引用格式: 张禹, 高新. 动态场景下基于实例分割与光流的语义 SLAM 建图[J]. 微电子学与计算机, 2024, 41(2): 19-27.
ZHANG Y, GAO X. Semantic SLAM building based on instance segmentation and optical flow in dynamic scenes[J]. Microelectronics & Computer, 2024, 41(2): 19-27.
DOI: 10.19304/J.ISSN1000-7180.2023.0033

动态场景下基于实例分割与光流的语义 SLAM 建图

张 禹, 高 新
(沈阳工业大学 机械工程学院, 辽宁 沈阳 110870)

摘 要: 视觉同步定位与建图技术常用于室内智能机器人的导航, 但是其位姿是以静态环境为前提进行估计的。为了提升视觉即时定位与建图(Simultaneous Localization And Mapping, SLAM)在动态场景中的定位与建图的鲁棒性和实时性, 在原 ORB-SLAM2 基础上新增动态区域检测线程和语义点云线程。动态区域检测线程由实例分割网络和光流估计网络组成, 实例分割赋予动态场景语义信息的同时生成先验性动态物体的掩膜。为了解决实例分割网络的欠分割问题, 采用轻量级光流估计网络辅助检测动态区域, 生成准确性更高的动态区域掩膜。将生成的动态区域掩膜传入到跟踪线程中进行实时剔除动态区域特征点, 然后使用地图中剩余的静态特征点进行相机的位姿估计并建立语义点云地图。在公开 TUM 数据集上的实验结果表明, 改进后的 SLAM 系统在保证实时性的前提下, 提升了其在动态场景中的定位与建图的鲁棒性。

关键词: 即时定位与建图; 动态场景; 实例分割; 光流估计
中图分类号: TP391 **文献标识码:** A **文章编号:** 1000-7180(2024)02-0019-09

Semantic SLAM building based on instance segmentation and optical flow in dynamic scenes

ZHANG Yu, GAO Xin
(College of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China)

Abstract: The visual simultaneous localization and mapping technique is commonly used for indoor intelligent robot navigation, but its poses are estimated with static environment in mind. In order to improve the robustness and real-time performance of visual Simultaneous Localization And Mapping(SLAM) for localization and mapping in dynamic scenes, we add dynamic region detection threads and semantic point cloud threads to the original ORB-SLAM2. The dynamic region detection thread consists of the instance segmentation network and the optical flow estimation network. The instance segmentation gives semantic information to the dynamic scene while generating a priori dynamic object masks, and in order to solve the under-segmentation problem of the instance segmentation network, the lightweight optical flow estimation network is used to assist the detection of dynamic regions and generate dynamic region masks with higher accuracy. The generated dynamic region masks are passed into the tracking thread for real-time rejection of dynamic region feature points, and then the remaining static feature points in the map are used for the camera′s positional estimation and to build a semantic point cloud map. Experimental results on the publicly available TUM dataset show that the improved SLAM system improves the robustness of its localization and map building in dynamic scenes while ensuring real-time performance.

Key words: simultaneous localization and mapping(SLAM); dynamic scenes; instance segmentation; optical flow estimation

收稿日期: 2023-03-10; **修回日期:** 2023-03-15

1 引言

视觉定位与地图构建系统可以应用到许多机器人中，以相机为视觉传感器的机器人可以在一个未知环境中通过传感器采集的信息估计当前机器人的位姿并同时构建当前环境的增量式地图。该系统在运行过程中大都是以静态环境为前提，但是现实环境中存在动态物体如路过的行人、行进的汽车等。如果这种动态物体在场景中较多，会严重影响系统的定位精度和鲁棒性。

对于静态环境而言，视觉定位系统研究的相对成熟，如单目实时视觉即时定位与建图(Real-time Single Camera Simultaneous Localization And Mapping, Mono-SLAM)[1]，并行跟踪与地图构建(Parallel Tracking And Mapping, PTAM)[2]，实时跟踪与稠密地图构建(Dense Tracking And Mapping in Real-time, DTAM)[3]，大规模直接 SLAM(Large-Scale Direct SLAM, LSD-SLAM)[4]，快速半直接单目视觉里程计(Fast Semi-direct-monocular Visual Odometry, SVO)[5]，直接稀疏里程计(Direct Sparse Odometry, DSO)[6]，单目、双目和 RGB-D 相机的开源 SLAM 系统(Open-source SLAM System for Monocular, Stereo, and RGB-D Cameras, ORB-SLAM2)[7]系列，其中 ORB-SLAM2被认为是比较完备的系统，其相机的位姿根据静态环境中所提取的特征点直接计算得到。但其在动态环境中的定位精度会受环境中动态干扰对象的影响，从而出现长时间的位姿跟踪丢失等问题。

针对以上问题，目前 SLAM 系统处理场景中动态对象的方法主要分为两种：(1)基于多视几何及其相关改进方法。Alcantarilla 等[8]利用连续帧计算相机的位粗略计算出图像的稠密 3D 光流，通过计算两帧图像匹配点的马氏距离与阈值进行比较来剔除外点。Tan 等[9]利用随机抽样一致算法(Random Sample and Consensus, RANSAC)计算相邻帧图像的单位矩阵，然后将矩阵与前一帧图像进行矩阵运算得到变换后的图像，与当前帧图像作差后静态区域的像素值趋于 0，对于动态区域则产生大于 0 的像素值，经过后处理后得到分割后的结果。(2)不依赖相机自身运动的方法。Sheng 等[10]利用 Mask-RCNN(Region-Convolutional Neural Network)[11]对图像进行语义分割得到环境中的高动态区域，将不在动态区域位置上的像素点及其邻域像素点都视为静态，然后将静态区域直接嵌入 DSO 系统可以在 TUM 数据集上得到很好的实验结果。Li 等[12]将深度边缘点进行加权，并利用得到的权重判断一个点是动态点或静态点，从而剔除关键帧中的动态点。Bescos 等[13]利用 Mask-RCNN 语义分割网络和多视图几何算法对动态对象进行检测和剔除，以 ORB-SLAM2 为框架提出 DynaSLAM。Zhong 等[14]利用 SSD[15]网络根据先验信息标记场景中动态物体来过滤动态特征点，然后利用剩余的静态点进行建图。Yu 等[16]利用 SegNet[17]网络结合运动一致性检测剔除动态场景中的动态特征点提出 DS-SLAM。

利用多视图几何等方法在大多数动态场景中鲁棒性较低，而利用先验语义信息的方法，受训练集中的先验性动态物体种类和个数的限制，如果场景中出现训练集动态物体种类以外对象就会降低系统定位精度。对此本文提出一种结合实例分割和光流估计的视觉 SLAM 方法，在对视觉 SLAM 系统运行速度影响不大的前提下提升 SLAM 系统在动态场景中的鲁棒性。

2 系统框架

2.1 视觉 SLAM 结构

为了视觉 SLAM 系统在动态环境中有更好的定位精度并提升其对周围环境的语义理解，本文在 ORB-SLAM2 原有 3 个线程的基础上添加了动态区域检测线程和稠密点云建图线程，改进的视觉 SLAM 的整体框架如图 1 所示。

深度相机采集的图像帧输入到跟踪线程和动态区域检测线程。跟踪线程对传入的图像进行 ORB 特征点提取，通过前一帧图像估计出当前帧的位姿，然后基于当前帧位姿对局部地图进行跟踪。动态区域检测线程由于只使用实例分割网络对动态物体去除的结果存在漏检的情况，所以本文以光流估计网络作为光流预测模块，辅助实例分割模块对动态物体进一步检测，从而减小因为漏检场景中的动态物体而带来的对定位精度的影响。考虑 SLAM 系统的运行速度，选用 YOLACT++作为实例分割网络[18]和 LiteFlowNet2 作为光流估计网络[19]对场景中的动态物体进行处理。分割网络根据先验知识赋予图像中先验性动态物体语义信息，结合光流场数据生成动态区域掩膜，将结果传入到跟踪线程对动态区域内的特征点进行剔除后再参与跟踪局部地图，使用两个线程共同作用后的结果来决定是否生成关键帧。其中跟踪局部地图的作用是对局部地图进行更新，光束法平差(Bundle Adjustment, BA)优化当前帧位

姿和地图点并通过统计的内点个数判断是否跟踪成功。将生成的不含动态物体的关键帧用于建立局部地图和稠密点云地图。局部建图线程对传入的关键帧使用基于投影误差的方法进行外点剔除，用对极几何或者三角化的方法创建新的地图点来补充外点，将当前关键帧与共视关键帧进行地图点融合，进行局部 BA 优化后对冗余关键帧进行剔除。稠密点云构建线程用带有语义信息的关键帧和其对应的深度图像生成局部点云并将其赋予语义信息后剔除动态物体点云。

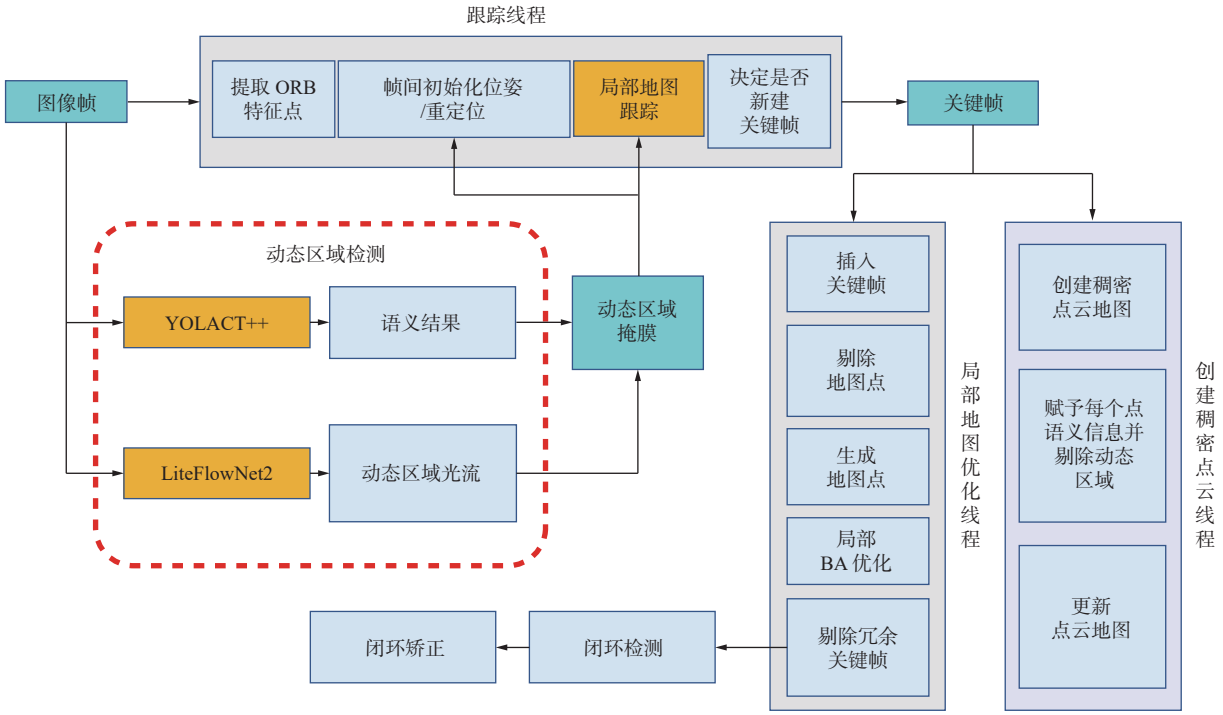


图 1　系统框架

Fig. 1 System framework

2.2　实例分割网络

本文使用 COCO 公开数据集，以 resnet50 为 backbone 对 YOLACT++网络进行训练，其网络结构如图 2 所示。

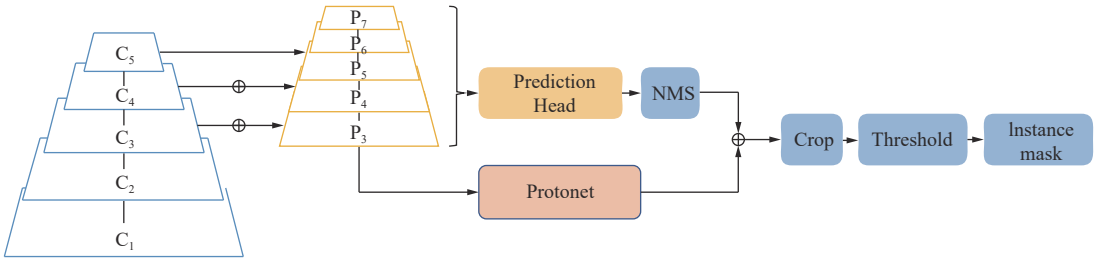


图 2　YOLACT 网络结构图

Fig. 2 YOLACT network structure

YOLACT 在现有的 one-stage 检测模型上增加了掩膜分枝，YOLACT 中并不存在明确的特征定位步骤。其将实例分割任务分为两个相对简单的并行任务，并最终组合起来形成掩膜。Prediction head 分支生成预测的 anchor 和 mask 掩膜，然后通过改进非极大值抑制(Non Maximum Suppression, NMS)更快速地滤除了低置信度的 anchor，并且使用共享卷积的方法达到提速的目的。Protonet 分支生成原型 mask 并与 Prediction head 分支生成的 mask 掩膜进行矩阵相乘运算得到图像中每一个物体的 mask，其中 Crop 是指把 bounding box 边界以外的 mask 清零，然后再进行阈值分割。YOLACT++为了提升精度在原框架的基础上加入了可变形卷积；采用更合理的 anchor 的大小、比例以及分配策略，使得每个 anchor 更容易被分配到正确的目标上；采用更加密集的 anchor，即在同等的输入分辨率下，这有助于

提高模型对小目标的检测和分割能力；减少了 FC 层并引入 压缩-激励 (Squeeze-and-Excitation, SE) 机制，在保证检测精度的同时，提高了计算效率。

2.3 光流估计网络

光流任务即估计两个连续帧之间的逐像素的位移来计算其运动向量。LiteFlowNet2 是一种轻量级的实时光流估计模块，其参数量相对于 FlowNet2[20] 的参数量缩小了 25.3 倍，运行速度比其快了 3.1 倍，在 RTX3070 显卡上的光流估计帧率达到实时动态物体检测的条件。

本文采用 MPI Sintel 的公开数据集对网络进行训练，其结构如图 3 所示，金字塔特征提取(NetC)和光流估计(NetE)的两个轻量级子网络构成了 LiteFlowNet2。NetC 将给定的一对图像分别转换成为两个多尺度高维特征金字塔。其中 $F_k(I)$表示特征金字塔的第 k 层，Tied weights 为双流子网络提供权重共享。NetE 由级联流场推理和光流正则化模块组成，其中级联流场推理模块 M:S 由描述子匹配单元 M 和子像素细化单元 S 组成。先对高阶特征进行逐像素匹配得到粗略的光流估计，再将粗光流进一步细化提高到亚像素精度。

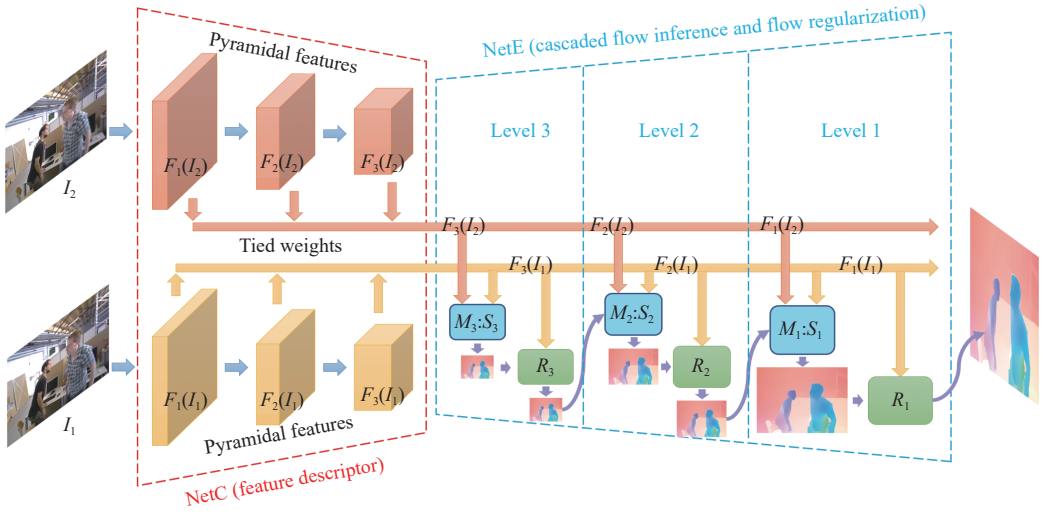


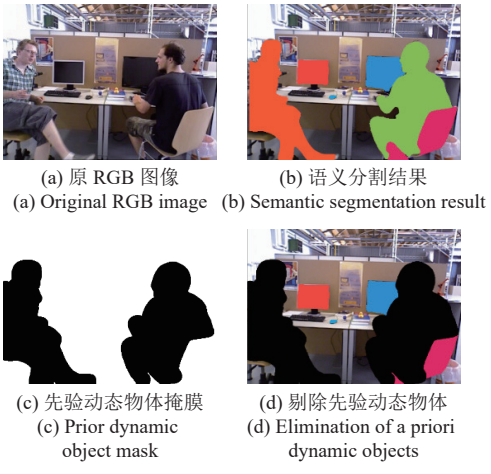
图 3 LiteFlowNet2 网络结构图

Fig. 3 LiteFlowNet2 network structure

3 动态物体的剔除

3.1 基于实例分割的动态物体检测

YOLACT++的高效性的实例分割网络，虽然在精度上相较于 Mask-RCNN 有略微的降低，但是在速度上是 Mask-RCNN 的 6 倍。为了更好地测试系统在未知动态环境中的性能，实例分割网络只使用了 COCO 数据集进行训练，并没有在 TUM 数据集上进行权重参数的调整。为了提升整个系统的运行速度，在动态 SLAM 系统新增 YOLACT++语义分割线程，使跟踪线程和动态区域检测线程同时运行，将接收到的 RGB 图像分别传入跟踪线程和动态区域检测线程，实例网络得到语义分割结果 $Mask_{seg}$ 如图 4(b) 所示，网络将拥有先验语义信息的内容按像素 p_i 分割出来，将处于静态区域 R_{static} 的像素点的值置为 0，将处于动态区域 $R_{dynamic}$ 的像素点的值置为 1。把结果转换为二值掩膜图像 $Mask_t$，考虑到分割网络分割结果轮廓与真实物体之间的误差，将先验性动态物体区域 $Mask_t$ 作 dilate 形态学处理，扩大了 YOLACT++对先验性动态物体轮廓的分割范围，最后将结果 Res_{seg} 输入到跟踪线程中。该过程如式(1)和式(2)所示：



(a) 原 RGB 图像 (a) Original RGB image

(b) 语义分割结果 (b) Semantic segmentation result

(c) 先验动态物体掩膜 (c) Prior dynamic object mask

(d) 剔除先验动态物体 (d) Elimination of a priori dynamic objects

图4 基于 YOLACT++的动态物体检测

Fig. 4 Dynamic object detection based on YOLACT++

$$Mask_{\text{seg}}(p_i)=\begin{cases}0 & p_i\in R_{\text{static}}\\ 1 & p_i\in R_{\text{dynamic}}\end{cases} \tag{1}$$

$$Res_{\text{seg}}=S^{\text{semantic}}\times\text{dilate}(Mask_t) \tag{2}$$

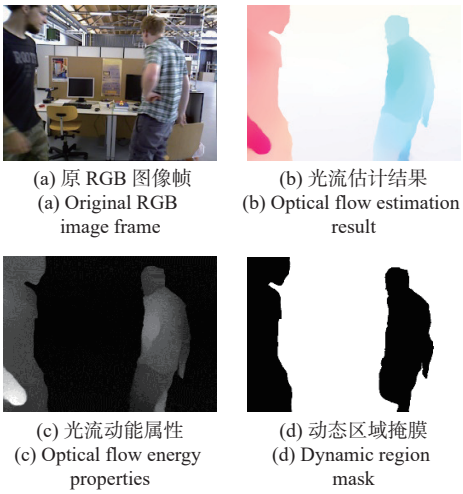
3.2　基于光流估计的动态区域检测

由于视觉 SLAM 系统通常会运行在未知的环境中，实例分割网络虽然可以检测出先验的动态物体，但在某些情况下如没有完全进入画面的人、手中翻动的书和转动的椅子等。在这种情况下实例分割网络会出现欠分割的现象，从而导致生成的动态掩膜不准确。为了能够更精准地提取这些动态区域，并且尽可能降低对 SLAM 系统的运行速度的影响，使用轻量级光流估计网络辅助识别动态区域。LiteFlowNet2 通过 x 方向的光流失量 $\boldsymbol{u}$ 和 y 方向的光流失量 $\boldsymbol{v}$ 动作向量估计出光流大小 e，然后将 e 归一化到 [0, 1] 的值域内得到 $e_{\text{normalization}}$，其可视化如图 5(c) 所示，使用全局阈值选择方法 OTSU 的最佳阈值来划分动态和静态区域后对图像进行二值化处理，将动态区域赋值为 0，静态区域赋值为 1，如图 5(d) 所示。以上过程如式(3)~式(5)所示：

$$e=\sqrt{f_u^2+f_v^2} \tag{3}$$

$$e_{\text{normalization}}=\frac{e-\min}{\max-\min} \tag{4}$$

$$Mask_{\text{flow}}=\text{OTSU}(255\times e_{\text{normalization}}) \tag{5}$$



(a) 原 RGB 图像帧 (a) Original RGB image frame　(b) 光流估计结果 (b) Optical flow estimation result

(c) 光流动能属性 (c) Optical flow energy properties　(d) 动态区域掩膜 (d) Dynamic region mask

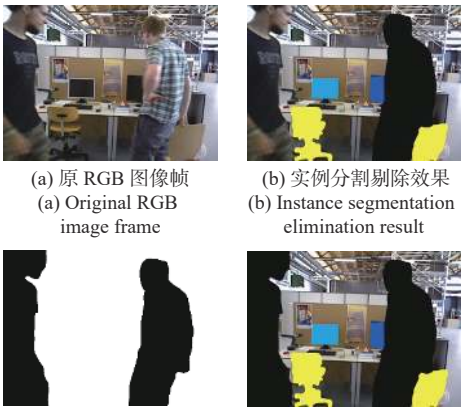
图 5　基于光流估计的动态区域检测

Fig. 5 Dynamic region detection based on optical flow field

将实例分割网络二值掩码 $Mask_{\text{seg}}$ 和光流估计二值掩码 $Mask_{\text{flow}}$ 进行矩阵相乘得到更准确的动态物体二值图，动态区域去除后的二值图如 6(c) 所示。将二值图输入到 SLAM 的跟踪线程中，对所提取到的 ORB 特征点进行动态点过滤，根据特征点的描述子判断特征点是否落在 *Mask* 内，如果特征点所在位置的 *Mask* 的值为 1，则认为该点是静态点参与建图，而值为 0 所对应的特征点被认为是动态点不参与建图，即仅仅保留在 *Mask* 内的关键点，其中 *Mask* 为语义分割和光流估计结合后产生的最终掩膜，如图 6(c) 所示。*Res* 为动态物体剔除后的最终效果，如图 6(d) 所示。以上过程如式(6)和式(7)所示：

$$Mask=Mask_{\text{seg}}\times Mask_{\text{flow}} \tag{6}$$

$$Res=S^{\text{semantic}}\times Mask \tag{7}$$



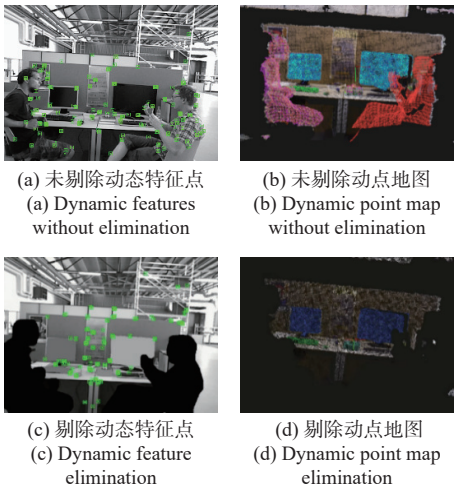
(a) 原 RGB 图像帧 (a) Original RGB image frame　(b) 实例分割剔除效果 (b) Instance segmentation elimination result

(c) 二者结合的动态掩膜 (c) Dynamic mask combining both　(d) 最终剔除效果 (d) Final elimination result

图 6　剔除动态区域

Fig. 6 Removal of dynamic regions

将剔除动态对象后的 RGB 图像以及 Depth 图像输入到点云生成线程中，通过点云库(Point Cloud Library, PCL)在线生成剔除动态物体后的语义点云地图，如图 7 所示。



(a) 未剔除动态特征点 (a) Dynamic features without elimination　(b) 未剔除动点地图 (b) Dynamic point map without elimination

(c) 剔除动态特征点 (c) Dynamic feature elimination　(d) 剔除动点地图 (d) Dynamic point map elimination

图 7　剔除动态对象的语义地图

Fig. 7 Semantic map with dynamic objects removed

4　实验

4.1　实验条件

本文实验平台为 AMD Ryzen 7 5800H with

Rade-on Graphics CPU、内存 16 G(3 200 MHz)、NVIDIA Ge-Force RTX 3070 Laptop GPU、显存 8 GB 的 PC，系统环境为安装到 KESU 移动机械硬盘的 Ubuntu20.04。SLAM 系统使用 C++编写，cmake 工程编译。实例分割和光流估计网络使用 Python 编写，使用 C++调用 Python 接口实现其功能。

实验使用 TUM 中的 rgbd_dataset_freiburg3 动态数据集进行测试，其中包含高动态场景和低动态场景，将其中高动态场景和低动态场景分别简称为 fr3_w(walking)和 fr3_s(sitting)，在高动态 walking 序列中，两人在办公室中处于高动态行走，主要评估视觉 SLAM 在有快速移动物体场景中的鲁棒性。在低动态 sitting 序列中，人坐在办公桌前交谈，并伴有手部细微动作，运动幅度较小，主要评估视觉 SLAM 系统对低动态对象的鲁棒性。其中 hs(halfsphere)、rpy、static 和 *xyz* 表示相机的不同移动方式。其中 hs 表示相机 z 作半球形状运动；rpy 表示相机作左右旋转和俯仰运动；static 表示手持相机保持静止；*xyz* 表示相机分别沿 *x* 轴、*y* 轴和 *z* 轴方向运动。

4.2 实验定位精度

本文采用视觉 SLAM 常用评估指标绝对轨迹误差(Absolute Track Error, ATE)和相对位姿误差(Relative Position Error, RPE)来评估定位精度。绝对轨迹误差是估计出的位姿和真实位姿之间直接的差值，能够很直观地反映出来 SLAM 算法精度和轨迹的一致性。相对位姿误差是相隔固定的时间的两帧位姿差相比于真实值的精度，相当于里程计的误差。表 1 和表 2 分别为本文算法相对于 ORB-SLAM2[7] 的 ATE 和 RPE，表 3 和表 4 分别为本文改进后的算法相较于 ORB-SLAM2 的 ATE 和 RPE 指标的性能提升。表中的精度评价指标分别表示为均方根误差 RMSE、中位数 Median、平均值 Mean 和标准差 STD，结果表明：本文算法的 ATE 和 RPE 误差在大部分情况下小于 ORB-SLAM2 的误差，以 ATE 中的均方根误差 RMSE 作为评估指标在 8 个数据集上的平均定位精度提升 53%，其中高动态场景下平均提升 83.5%。

表 1 TUM 的 ATE

Tab. 1 ATE for TUM

Sequence	ORB-SLAM2				Ours			
	RMSE	Median	Mean	STD	RMSE	Median	Mean	STD
fr3_s_hs	0.026 2	0.017 7	0.021 1	0.015 5	0.023 5	0.018 3	0.020 4	0.011 7
fr3_s_rpy	0.061 2	0.035 0	0.049 0	0.036 7	0.039 2	0.025 6	0.032 8	0.021 4
fr3_s_static	0.013 3	0.012 0	0.012 5	0.004 5	0.007 5	0.006 5	0.006 3	0.004 1
fr3_s_*xyz*	0.012 2	0.009 9	0.010 4	0.006 3	0.012 1	0.010 1	0.010 5	0.006 0
fr3_w_hs	0.297 3	0.208 4	0.241 2	0.173 8	0.025 6	0.013 4	0.018 2	0.014 2
fr3_w_rpy	0.932 7	0.820 6	0.817 1	0.449 8	0.021 7	0.012 6	0.016 9	0.013 6
fr3_w_static	0.013 7	0.006 2	0.008 5	0.010 7	0.007 1	0.006 1	0.006 2	0.003 5
fr3_w_*xyz*	0.386 0	0.349 7	0.346 3	0.170 5	0.012 8	0.009 2	0.011 0	0.006 6

表 2 TUM 的 RPE

Tab. 2 RPE for TUM

Sequence	ORB-SLAM2				Ours			
	RMSE	Median	Mean	STD	RMSE	Median	Mean	STD
fr3_s_hs	0.024 4	0.016 2	0.019 3	0.014 9	0.020 1	0.013 2	0.016 4	0.011 6
fr3_s_rpy	0.028 0	0.020 3	0.023 2	0.015 7	0.033 3	0.014 7	0.023 6	0.023 5
fr3_s_static	0.019 0	0.017 4	0.016 5	0.009 3	0.006 7	0.006 1	0.006 0	0.002 9
fr3_s_*xyz*	0.020 1	0.011 6	0.014 8	0.013 7	0.015 4	0.010 7	0.012 7	0.008 7
fr3_w_hs	0.035 7	0.016 2	0.024 8	0.025 7	0.025 6	0.017 7	0.021 3	0.014 2
fr3_w_rpy	0.045 1	0.025 9	0.034 2	0.029 5	0.021 7	0.012 6	0.016 9	0.013 6
fr3_w_static	0.027 7	0.010 4	0.015 9	0.022 7	0.007 1	0.006 1	0.006 2	0.003 5
fr3_w_*xyz*	0.043 9	0.025 2	0.032 7	0.029 3	0.012 8	0.009 2	0.010 9	0.006 5

表 3　TUM 的 ATE 性能提升

Tab. 3　Improvement of ATE for TUM　　%

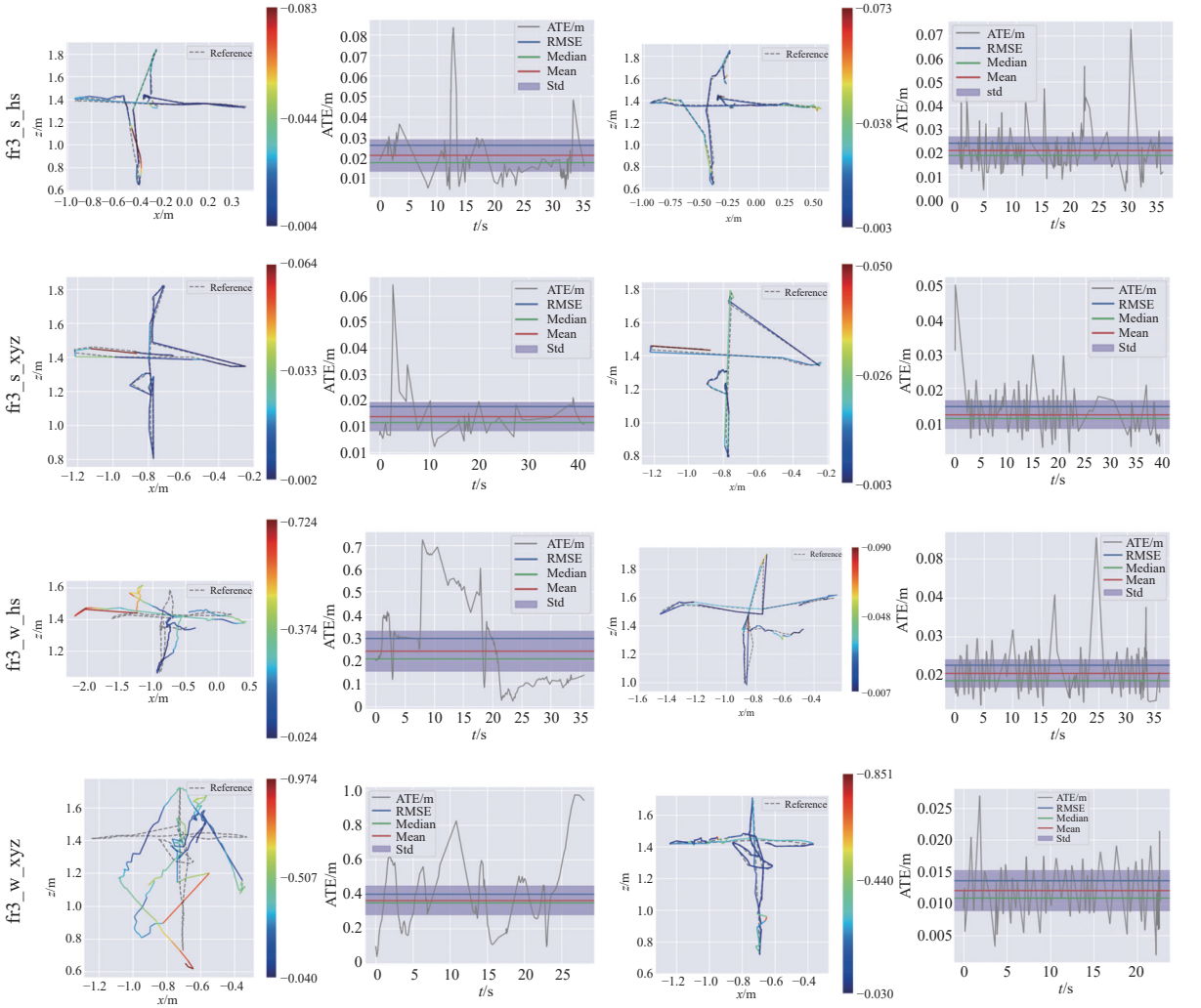
Sequence	RMSE	Median	Mean	STD
fr3_s_hs	10.3	−3.4	3.3	24.5
fr3_s_rpy	35.9	26.9	33.1	41.7
fr3_s_static	43.6	45.8	49.6	8.9
fr3_s_*xyz*	0.8	−2.0	−1.0	4.8
fr3_w_hs	91.4	93.6	92.5	91.8
fr3_w_rpy	97.7	98.5	97.9	97.0
fr3_w_static	48.2	1.6	27.1	67.3
fr3_w_*xyz*	96.7	97.4	96.8	96.1

图 8 为 ORB-SLAM2 算法和本文算法在 TUM 数据集中低动态和高动态场景下的轨迹和误差图比较。从图 8(a) 和图 8(c) 中可以看出，本文改进后的算法的估计轨迹与真实轨迹重合度更高，特别是在高动态场景中尤为明显，其中虚线表示真实的相机运动轨迹，实线表示 SLAM 算法的估计轨迹，右侧色柱条数值表示相机的运动速度。图 8(b) 和图 8(d) 中的单帧误差曲线波动较小，说明改进后的算法精准度和稳定性都较高。

表 4　TUM 的 RPE 性能提升

Tab. 4　Improvement of RPE for TUM　　%

Sequence	RMSE	Median	Mean	STD
fr3_s_hs	17.6	18.5	15.0	22.1
fr3_s_rpy	−18.9	27.6	−1.7	−49.7
fr3_s_static	64.7	64.9	63.6	68.8
fr3_s_*xyz*	23.4	7.8	14.2	36.5
fr3_w_hs	28.3	−9.3	14.1	44.7
fr3_w_rpy	51.9	51.4	50.6	53.9
fr3_w_static	74.4	41.3	61.0	84.6
fr3_w_*xyz*	70.8	63.5	66.7	77.8



(a) ORB-SLAM2 轨迹　(b) ORB-SLAM2 误差　(c) 改进之后的轨迹　(d) 改进之后的误差

(a) ORB-SLAM2 trajectory　(b) ORB-SLAM2 error　(c) The trajectory after improvement　(d) The error after improvement

图 8　TUM 的轨迹与误差比较

Fig. 8 The trajectory and error comparison of TUM

4.3 实验算法运行时间

算法各模块耗时如表 5 所示，对算法运行速度影响较大的实例分割网络和光流估计网络均使用 GPU 计算加速与语义点云线程和跟踪主线程并行运行，跟踪线程和动态特征点剔除部分对算法运行效率影响不大，整体上降低了动态区域检测模块对 SLAM 系统计算速度的影响。

表 5 TUM 的 ATE 性能提升

Tab. 5 Improvement of ATE for TUM

Module	Time/ms
LiteFlowNet2	23.6
YOLACT++	22.9
Tracking	9.7
动态特征点剔除	7.5

DynaSLAM[13] 中的 Mask-RCNN 模块在 GPU 上的运行效率是 5 帧/s，而本文算法在 GPU 上运行速度达到了 16 帧/s，运行效率提升了 68.8%。

5 结束语

整个动态 SLAM 系统建立在 ORB-SLAM2 基础上，新增了实例分割和光流估计网络对动态区域进行检测。为了解决实例分割在某些情况下的欠分割问题，以光流估计网络作为辅助模块进行动态区域检测，根据其光流矢量大小计算出其归一化后的动能图，经过阈值分割后得到动态区域掩膜。将处于动态区域掩膜的特征点进行过滤，仅使用静态特征点进行位姿估计和语义建图。在 TUM 动态数据集上实验结果表明，本文算法相较于 ORB-SLAM2 算法的平均定位精度提升了 53%，相较于 DynaSLAM 算法运行效率提升了 68.8%。本文算法虽然比 ORB-SLAM2 增加了额外时间消耗，但提升了视觉 SLAM 在动态场景中的定位精度和鲁棒性。

参考文献：

[1] 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. DOI: 10.1109/TPAMI.2007.1049.

[2] PIRE T, FISCHER T, CASTRO G, et al. S-PTAM: Stereo parallel tracking and mapping[J]. Robotics and Autonomous Systems, 2017, 93: 27-42. DOI: 10.1016/j.robot.2017.03.019.

[3] NEWCOMBE R A, LOVEGROVE S J, DAVISON A J. DTAM: Dense tracking and mapping in real-time[C]//2011 International Conference on Computer Vision. Piscataway: IEEE, 2011: 2320-2327. DOI: 10.1109/ICCV.2011.6126513.

[4] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: large-scale direct monocular SLAM[C]//13th European Conference on Computer Vision. Heidelberg: Springer, 2014: 834-849. DOI: 10.1007/978-3-319-10605-2_54.

[5] FORSTER C, PIZZOLI M, SCARAMUZZA D. SVO: fast semi-direct monocular visual odometry[C]//2014 IEEE International Conference on Robotics and Automation (ICRA). Piscataway: IEEE, 2014: 15-22. DOI: 10.1109/ICRA.2014.6906584.

[6] WANG R, SCHWÖRER M, CREMERS D. Stereo DSO: large-scale direct sparse visual odometry with stereo cameras[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 3923-3931. DOI: 10.1109/ICCV.2017.421.

[7] 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. DOI: 10.1109/TRO.2017.2705103.

[8] ALCANTARILLA P F, YEBES J J, ALMAZÁN J, et al. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments[C]//2012 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 2012: 1290-1297. DOI: 10.1109/ICRA.2012.6224690.

[9] TAN W, LIU H M, DONG Z L, et al. Robust monocular SLAM in dynamic environments[C]//2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). Piscataway: IEEE, 2013: 209-218. DOI: 10.1109/ISMAR.2013.6671781.

[10] SHENG C, PAN S G, GAO W, et al. Dynamic-DSO: direct sparse odometry using objects semantic information for dynamic environments[J]. Applied Sciences, 2020, 10(4): 1467. DOI: 10.3390/app10041467.

[11] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision. Piscataway: IEEE, 2017: 2980-2988. DOI: 10.1109/ICCV.2017.322.

[12] LI S L, LEE D. RGB-D SLAM in dynamic environments using static point weighting[J]. IEEE Robotics and Automation Letters, 2017, 2(4): 2263-2270. DOI: 10.1109/LRA.2017.2724759.

[13] 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. DOI: 10.1109/WACV.2018.00115.

[14] ZHONG F W, WANG S, ZHANG Z Q, et al. Detect-SLAM: making object detection and SLAM mutually beneficial[C]//2018 IEEE Winter Conference on Applica-

tions of Computer Vision (WACV). Piscataway: IEEE, 2018: 1001-1010.

[15] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multiBox detector[C]//14th European Conference on Computer Vision.Heidelberg: Springer, 2016: 21-37. DOI: 10.1007/978-3-319-46448-0_2.

[16] YU C, LIU Z X, LIU X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway: IEEE, 2018: 1168-1174. DOI: 10.1109/IROS.2018.8593691

[17] 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. DOI: 10.1109/TPAMI.2016.2644615.

[18] BOLYA D, ZHOU C, XIAO F Y, et al. YOLACT: real-time instance segmentation[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 9156-9165. DOI: 10.1109/ICCV.2019.00925.

[19] HUI T W, TANG X O, LOY C C. A lightweight optical flow CNN—Revisiting data fidelity and regularization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(8): 2555-2569. DOI: 10.1109/TPAMI.2020.2976928.

[20] ILG E, MAYER N, SAIKIA T, et al. FlowNet 2.0: evolution of optical flow estimation with deep networks[C]// Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017: 1647-1655. DOI: 10.1109/CVPR.2017.179

作者简介:

张　禹　博士, 教授, zhangyu_nt@163.com

高　新(通信作者)　硕士研究生, 15841132876@163.com