第 42 卷 第 14 期/2022 年 7 月/光学学报 研究论文

光学学报

动态场景下基于光流和实例分割的视觉SLAM方法

徐陈，周怡君，罗晨*

东南大学机械工程学院，江苏 南京 211189

摘要 为提升动态场景中视觉SLAM（Simultaneous Localization and Mapping）系统的定位精度和鲁棒性，提出一种基于光流和实例分割的视觉SLAM方法。针对动态物体和静态背景光流方向的不一致性，提出一种高实时性动态区域掩模检测算法，从而在ORB-SLAM2原有跟踪线程中实时地剔除处于动态区域掩模中的特征点。利用已有深度图和跟踪线程位姿估计的信息去除相机运动相关光流，然后聚类动态物体自身运动产生的光流幅值，从而实现高精度的动态区域掩模检测，并结合对极几何约束剔除局部建图线程中的动态路标点。在TUM和KITTI数据集上的测试结果表明，在高动态场景下，本文算法相较ORB-SLAM2、Detect-SLAM、DS-SLAM，定位精度平均提升97%、64%和44%。相较DynaSLAM，本文算法在一半的高动态场景中定位精度平均提升20%，这验证了本文算法在高动态场景中提升了系统定位精度和鲁棒性。

关键词 机器视觉；视觉里程计；动态场景；光流；运动物体检测；实例分割

中图分类号 TP242.6 **文献标志码** A **DOI**：10.3788/AOS202242.1415002

Visual SLAM Method Based on Optical Flow and Instance Segmentation for Dynamic Scenes

Xu Chen, Zhou Yijun, Luo Chen*

School of Mechanical Engineering, Southeast University, Nanjing 211189, *Jiangsu, China*

Abstract In order to improve the accuracy and robustness of visual SLAM (Simultaneous Localization and Mapping) systems in dynamic scenes, a visual SLAM algorithm based on optical flow and instance segmentation is proposed. Aiming at the inconsistency of optical flow direction between dynamic objects and static background, feature points in the dynamic region mask can be eliminated in the original tracking thread of ORB-SLAM2 in real time. We use the existing depth map and tracking thread pose estimation information to remove the optical flow related to camera motion and then cluster the optical flow amplitude generated by the dynamic object's own motion to achieve high-precision dynamic area mask detection. The dynamic landmarks in the local mapping thread are eliminated combined with epipolar geometric constraints. Finally, the test results on TUM and KITTI datasets show that in high dynamic scenes, compared with ORB-SLAM2, Detect-SLAM, and DS-SLAM, the accuracy of the proposed algorithm is improved by 97%, 64%, and 44% on average. Compared with DynaSLAM, the accuracy has an average increase of 20% in half of the high dynamic scenes, which verifies that the proposed algorithm improves the accuracy and robustness of the system in high dynamic scenes.

Key words machine vision; visual odometry; dynamic scene; optical flow; motion object detection; instance segmentation

1 引 言

视觉同步定位与地图构建系统是建立在场景中物体都是静态的假设前提下，但是现实场景中总会存在诸如行走的人、移动的汽车等动态物体。若此类物体在视场中占很大比例，会严重影响系统的定位精度和鲁棒性。

目前动态环境下视觉定位问题的研究主要分为两类。一类是基于传统的多视图几何方法及其改进方法。Alcantarilla等[1]通过环境的稠密场景流及其派生

收稿日期：2021-11-30；**修回日期**：2021-12-10；**录用日期**：2022-01-24

基金项目：国家自然科学基金（51975119）、江苏省“六大人才高峰”高层次人才项目（GDZB-002）

通信作者：*chenluo@seu.edu.cn

的残差运动似然来检测动态物体，从而改善了SLAM（Simultaneous Localization and Mapping）系统在动态场景中的性能。但是该系统在纹理较弱场景中容易将静态点误检为动态点。Kerl等[2]基于光度一致性的假设，构造了融合运动先验权重像素点之间的光度误差函数，然后通过最小化误差函数来优化相机的位姿，然而该方法只能剔除一部分动态特征，系统鲁棒性较低。Wei等[3]提出了一种新的在线关键帧表示和更新方法，能够在动态环境中有效地检测和处理物体外观或结构的变化，实现了自适应地建模动态环境，同时给出了一种新的基于先验的自适应RANSAC（Random Sample and Consensus）算法（PARSAC），该算法即便在离群点占比较低时，也能高效地去除离群点。Li等[4]提出一种深度边缘点静态加权方法，采用静态权重表示一个点属于静态环境的概率，并按此权重剔除关键帧中的动态特征点。Sun等[5]基于自身运动补偿图像差分、粒子滤波和深度图前景矢量化精确地跟踪运动面片，并将其集成到深度相机（RGB-D）SLAM的前端作为预处理，过滤掉与运动对象相关的数据。张磊等[6]基于动态特征剔除图像与点云中的运动部分，并通过自适应加权的方法有效融合了视觉和激光信息，提高了位姿估计精度。另一类是利用语义信息从背景分割对象以区分动态点和静态点的方法。Bescos等[7]利用Mask-RCNN和多视角几何方法检测动态物体，并在ORB-SLAM2系统的基础上提出DynaSLAM算法。通过Mask-RCNN（Region-Convolutional Neural Network）和多视角几何方法检测动态特征点，并将其过滤。Zhong等[8]使用SSD（Single Shot MultiBox Detector）检测图像中的物体，依据先验概率标记动态物体种类，过滤动态物体的所有特征点，然后估计相机位姿。Yu等[9]利用SegNet获取图像中的语义信息，并结合运动一致性检测来过滤动态物体上的动态点，在ORB-SLAM的基础上提出DS-SLAM（semantic visual SLAM towards dynamic environments）算法。Wang等[10]通过深度神经网络将场景中物体的运动状态分为非静态和静态。在此基础上，提出了去除跟踪线程和局部建图线程内运动地图点的语义数据关联方法。作者改进了CenterNet作为移动对象检测线程，以提供语义信息和粗定位，并提出了一种新颖的语义数据关联方法，将语义信息从二维拓展到三维。徐雪松等[11]通过优化重投影误差得到单应性矩阵，求解运动补偿帧，生成4帧差分图。对此4帧差分图进行滤波、二值化和形态学处理，再结合目标检测网络输出的结果，检测并剔除了图像动态区域。卢金等[12]将语义信息与几何特征紧密结合，提出了一种多尺度的随机采样一致性方法，该方法提高了外点检测的鲁棒性。

基于传统多视图几何的方法在大部分场景运动的情况下鲁棒性很低；通过融合语义信息区分动态点和静态点的方式会将潜在运动特征点（实际静止的运动类别的物体，如停在路边的车）都剔除，且受制于语义分割网络的先验训练，若场景有训练集中未出现的动态物体，则会降低系统鲁棒性。针对上述两种方法存在的问题，本文提出一种结合实例分割和光流的视觉SLAM方法。

2 动态场景下的视觉SLAM算法

2.1 视觉SLAM系统框架

本文采用RGB-D相机作为传感器。图1为完整SLAM系统框架，灰色框内为ORB-SLAM2的线程，虚线框内为本文在原有跟踪、局部建图、回环检测三个线程的基础上增加的运动区域检测线程。图1中sim3表示在ORB-SLAM2中随机采样3对点，并通过计算得来的相似变换矩阵

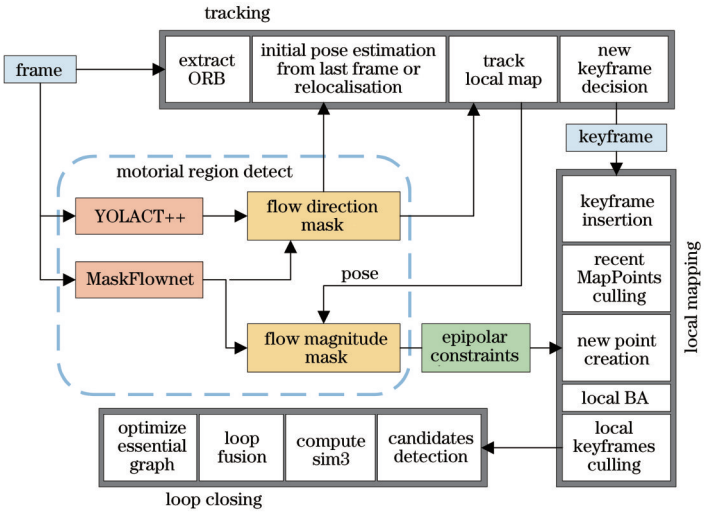

图1 系统框架图

Fig. 1 System framework

当对进入跟踪线程的新图像帧进行特征检测时，将图像传入运动区域检测线程并行进行光流网络和实例分割网络的前向推理。然后基于光流方向的动态区域检测模块，根据光流场数据和实例分割的结果进行动态区域的检测，生成基于光流方向的动态区域掩模用于跟踪线程的初始位姿估计和局部地图跟踪模块，

并对动态区域内的特征点进行剔除。

由于深度图失效点修复耗时较长，为保证实时性，采用基于光流场幅值的动态区域检测模块而不对每一帧图像进行检测。若跟踪线程阶段将当前帧作为关键帧，图 1 中基于光流场幅值的动态区域检测模块根据深度信息、光流场信息和跟踪线程中局部地图跟踪提供的帧间位姿，生成相应的掩模。光流预测结果、相机位姿变换和深度值受噪声影响，导致检测出的动态区域存在误差，因此在生成动态区域掩模后，epipolar constraints 模块对掩模内的特征点进行对极几何约束的检查，将掩模之内不符合对极几何约束的特征点判定为外点，且不生成对应的地图路标点。

基于光流方向的动态区域检测模块满足实时性要求，但实例分割的漏检会影响该模块的检测精度。因此，在局部建图线程关键帧路标点创建过程中，利用基于光流场幅值的动态区域检测模块对路标点进行进一步筛选，可减小实例分割漏检的影响。本文在保证实时性、追求高准确性的前提下选用 YOLACT++[13] 作为实例分割模块，MaskFlownet-S[14] 作为光流预测模块。

2.2 对极几何约束

如图 1 中 epipolar constraints 模块所示，基于光流幅值的动态区域掩模（flow magnitude mask）生成之后，使用对极几何约束对掩模区域外的静态特征点进行进一步确认，其原理如图 2 所示（T_{12} 为 O_1 点到 O_2 点的位姿变换矩阵），利用位于两个不同位置的相机对点 P 进行观测，假设点 P 在两帧图像上的对应坐标向量分别为 $\boldsymbol{x}_1=\begin{bmatrix}u_1 & v_1 & 1\end{bmatrix}^{\mathrm{T}}$ 和 $\boldsymbol{x}_2=\begin{bmatrix}u_2 & v_2 & 1\end{bmatrix}^{\mathrm{T}}$，则 O_2 帧上极线的方向向量 $\boldsymbol{l}'$ 为

$$\boldsymbol{l}'=\begin{bmatrix}X'\\Y'\\Z'\end{bmatrix}=\boldsymbol{F}\boldsymbol{x}_1, \tag{1}$$

式中：$\boldsymbol{F}$ 为相机两位置之间的基础矩阵；$\begin{bmatrix}X' & Y' & Z'\end{bmatrix}^{\mathrm{T}}$ 为极线方向向量。则 $\boldsymbol{x}_2$ 到极线的距离 d 为

$$d=\frac{\left|\boldsymbol{x}_2^{\mathrm{T}}\boldsymbol{F}\boldsymbol{x}_1\right|}{\sqrt{\left|X'\right|^2+\left|Y'\right|^2}}。 \tag{2}$$

理想情况下距离 $d=0$，由于相机采集的图像存在噪声和畸变，点与极线间距离不为 0。因此，设置阈值 d_{th}，若求得的距离大于此阈值 d_{th}，则认为该点不符合极线约束。

2.3 基于光流场矢量方向的动态区域检测

光流是空间物体的运动投影到相机成像平面上运动的瞬时速度。光流场是包含了整个图像所有像素瞬时运动信息的矢量场，通常以二维矢量的形式表示，即光流场中每一像素点的光流矢量可分解为水平分量和竖直分量。

根据反正切函数计算像素坐标系下当前帧光流场中各个像素点的光流矢量与水平轴正方向的夹角（取夹角范围为 0～2π），将夹角线性映射到 0～255 的灰度范围中，得到如图 3(a)所示的光流场矢量方向可视化图，由于光流矢量的方向角为 0 和 2π，图中存在一条黑白分界线。

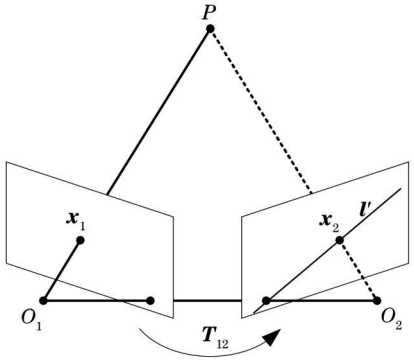

图 2 对极几何约束

Fig. 2 Epipolar geometry constraints

为避免后续边缘检测检出此分界线，将各像素点的光流矢量与像素坐标系水平轴的夹角按式(3)转换（此转换不影响光流场矢量与水平轴正方向夹角的连续性），得到如图 3(b)所示没有分界线的光流场矢量方向图。

$$f_{\mathrm{angel}}=\begin{cases}\theta, & 0\leqslant\theta<\pi\\ 2\pi-\theta, & \pi\leqslant\theta<2\pi\end{cases}, \tag{3}$$

式中：θ 为光流矢量与水平轴正方向的夹角；f_{angel} 为转换之后的光流矢量与水平轴正方向的夹角。

图 3(b)为有动态物体的光流场矢量方向图。显而易见，图 3(b)中的动态物体（人）的边界和背景存在明显的差异，这是由动态物体相对于相机的运动与背景相对于相机的运动不一致导致。基于上述矢量方向图，采用 Canny[15] 边缘检测即可提取场景中动态物体的边缘，如图 3(c)所示。

若相机在三维空间中连续运动，则相机所拍摄的图像帧中场景发生变化。假设相机处于静止状态，可看作场景相对于相机在运动。由于相机成像在二维平面上，场景的三维运动则对应地投影到相机成像平面上的二维运动。

将相机拍摄的场景（图像区域）看作一个均质刚体，则相机成像平面上该帧图像的二维运动可分解为绕旋转中心（图像几何中心，相机光心）的转动和旋转中心自身的平动。若平动的速度为 0，只有转动速度，则图像几何中心的速度为 0，相应的图像几何中心的像素点的光流矢量也为 0。在平动速度为 0 的条件下，图像中其他像素点处光流矢量仅由自身的旋转运动（旋转角速度为 ω_{o}）导致，故其他像素点处光流矢量与图像几何中心指向该像素点的向量相垂直。因此，图像几何中心（瞬时速度矢量为 0 的点，也称为速度瞬心）周围的光流矢量方向夹角会从 0 急剧变化到 2π。若平动的速度不为 0，则速度瞬心位置会发生偏移[图 3(d)]，从而导致边缘检测结果中出现如图 3(d)中 O' 处区域所示的如同聚集性褶皱的复杂边缘。

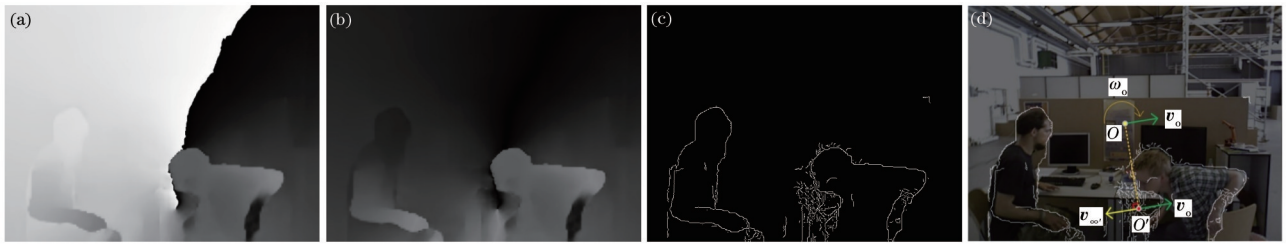

图 3　光流场方向的动态区域检测。(a)原始光流场矢量方向可视化图;(b)变换后光流场矢量方向可视化图;(c)光流场矢量方向图边缘;(d)速度瞬心

Fig. 3　Dynamic region detection of optical flow field direction. (a) Visualization of vector direction of original optical flow field; (b) visualization of vector direction of optical flow field after transformation; (c) edge of optical flow field vector direction; (d) instantaneous centre of velocity

当图像无旋转时,速度瞬心位于无穷远处;当图像旋转速度远小于平动速度时,速度瞬心位于图像之外,不会出现褶皱边缘;而当速度瞬心位于图像之内时,会出现褶皱边缘。因此,褶皱边缘的出现与否取决于速度瞬心是否位于图像内。如图 4 所示,以速度瞬心 O 为坐标系原点建立图示坐标系,θ_1、θ_2、θ_3、θ_4 分别为 $\angle AOB$、$\angle BOC$、$\angle COD$、$\angle DOA$ 的大小,令 4 个夹角之和为 θ_{sum}($\theta_{sum}=\theta_1+\theta_2+\theta_3+\theta_4$)。如图 4(a)所示,当图像占据两个象限时,$\theta_{sum}\leqslant 2\pi$[如图 4(d)所示,当且仅当速度瞬心位于图像边界上时,$\theta_{sum}=2\pi$];如图 4(b)所示,当图像仅占据一个象限时,$\theta_{sum}\leqslant\pi$[如图 4(e)所示,当且仅当速度瞬心位于图像角点上时,$\theta_{sum}=\pi$];如图 4(c)所示,当图像占据所有 4 个象限(速度瞬心位于图像之内)时,$\theta_{sum}=2\pi$。

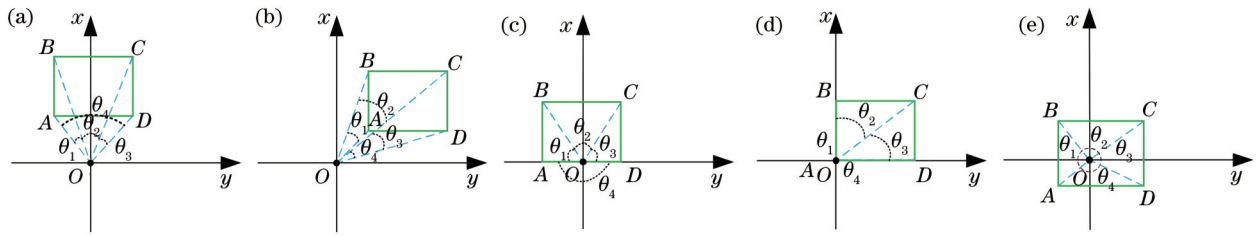

图 4　速度瞬心位置图。(a)占据两个象限;(b)占据一个象限;(c)占据四个象限;(d)占据两个象限及边界;(e)占据一个象限及边界

Fig. 4　Velocity instantaneous centre positions. (a) Occupying two quadrants; (b) occupying single quadrant; (c) occupying four quadrants; (d) occupying two quadrants and their boundaries; (e) occupying single quadrant and its boundaries

图像上各像素点的光流矢量与速度瞬心指向该像素点的向量相垂直,故光流矢量与水平轴正方向的夹角和速度瞬心指向该像素点的向量与水平轴正方向的夹角之间相差 $\pi/2$。因此,图像 4 个角点处的光流矢量与水平轴正方向的夹角之间的差值等于速度瞬心指向该像素点的向量与水平轴正方向的夹角之间的差值,故可用式(3)求得的角度来计算 4 个夹角之和 θ_{sum}。由于光流网络预测结果存在误差,选取阈值 θ_{th},当 $\theta_{sum}>\theta_{th}$ 时,速度瞬心位于图像内。

若速度瞬心位于图像内,为避免速度瞬心周围的褶皱边缘对动态目标边缘的干扰、便于后续边缘检测,本文将其移出图像可见区域。像素点的光流矢量可分解为旋转分量与平动分量,所有像素点的平动分量相同,关于相机光心对称的一对像素点的旋转分量大小相同、方向相反。按式(4)将图像的所有光流矢量求取平均值,则可得到平动分量的近似值(图像中存在动态区域,图像几何中心与相机光心不重合,光流预测结果存在误差)。

$$\begin{cases}\bar{u}=\sum_{i=1}^{M}\sum_{j=1}^{N}\dfrac{u(i,j)}{MN}\\ \bar{v}=\sum_{i=1}^{M}\sum_{j=1}^{N}\dfrac{v(i,j)}{MN}\end{cases},\tag{4}$$

式中:$\bar{u}$ 为光流场在横轴方向的均值;$\bar{v}$ 为光流场在纵轴方向的均值;M 为图像的宽;N 为图像的高。

然后按式(5)计算移除速度瞬心后的光流场:

$$\begin{cases}u_{rm}(i,j)=u(i,j)+\dfrac{K}{\bar{u}^2+\bar{v}^2}\bar{u}\\ v_{rm}(i,j)=v(i,j)+\dfrac{K}{\bar{u}^2+\bar{v}^2}\bar{v}\end{cases},$$
$$i=1,2,\cdots,M,\ j=1,2,\cdots,N,\tag{5}$$

式中:u_{rm} 为光流场横轴分量移除速度瞬心后的值;v_{rm} 为光流场纵轴分量移除速度瞬心后的值;K 为常数。将速度瞬心按原本偏移方向移出图像区域,然后进行边缘检测,提取到如图 5(a)所示的动态物体边缘。

从图 5(a)中可以看出,动态物体的轮廓线仍有断开。对提取出的轮廓进行闭运算(先膨胀后腐蚀)以连接断开的轮廓,如图 5(b)所示。边缘检测出的轮廓精确度不高,且物体可能部分运动、部分静止,导致运动部分和静止部分交接边缘难以检出。将求得轮廓的外接矩形作为存在动态物体的区域,与实例分割出的潜在运动物体掩模取交集,以此得到较为精确的运动物体的轮廓,最终的光流场方向的动态区域掩模如图 5(c)所示。基于光流场方向的动态区域检测算法流程图如图 6 所示。

图 5 光流场方向的动态区域掩模。(a)光流场矢量方向图边缘;(b)边缘开运算结果;(c)动态区域掩模

Fig. 5 Dynamic region mask in optical flow direction. (a) Edge of optical flow field vector direction; (b) result of edge open operation; (c) dynamic region mask

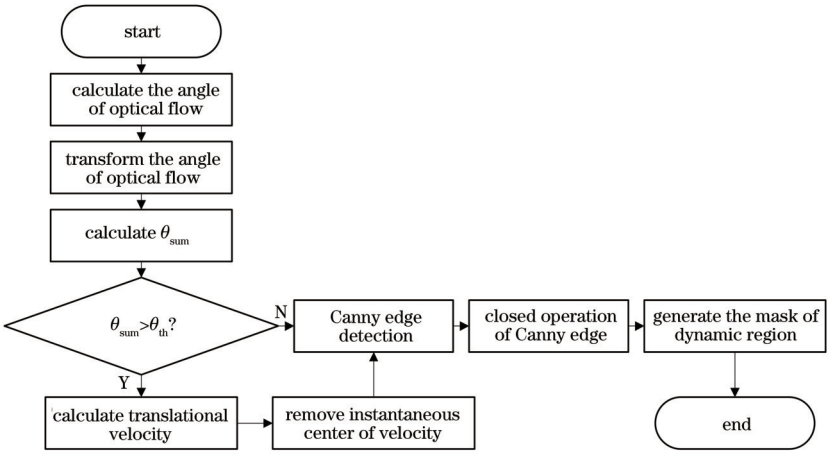

图 6 基于光流场方向的动态区域检测算法流程图

Fig. 6 Flow chart of dynamic region detection algorithm based on optical flow direction

2.4 基于光流场矢量幅值的动态区域检测

假设当前帧 I_t 的光流场 $\boldsymbol{f}_t$ 由两部分组成,一部分是相机自身运动所导致的静态场景的光流 $\boldsymbol{f}_t^{(\mathrm{static})}$,另一部分是场景中动态物体的运动所导致的光流 $\boldsymbol{f}_t^{(\mathrm{motorial})}$,则由相机自身运动所导致的静态场景光流 $\boldsymbol{f}_t^{(\mathrm{static})}$ 为

$$\boldsymbol{f}_t^{(\mathrm{static})}=\boldsymbol{K}\boldsymbol{T}_{t\to t+1}d_t^{(i)}\boldsymbol{K}^{-1}\boldsymbol{p}_i-\boldsymbol{p}_i,\ i\in I_t, \tag{6}$$

式中:$\boldsymbol{K}$ 为相机的内参矩阵;$\boldsymbol{T}_{t\to t+1}$ 为当前帧到下一帧的相机位姿变换矩阵;$\boldsymbol{p}_i$ 为当前帧中的像素坐标;$d_t^{(i)}$ 为该像素点对应的深度值。本文采用 RGB-D 相机来获取像素的深度值,将基于光流场矢量幅值的动态区域检测结果应用于局部建图线程中。式(6)中的 $\boldsymbol{T}_{t\to t+1}$ 采用跟踪线程中估计的位姿。

如图 7(a)所示,量程限制和物体表面材料特性使得 RGB-D 相机测量的深度图中存在失效点。本文采用 Telea[16] 提出的基于快速行进方法(FMM)[17] 的图像修复算法对深度图失效点进行填充。图 7(b)为修复后深度图。

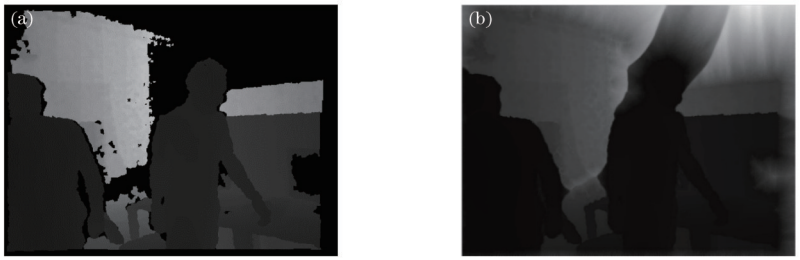

图 7 深度图。(a)原深度图;(b)修复后深度图

Fig. 7 Depth images. (a) Original depth map; (b) depth map after inpainting

如图 8 所示,白色背景为图像已知区域,Ω 为图像中的待修复区域,$\delta\Omega$ 为待修复区域的边界,p 为边界上任意一点。在点 p 周围的图像已知区域内,选择一个以 ε 为尺度的邻域 $B(\varepsilon)$,则 p 点的像素值可由邻域 $B(\varepsilon)$ 内的像素值近似计算得到。当尺度 ε 足够小时,给定点 q 的像素值 $I(q)$ 和其梯度值 $\nabla I(q)$,则点 p 的一阶估计为

$$I_q(p)=I(q)+\nabla I(q)(p-q)。 \tag{7}$$

那么,点 p 的像素值可表示为

$$I(p)=\frac{\sum_{q\in B(p)}\omega(p,q)\left[I(q)+\nabla I(q)(p-q)\right]}{\sum_{q\in B(p)}\omega(p,q)},\tag{8}$$

式中：$\omega(p,q)$为权重分布函数，其值由p、q之间的方向、距离和p到初始边界的距离所决定。处理完边界上的所有像素点后，不断迭代式(8)，并逐步收缩待修复区域的边界，直至整个区域都被修复。

上述过程中，需要求出当前边界上像素点与初始边界上的像素点的距离，再根据距离的大小顺序向待修复区域的内部逐步收缩。为此采用FMM，FMM可以明确地维持一个狭窄的通道，使得已知区域和未知区域被明显分割开。

当前帧I_t的光流场$\boldsymbol{f}_t$可视化图如图9(a)所示。通过式(6)计算得到静态场景的光流$\boldsymbol{f}_t^{(\mathrm{static})}$可视化图如图9(b)所示。按式(9)求得动态物体的运动所导致的光流$\boldsymbol{f}_t^{(\mathrm{motorial})}$，如图9(c)所示。$\boldsymbol{f}_t^{(\mathrm{motorial})}$可表示为

$$\boldsymbol{f}_t^{(\mathrm{motorial})}=\boldsymbol{f}_t-\boldsymbol{f}_t^{(\mathrm{static})}。\tag{9}$$

采用K-means聚类算法将动态物体的运动所导致的光流$\boldsymbol{f}_t^{(\mathrm{motorial})}$的幅值图中所有像素分为两类，一类为静态区域，另一类为动态区域，再按式(10)得到基于光流场幅值的动态区域掩模$\boldsymbol{M}_{t\to t+1}$，如图9(d)所示。

图9　基于光流场幅值的动态区域检测。(a)原光流；(b)相机运动光流；(c)动态物体光流；(d)动态区域掩模

Fig. 9　Dynamic region detection based on optical flow field amplitude. (a) Original optical flow; (b) optical flow of camera motion; (c) optical flow of dynamic objects; (d) dynamic region mask

$$\boldsymbol{M}_{t\to t+1}(\boldsymbol{p}_i)=\begin{cases}0, & \boldsymbol{p}_i\in\boldsymbol{R}_{\mathrm{static}}\\ 1, & \boldsymbol{p}_i\in\boldsymbol{R}_{\mathrm{motorial}}\end{cases},\tag{10}$$

式中：$\boldsymbol{R}_{\mathrm{static}}$为静态区域；$\boldsymbol{R}_{\mathrm{motorial}}$为动态区域。基于光流场方向的动态区域检测算法流程图如图10所示。

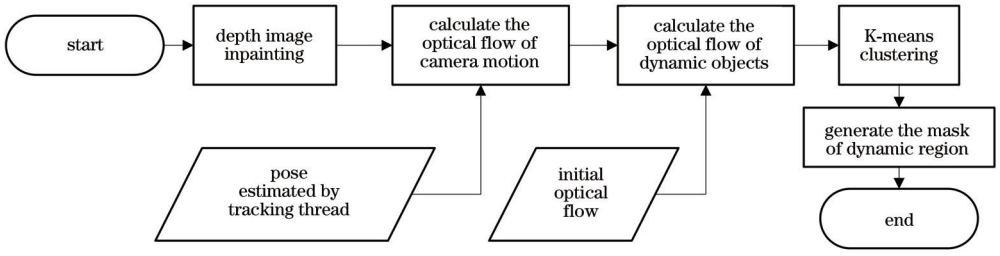

图10　基于光流场幅值的动态区域检测算法流程图

Fig. 10　Flow chart of dynamic region detection algorithm based on optical flow field amplitude

3　光流和实例分割网络

3.1　光流网络

光流预测任务(optical flow estimation)即给定一张原始图像与一张目标图像，建立一个表示从原始图像的每个像素到目标图像的对应关系的流场(flow field)。理想情况下，目标图像通过流场形变得到的形变图像应该与原始图像非常相似。但是，前景与背景之间的相对位移产生的遮挡区域(occlusions)给形变图像的获取带来了歧义与无效信息，使得光流预测结果准确率降低。

MaskFlownet[14]是一种可学习遮挡掩模的非对称特征匹配模块，该模块可预测遮挡区域、过滤特征形变带来的无效信息。通过端到端的方式无监督学习遮挡掩模，显著提升了网络的性能。

MaskFlownet包括一种轻量级的光流预测网络MaskFlownet-S，MaskFlownet-S相比MaskFlownet推理速度更快但精度稍低，本文采用FlyingChairs、FlyingThings3D和MPI Sintel数据集训练此网络。如图11所示，MaskFlownet-S的结构为结合可学习遮挡

掩模的非对称特征匹配模块(AsymOFMMs)的特征金字塔网络(FPN)结构。图 11 中可学习遮挡掩模的非对称特征匹配模块非对称地引入变形卷积(deformable convolution),即根据当前流场 ϕ 将目标特征图进行变形的同时进行一次额外的卷积,该方法打破了原始特征图与目标特征图的对称性。然后将可学习遮挡掩模 θ 以相乘的方式作用在形变之后的特征图上,过滤重影现象的干扰信息。最后,添加权衡项 μ 弥补过滤遮挡区域后缺失的原本携带的信息。图 11 中 $F^{1}(I_1)$ 和 $F^{1}(I_2)$ 为输入图像 I_1 和 I_2 经过特征金字塔采样后的特征图。

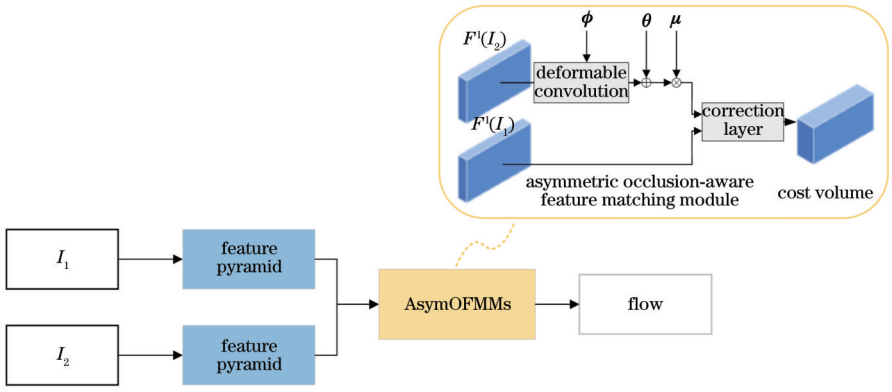

图 11　MaskFlownet-S 网络结构图

Fig. 11　MaskFlownet-S network structure

3.2　实例分割网络

本文采用 COCO 数据集训练 YOLACT++网络。YOLACT++[13]是对 YOLACT[18]的改进,YOLACT 模型采用 ResNet101 作为骨干网络。如图 12 所示,C1~C5 分别对应 ResNet 的 conv1~conv5 这 5 个卷积模块,图中 P3~P7 为 FPN,通过上采样和下采样得到不同尺度的特征图。此结构使输入图像在所有尺度上都具有较强的语义信息,金字塔中浅层特征的精确性更高,深层特征的鲁棒性更好。

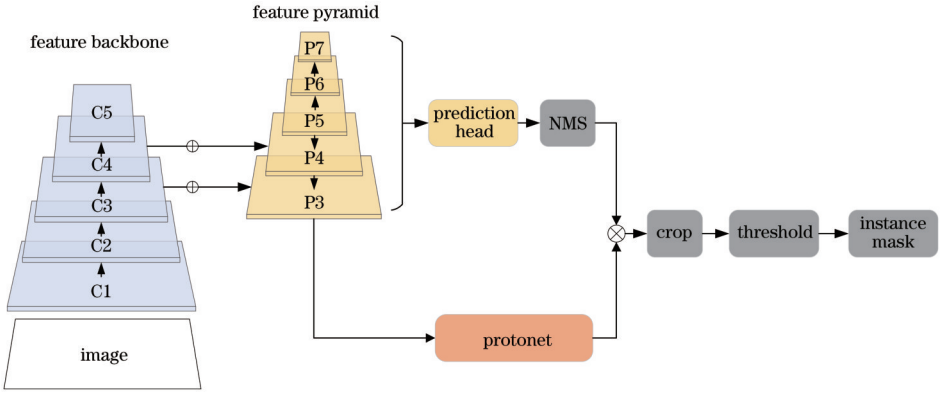

图 12　YOLACT 网络结构图

Fig. 12　YOLACT network structure

YOLACT 通过 prediction head 和 protonet 两个并行子网络实现实例分割。prediction head 分支生成目标 anchor、anchor 的位置参数、目标物体类别及其置信度、mask 掩码系数,之后通过非极大值抑制去除多余的 anchor。该分支采用共享卷积网络提高速度,达到实时分割的目的。protonet 分支生成一组原型 mask,将原型 mask 和 prediction head 分支生成的 mask 掩码系数相乘,然后进行裁剪和阈值分割,得到图像中目标物体的分割结果。

YOLACT++在 YOLACT 中引入可变形卷积,使采样点更符合物体本身的形状和尺寸;改变 anchor 的尺度大小和纵横比以增加 anchor 数量;添加 mask re-scoring 分支,通过生成 mask 的 IOU(Intersection over Union)与对应的分类置信度之间的乘积来对每个分割结果进行评分,以获得更好的结果。得益于此,YOLACT++在保证实时性的前提下,大幅提升了精度(mAP)。

4　实　　验

4.1　实验条件

本文实验平台为 Intel i7 9700 3.00 GHz CPU、32G 内存、RTX2080Ti 显卡的 PC,系统环境为 Ubuntu 18.04。SLAM 系统使用 C++编写,动态区域检测线程使用 Python 编写。实验使用 TUM 数据集和 KITTI 数据集,该数据集被广泛用于测试 SLAM 算法在室内动态环境下的定位准确性和鲁棒性。TUM

数据集包含两类场景，即高动态场景和低动态场景。将高动态场景简称为w(walking)，在高动态场景中，人在场景中围绕桌子行走，运动幅度较大；将低动态场景简称为s(sitting)，在低动态场景中，人坐在椅子上进行交谈，运动幅度较小。每个场景都包含4种不同的摄像机运动轨迹，分别是halfsphere、rpy、static和xyz。在halfsphere(简称hs)轨迹中，相机沿着半球运动；在rpy轨迹中，相机进行摇摆、俯仰运动；在static轨迹中，相机的位置保持不变；在xyz轨迹中，相机分别沿着x轴、y轴和z轴运动。KITTI数据集由一款搭载相机、激光雷达、高精度GPS/IMU导航系统的多传感器汽车平台采集而来。该数据集捕捉了具有真实交通状况的室外场景，包括乡村地区、城镇地区、高速公路等多种工况。

4.2 实验定位精度

本文采用绝对轨迹误差(ATE)和相对姿态误差(RPE)来评估算法的定位精度。表1～3分别为ATE、相对平移误差、相对旋转误差，表4～6分别为本文算法相对于ORB-SLAM2的ATE、相对平移误差、相对旋转误差的性能提升。ATE用于评估估计轨迹的全局一致性，表明了每帧相机位姿估计值与真实值之间的差值；相对位姿误差用于评估估计轨迹局部的旋转或平移误差，反映了相隔固定时间差的两帧间估计位姿变换矩阵与两帧间真实位姿变化矩阵之间的差值。每个误差表中采用均方根误差(RMSE)、中位数(median)、平均值(mean)和标准差(SD)作为评价指标。从表1可以看出，本文算法在TUM高动态场景中的定位精度相较于ORB-SLAM2平均提升97.2%。表中fr3代表TUM中的一个数据集。

表1 TUM的ATE

Table 1 ATE for TUM

unit: m

Sequence	ORB-SLAM2				Ours			
	RMSE	Median	Mean	SD	RMSE	Median	Mean	SD
fr3/s/hs	0.0234	0.0170	0.0189	0.0138	0.0162	0.0128	0.0141	0.0079
fr3/s/rpy	0.0223	0.0122	0.0169	0.0146	0.0180	0.0109	0.0142	0.0112
fr3/s/static	0.0095	0.0075	0.0083	0.0046	0.0057	0.0043	0.0049	0.0030
fr3/s/xyz	0.0089	0.0071	0.0077	0.0044	0.0082	0.0060	0.0070	0.0044
fr3/w/hs	0.5812	0.4899	0.4909	0.3112	0.0186	0.0149	0.0163	0.0090
fr3/w/rpy	0.8416	0.6353	0.7147	0.4445	0.0320	0.0232	0.0261	0.0184
fr3/w/static	0.3844	0.3309	0.3590	0.1374	0.0078	0.0063	0.0069	0.0037
fr3/w/xyz	0.6867	0.5446	0.6209	0.2932	0.0145	0.0103	0.0121	0.0081

表2 TUM的相对平移误差

Table 2 Relative translation error of TUM

unit: m/frame

Sequence	ORB-SLAM2				Ours			
	RMSE	Median	Mean	SD	RMSE	Median	Mean	SD
fr3/s/hs	0.0081	0.0058	0.0068	0.0045	0.0128	0.0090	0.0106	0.0071
fr3/s/rpy	0.0132	0.0075	0.0098	0.0089	0.0117	0.0075	0.0091	0.0073
fr3/s/static	0.0055	0.0041	0.0047	0.0028	0.0052	0.0040	0.0045	0.0026
fr3/s/xyz	0.0083	0.0067	0.0072	0.0042	0.0081	0.0064	0.0070	0.0041
fr3/w/hs	0.0242	0.0133	0.0186	0.0155	0.0135	0.0091	0.0110	0.0079
fr3/w/rpy	0.0715	0.0166	0.0240	0.0674	0.0182	0.0108	0.0139	0.0117
fr3/w/static	0.0178	0.0076	0.0119	0.0132	0.0063	0.0039	0.0048	0.0040
fr3/w/xyz	0.0372	0.0172	0.0224	0.0298	0.0123	0.0080	0.0099	0.0074

表7为本文算法与目前领先的SLAM算法(DS-SLAM[9]、Dyna-SLAM[7]、Detect-SLAM[8]和SOF-SLAM[19])的对比，可看出本文算法在TUM高动态场景中的误差都比DS-SLAM小，相较Dyna-SLAM在w-hs、w-rpy和w-xyz中的定位精度均有一定的提升，且在w-hs、w-rpy场景中精度提升超过20%，和其他所有算法相比，w-hs和w-rpy场景中本文算法精度最高。

图13、14分别为TUM数据集的低动态场景和高动态场景各个序列下ORB-SLAM2和本文算法估计的轨迹和真实轨迹的对比和误差图。对于高动态的场景中，从轨迹图中可以看出本文算法估计的轨迹与灰色的真实轨迹偏差很小，而ORB-SLAM2估计的轨迹与真实轨迹偏差较大。从误差图中可以看出本文算法的单帧误差波动不大，而ORB-SLAM2的单帧误差波动很大，由于误差较大的帧中的动态区域占比很大，而这种帧在序列中总是集中出现，故ORB-SLAM2的单帧误差波动较大。在低动态场景中，由于ORB-SLAM2

表 3　TUM 的相对旋转误差

Table 3　Relative rotation error of TUM　　unit: (°)/frame

Sequence	ORB-SLAM2				Ours			
	RMSE	Median	Mean	SD	RMSE	Median	Mean	SD
fr3/s/hs	0.3620	0.2626	0.3103	0.1864	0.4095	0.3064	0.3508	0.2113
fr3/s/rpy	0.4097	0.2947	0.3396	0.2291	0.4036	0.3013	0.3443	0.2108
fr3/s/static	0.1646	0.1204	0.1411	0.0848	0.1638	0.1190	0.1390	0.0867
fr3/s/xyz	0.3152	0.2354	0.2685	0.1651	0.3134	0.2361	0.2669	0.1643
fr3/w/hs	0.6217	0.4168	0.5034	0.3648	0.4281	0.3002	0.3569	0.2363
fr3/w/rpy	1.5127	0.4391	0.5985	1.3892	0.4785	0.3219	0.3858	0.2829
fr3/w/static	0.3719	0.2070	0.2729	0.2526	0.1815	0.1288	0.1520	0.0992
fr3/w/xyz	0.7920	0.4216	0.5210	0.5965	0.3902	0.2354	0.2794	0.2723

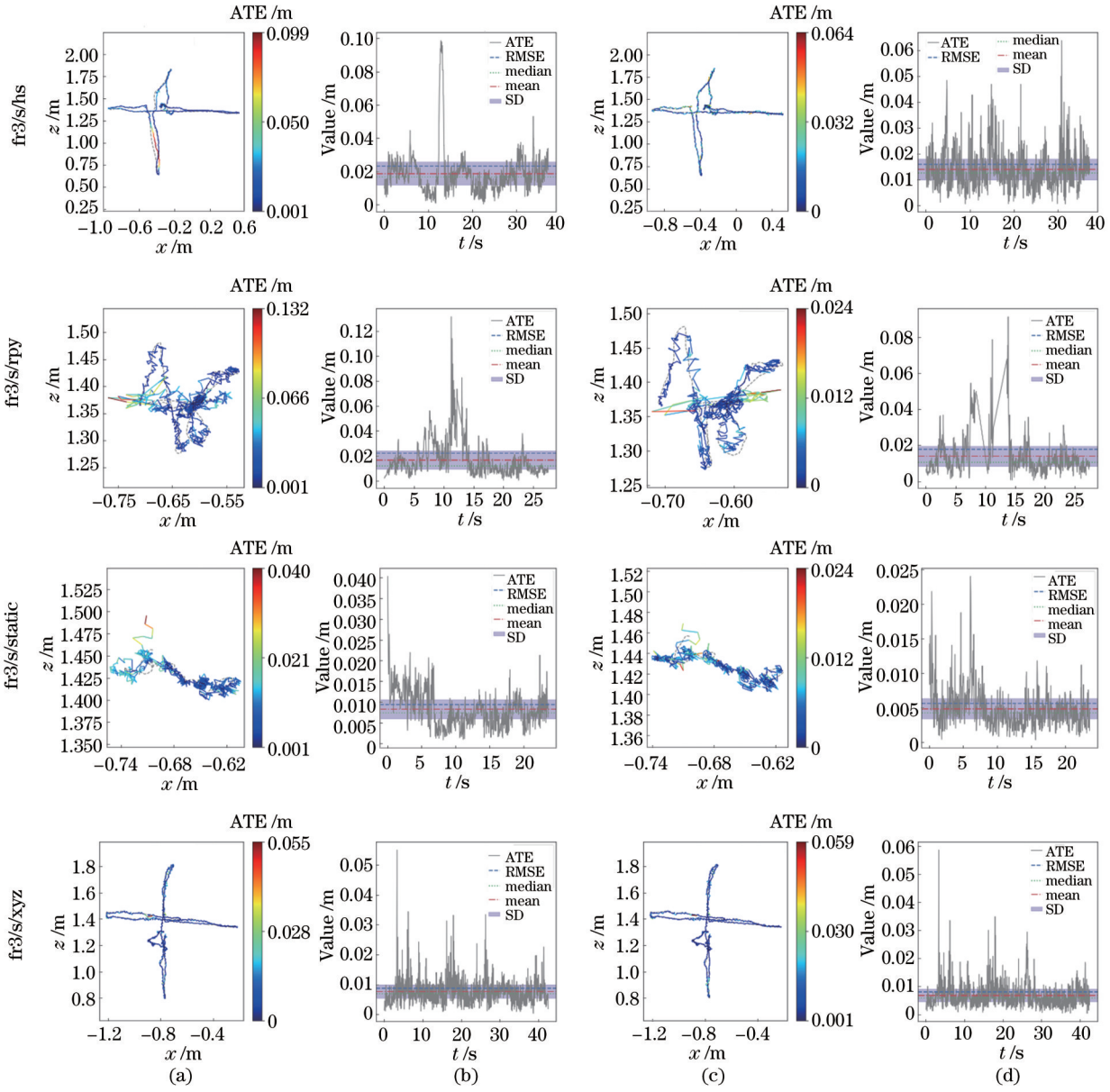

图 13　TUM 的轨迹与误差图。(a)(b) ORB-SLAM2 的轨迹与误差；(c)(d)本文算法的轨迹与误差

Fig. 13　Trajectory and error of TUM. (a)(b) Trajectory and error of ORB-SLAM2; (c)(d) trajectory and error of proposed algorithm

本身具有一定动态特征剔除能力，故其和本文算法估计的轨迹与真实轨迹的偏差都较小。

本文算法与 ORB-SLAM2、Dyna-SLAM[7] 和 VDO-SLAM[20] 针对 KITTI 双目数据集的测试结果对

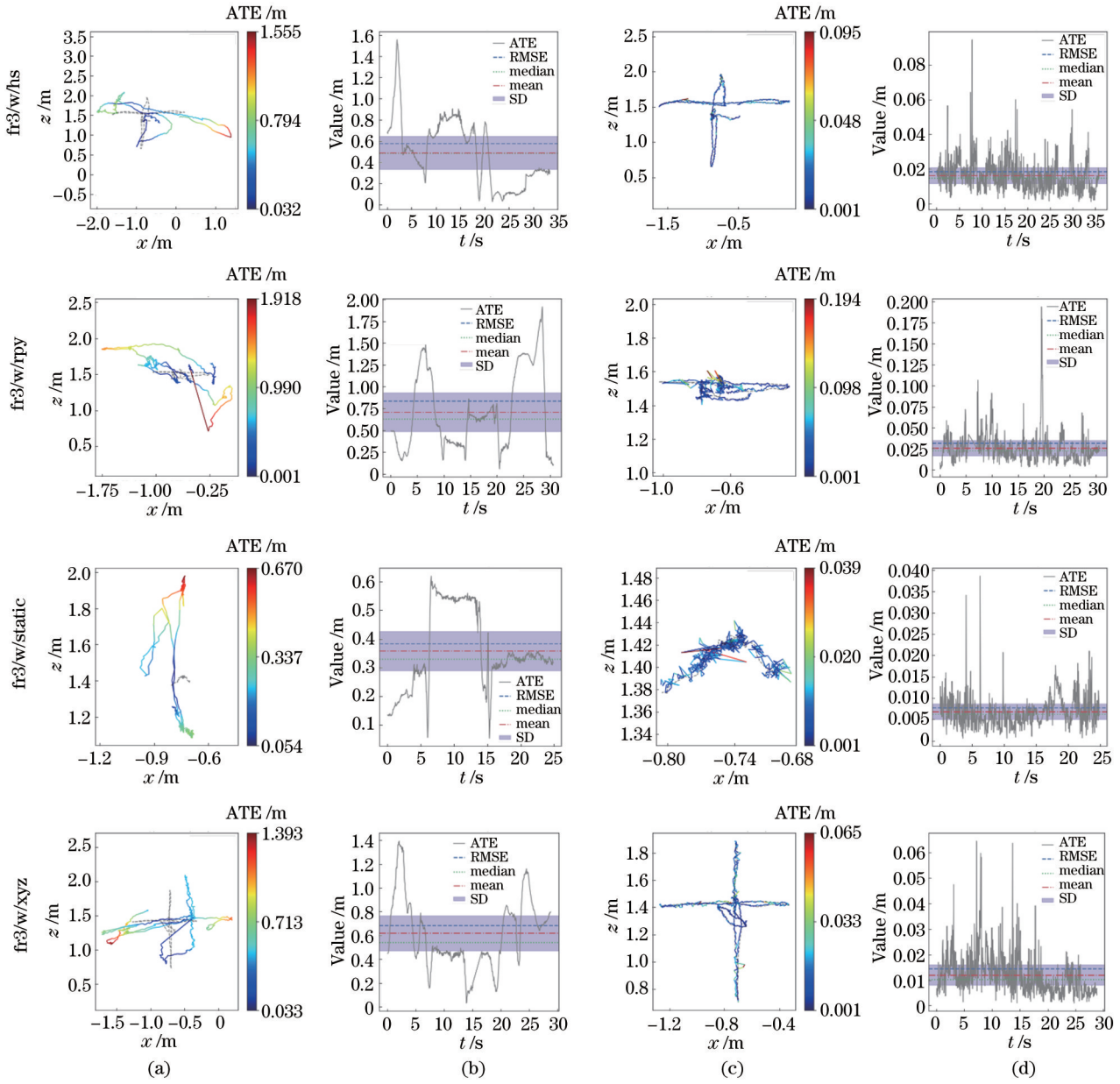

图 14　TUM 的轨迹与误差图。(a)(b) ORB-SLAM2 的轨迹与误差;(c)(d)本文算法的轨迹与误差

Fig. 14　Trajectory and error of TUM. (a)(b) Trajectory and error of ORB-SLAM2; (c)(d) trajectory and error of proposed algorithm

表 4　TUM 的 ATE 性能提升

Table 4　Improvement of ATE for TUM　　unit: %

Sequence	RMSE	Median	Mean	SD
fr3/s/hs	30.8	24.4	25.1	42.9
fr3/s/rpy	19.1	10.9	16.0	23.3
fr3/s/static	39.9	42.5	41.3	35.3
fr3/s/xyz	7.7	15.0	10.3	0
fr3/w/hs	96.8	97.0	96.7	97.1
fr3/w/rpy	96.2	96.3	96.3	95.9
fr3/w/static	98.0	98.1	98.1	97.3
fr3/w/xyz	97.9	98.1	98.1	97.2

表 5　TUM 的相对平移误差性能提升

Table 5　Improvement of relative translation error for TUM

unit: %

Sequence	RMSE	Median	Mean	SD
fr3/s/hs	−57.4	−55.5	−57.3	−57.5
fr3/s/rpy	11.9	0.4	7.1	18.0
fr3/s/static	5.2	1.1	4.2	8.0
fr3/s/xyz	2.7	4.2	3.1	1.4
fr3/w/hs	44.1	31.6	40.8	49.2
fr3/w/rpy	74.6	34.8	42.0	82.7
fr3/w/static	64.7	48.3	59.8	69.3
fr3/w/xyz	66.9	53.6	55.9	75.1

比如表 8 所示,本文算法的相对平移误差相较于 ORB-SLAM2 和 Dyna-SLAM 分别平均减小 12% 和 9%,相

表 6 TUM的相对旋转误差性能提升

Table 6 Improvement of relative rotation error for TUM

unit: %

Sequence	RMSE	Median	Mean	SD
fr3/s/hs	−13.1	−16.7	−13.1	−13.3
fr3/s/rpy	1.5	−2.2	−1.4	8.0
fr3/s/static	0.5	1.2	1.5	−2.3
fr3/s/xyz	0.6	−0.3	0.6	0.5
fr3/w/hs	31.1	28.0	29.1	35.2
fr3/w/rpy	68.4	26.7	35.5	79.6
fr3/w/static	51.2	37.8	44.3	60.7
fr3/w/xyz	50.7	44.2	46.4	54.3

较于VDO-SLAM平均减小69%。表8中RPE表示平均相对位姿误差，RPE可分为RPEt和RPEr，RPEt表示平均相对平移误差，RPEr表示平均相对旋转误差。由于KITTI数据集中并没有深度图数据，若是添加深度预测网络进行预处理，则需要同时推理三个深度学习网络模型，这大大降低了系统的实时性。而基于光流方向的动态区域检测不需要深度图数据，故本文仅针对KITTI双目数据集采用基于光流方向的动态区域检测算法。由表8中数据可知，本文算法的RMSE和RPEt相较于对比算法均有减小，但定位精度提升

表 7 TUM的绝对轨迹RMSE对比

Table 7 Comparison of RMSE of absolute trajectory for TUM

unit: m

Sequence	DS-SLAM	Dyna-SLAM	Detect-SLAM	SOF-SLAM	Ours
fr3/s/hs	—	0.0170	0.0231	—	**0.0162**
fr3/s/rpy	—	—	—	—	**0.0180**
fr3/s/static	0.0065	—	—	0.0100	**0.0057**
fr3/s/xyz	—	0.0150	0.0201	—	**0.0082**
fr3/w/hs	0.0303	0.0250	0.0514	0.0290	**0.0186**
fr3/w/rpy	0.4442	0.0400	0.2959	**0.0270**	0.0320
fr3/w/static	0.0081	**0.0060**	—	0.0070	0.0078
fr3/w/xyz	0.0247	0.0150	0.0241	0.0180	**0.0145**

幅度不如表7所示的TUM数据集测试结果，其具体原因归结为以下两点：1)KITTI数据集的大部分序列场景中的潜在移动目标(车辆)均为静止，存在动态目标的序列(如01和09)中的动态目标仅占图像中小比例(TUM数据集中动态目标占图像比例较大)，故性能提升不明显；2)仅采用基于光流方向的动态区域检测，没有深度图的数据集导致无法使用基于光流幅值的算法再次进行检测，故无法检出实例分割漏检的目标。但第二种情况中基于光流方向的动态区域检测算法的漏检率较低，其主要原因还是在于第一点。

表 8 KITTI数据集测试结果对比

Table 8 Comparison of test results on KITTI dataset

Sequence	ORB-SLAM2			DynaSLAM			VDO-SLAM			Ours		
	RMSE of ATE / m	RPEt / (m·frame^{-1})	RPEr / [(°)·frame^{-1}]	RMSE of ATE / m	RPEt / (m·frame^{-1})	RPEr / [(°)·frame^{-1}]	RMSE of ATE / m	RPEt / (m·frame^{-1})	RPEr / [(°)·frame^{-1}]	RMSE of ATE / m	RPEt / (m·frame^{-1})	RPEr / [(°)·frame^{-1}]
00	1.30	0.04	0.06	1.40	0.04	0.06	—	0.05	**0.05**	**1.24**	**0.02**	0.06
01	10.40	0.05	**0.04**	9.40	0.05	**0.04**	—	0.12	**0.04**	**9.03**	**0.05**	**0.04**
02	5.70	0.04	0.03	6.70	0.04	0.03	—	0.04	**0.02**	**5.35**	**0.02**	0.05
03	0.60	0.07	**0.04**	0.60	0.07	**0.04**	—	0.09	**0.04**	**0.59**	**0.02**	**0.04**
04	0.20	0.07	0.06	0.20	0.07	0.06	—	0.11	0.05	**0.17**	**0.02**	**0.03**
05	0.80	0.06	0.03	0.80	0.06	0.03	—	0.10	**0.02**	**0.77**	**0.01**	0.04
06	0.80	0.02	0.04	0.80	0.02	0.04	—	0.02	0.05	**0.72**	**0.02**	**0.03**
07	0.50	0.05	0.07	0.50	0.05	0.07	—	—	—	**0.46**	**0.01**	**0.04**
08	3.60	0.08	**0.04**	3.50	0.08	**0.04**	—	—	—	**3.19**	**0.03**	0.05
09	3.20	0.06	**0.05**	1.60	0.06	**0.05**	—	—	—	**1.57**	**0.02**	**0.05**
10	1.00	0.07	**0.04**	1.20	0.07	**0.04**	—	—	—	**0.96**	**0.01**	0.05

图15所示为本文算法得到的KITTI数据集中01和09两个有较多动态目标的序列的轨迹与误差图，根据表8中相应数据，相较于其他序列，01和09序列的定位精度提升较大，验证了本文算法的效果。

4.3 实验算法运行时间

算法各模块的耗时如表9所示，其中静态场景光流计算采用CUDA(Compute Unified Device Architecture)并行计算加速。光流场幅值掩模耗时较长，仅用于局部建图线程关键帧的处理，该线程与跟踪主线程并行执行，对算法运行效率影响不大。对算法影响较大的为实例分割网络和光流预测网络。未来可更换硬件条件更好的平台或采用高精度且更快速的实例分割网络和光流预测网络，使系统实时性得到进一步提升。

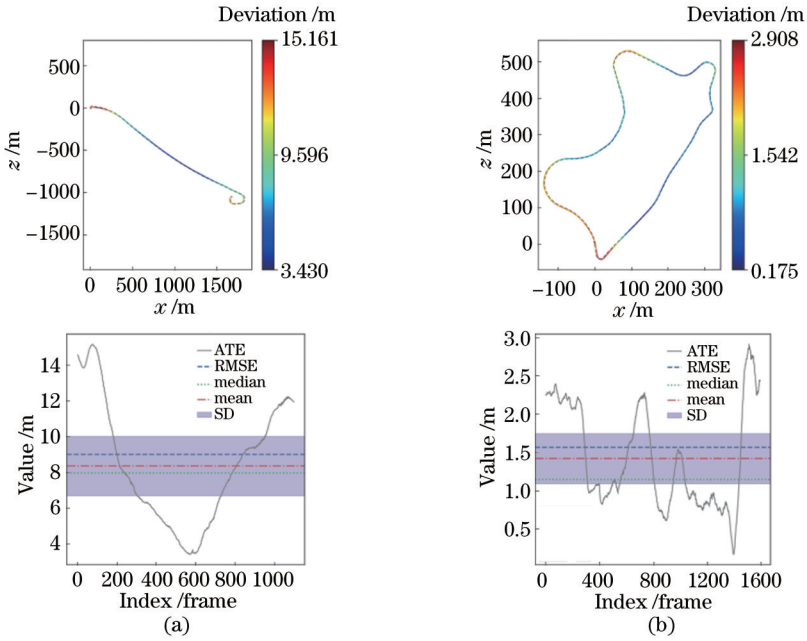

图 15　KITTI轨迹与误差图。(a) KITTI 01序列;(b) KITTI 09序列

Fig. 15　Trajectory and error of KITTI. (a) Sequence 01 of KITTI; (b) sequence 09 of KITTI

表 9　各模块运行时间

Table 9　Running time of each module　　unit: ms

Module		Time
MaskFlownet-S		27.6
YOLACT++		23.2
Optical flow field direction mask		8.1
Optical flow field amplitude mask	Depth image inpainting	73.2
	Optical flow of static scene	2.1
	K-means clustering	10.8

5　结　　论

为减少场景中的动态物体对SLAM系统精度的影响,提出了一种针对动态环境的更为鲁棒的视觉SLAM算法。该系统建立在ORB-SLAM2的基础上,基于光流和实例分割来过滤动态物体上的特征点。对每帧图像同时进行实例分割网络和光流网络的推理,之后根据光流场矢量方向信息进行动态区域掩模的检测,处于动态区域掩模中的特征点将被过滤,然后根据光流场幅值信息检测出的动态区域掩模和对极几何约束对局部建图线程中新建的地图点进行筛选。在TUM数据集和KITTI数据集上对本文算法进行了测试。TUM数据集上的测试结果表明,高动态场景下本文算法与ORB-SLAM2、Detect-SLAM、DS-SLAM算法相比,定位精度平均提升97%、64%、44%,与Dyna-SLAM相比,在一半的高动态场景中定位精度平均提升20%;KITTI数据集测试结果表明,本文算法相较ORB-SLAM2和Dyna-SLAM,定位精度平均提升12%和9%。与VDO-SLAM相比,本文算法的定位精度平均提升69%。这验证了本文算法在高动态场景中具有更好的定位精度和鲁棒性。

参　考　文　献

[1] 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, May 14-18, 2012, Saint Paul, MN, USA. New York: IEEE Press, 2012: 1290-1297.

[2] Kerl C, Sturm J, Cremers D. Robust odometry estimation for RGB-D cameras[C]//2013 IEEE International Conference on Robotics and Automation, May 6-10, 2013, Karlsruhe, Germany. New York: IEEE Press, 2013: 3748-3754.

[3] Wei T, Liu H M, Dong Z L, et al. Robust monocular SLAM in dynamic environments[C]//2013 IEEE International Symposium on Mixed and Augmented Reality, October 1-4, 2013, Adelaide, SA, Australia. New York: IEEE Press, 2013: 209-218.

[4] 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.

[5] Sun Y X, Liu M, Meng M Q H. Improving RGB-D SLAM in dynamic environments: a motion removal approach[J]. Robotics and Autonomous Systems, 2017, 89: 110-122.

[6] 张磊, 徐孝彬, 曹晨飞, 等. 基于动态特征剔除图像与点云融合的机器人位姿估计方法[J]. 中国激光, 2022, 49(6): 0610001.
Zhang L, Xu X B, Cao C F, et al. Robot pose estimation method based on image and point cloud fusion

with dynamic feature elimination[J]. Chinese Journal of Lasers, 2022, 49(6): 0610001.

[7] 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.

[8] 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 Applications of Computer Vision, March 12-15, 2018, Lake Tahoe, NV, USA. New York: IEEE Press, 2018: 1001-1010.

[9] 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), October 1-5, 2018, Madrid, Spain. New York: IEEE Press, 2018: 1168-1174.

[10] Wang S, Lü X D, Li J B, et al. Coarse semantic-based motion removal for robust mapping in dynamic environments[J]. IEEE Access, 2020, 8: 74048-74064.

[11] 徐雪松, 曾昱. 基于动态目标检测的视觉同步定位与地图构建算法[J]. 激光与光电子学进展, 2021, 58(16): 1615003.
Xu X S, Zeng Y. Visual simultaneous localization and mapping algorithm based on dynamic target detection[J]. Laser & Optoelectronics Progress, 2021, 58(16): 1615003.

[12] 卢金, 刘宇红, 张荣芬. 面向动态场景的语义视觉里程计[J]. 激光与光电子学进展, 2021, 58(6): 0611001.
Lu J, Liu Y H, Zhang R F. Semantic-based visual odometry towards dynamic scenes[J]. Laser & Optoelectronics Progress, 2021, 58(6): 0611001.

[13] Bolya D, Zhou C, Xiao F Y, et al. YOLACT++ better real-time instance segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(2): 1108-1121.

[14] Zhao S Y, Sheng Y L, Dong Y, et al. MaskFlownet: asymmetric feature matching with learnable occlusion mask[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 6277-6286.

[15] Canny J. A computational approach to edge detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, 8(6): 679-698.

[16] Telea A. An image inpainting technique based on the fast marching method[J]. Journal of Graphics Tools, 2004, 9(1): 23-34.

[17] Sethian J A, Vladimirsky A. Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes[J]. Proceedings of the National Academy of Sciences of the United States of America, 2000, 97(11): 5699-5703.

[18] Bolya D, Zhou C, Xiao F Y, et al. YOLACT: real-time instance segmentation[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV), October 27-November 2, 2019, Seoul, Korea (South). New York: IEEE Press, 2019: 9156-9165.

[19] Cui L Y, Ma C W. SOF-SLAM: a semantic visual SLAM for dynamic environments[J]. IEEE Access, 2019, 7: 166528-166539.

[20] Zhang J, Henein M, Mahony R, et al. VDO-SLAM: a visual dynamic object-aware SLAM system[EB/OL]. (2020-05-22) [2021-02-05]. https://arxiv.org/abs/2005.11052.