在运行速度上超越了上述的 2 种算法。因此，本文算法在保证定位精度和鲁棒性的同时，也具有较好的实时性。

表 5　同类型算法 ATE 实验结果对比

Tab.5 Comparison of ATE experimental results of similar algorithms

序列	ORB-SLAM2	DVO-SLAM	Detect-SLAM	DS-SLAM	Dyna-SLAM	Ours
Walking_xyz	0.651 3	0.596 6	0.021 8	0.024 7	**0.015 0**	0.018 3
Walking_ static	0.385 3	—	—	**0.008 1**	0.009 0	0.008 5
Walking_rpy	0.823 6	0.730 4	0.076 7	0.444 2	0.040 0	**0.028 1**
Walking_half	0.349 6	0.528 7	0.052 2	0.030 3	**0.025 0**	0.028 3
Sitting_static	0.008 2	0.050 5	—	0.006 4	0.006 5	**0.005 6**

注：“—”是指原论文中并未提供该数据，粗体表示最优结果，而次优的结果则用下划线标注。

表 6　跟踪时间对比

Tab.6 Comparison of tracking time

算法	语义信息获取	运行环境	单帧跟踪时间/ ms
ORB-SLAM2	—	Intel i7 CPU, NVIDIA RTX 3060 GPU	21.03
DS-SLAM	SegNet	Intel i7 CPU,P4000 GPU	59.40
Dyna-SLAM	Mask R-CNN	Nvidia Tesla M40 GPU	700
Ours	YOLOv5s-MobileNetV3	Intel i7 CPU, NVIDIA RTX 3060 GPU	28.26

3 结语

本文提出了一种面向室内动态场景的视觉SLAM算法，在原有ORB-SLAM2的视觉里程计上，添加了目标检测线程和剔除动态特征点模块。融合YOLOv5s 轻量级目标检测网络实时检测动态物体，在提取 ORB 特征点的同时获取语义信息并剔除先验的动态特征点。考虑到存在潜在动态物体和漏检的情况，结合光流法和对极几何约束对可能残存的动态特征点进一步剔除。实验结果表明，与 ORB-SLAM2 相比，改进后的算法在高动态序列下能够大幅提升SLAM系统的定位精度和鲁棒性，跟踪线程中处理一帧图像平均只需 28.26 ms，与其他同类型优秀算法相比，本文算法在定位精度和实时性方面都具有一定的优势，有望未来在包装行业扮演更加重要的角色。但是，还存在一些需要改进的地方，例如，需要提高算法的实时性，或者构建质量较高语义八叉树地图，以便于能够执行导航等更高级的任务。

参考文献：

[1] 王柯赛，姚锡凡，黄宇，等. 动态环境下的视觉SLAM研究评述[J]. 机器人, 2021, 43(6): 715-732.
WANG K S, YAO X F, HUANG Y, et al. Review of Visual SLAM in Dynamic Environment[J]. Robot, 2021, 43(6): 715-732.

[2] KLEIN G, MURRAY D. Parallel Tracking and Mapping for Small AR Workspaces[C]// 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE, 2007: 225-234.

[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] ENGEL J, KOLTUN V, CREMERS D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(3): 611-625.

[5] YANG N, STUMBERG L, WANG R, et al. D3vo: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 1281-1292.

[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] 张有全，祁宇明，邓三鹏，等. 直接法和共视图优化的视觉惯性 SLAM 系统研究[J]. 自动化与仪器仪表, 2022, 271(5): 197-203.
ZHANG Y Q, QI Y M, DENG S P, et al. Research on Visual-Inertial SLAM System Based on Direct Method and Common View Optimization[J]. Automation and Instrumentation, 2022, 271(5): 197-203.

[8] 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.
ZHU D Y, ZHONG Y, YANG G C, et al. Research Progress on Motion Segmentation of Visual Localization and Mapping in Dynamic Environment[J]. Journal of Computer Applications, 2023, 43(8): 2537-2545.

[9] ZHONG F, WANG S, ZHANG Z, et al. Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial[C]// 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018: 1001-1010.

[10] LIU W, ANGUELOV D, ERHAN D. SSD: Single Shot MultiBox Detector[C]// European Conference on Computer Vision, 2016: 21-37.

[11] YU C, LIU Z, LIU X, et al. DS-SLAM: a Semantic Visual SLAM Towards Dynamic Environments[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Piscataway, USA: IEEE, 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] 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.

[14] JOHNSON J W. Adapting Mask-Rcnn for Automatic Nucleus Segmentation[J]. arXiv e-prints, 2018, 3: 1-7.

[15] YAN L, HU X, ZHAO L, et al. DGS-SLAM: A Fast and Robust RGBD SLAM in Dynamic Environments Com-

bined by Geometric and Semantic Information[J]. Remote Sensing, 2022, 14(3): 795-819.

[16] 李博, 段中兴. 室内动态环境下基于深度学习的视觉里程计[J]. 小型微型计算机系统, 2023, 44(01): 49-55.
LI B, DUAN Z X. Visual Odometer Based on Deep Learning in Dynamic Indoor Environment[J]. Journal of Chinese Computer Systems, 2023, 44(1): 49-55.

[17] 伍子嘉, 陈航, 彭勇, 等. 动态环境下融合轻量级 YOLOv5s 的视觉 SLAM[J]. 计算机工程, 2022, 48(08): 187-195.
WU Z J, CHEN H, PENG Y, et al. Visual SLAM with Lightweight YOLOv5s in Dynamic Environment[J]. Computer Engineering, 2022, 48(8): 187-195.

[18] HOWARD A, SANDLER M, CHU G, et al. Searching for Mobilenetv3[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 1314-1324.

[19] LI G, YU L, FEI S. A Binocular MSCKF-Based Visual Inertial Odometry System Using LK Optical Flow[J]. Journal of Intelligent & Robotic Systems, 2020, 100(3): 1179-1194.

[20] 张恒, 徐长春, 刘艳丽, 等. 基于语义分割动态特征点剔除的 SLAM 算法[J]. 计算机应用研究, 2022, 39(5): 1472-1477.
ZHANG H, XU C C, LIU Y L, et al. SLAM Algorithm Based on Semantic Segmentation and Dynamic Feature Point Elimination[J]. Application Research of Computers, 2022, 39(5): 1472-1477.