DOI: 10.19650/j.cnki.cjsi.J2209856

动态场景下基于光流的语义 RGBD-SLAM 算法*

刘钰嵩[1],何 丽[1],袁 亮[2],齐继超[1]
(1. 新疆大学智能制造现代产业学院 乌鲁木齐 830047; 2. 北京化工大学信息科学与技术学院 北京 100029)

摘 要:为解决传统的同时定位与建图算法在复杂动态环境下容易受到动态目标干扰而导致定位精度差和建图错误的问题,提出了一种动态场景下基于光流的语义 RGBD-SLAM 算法。首先,通过优化的二维相邻帧透视矫正方法,对当前帧进行透视矫正以补偿相机运动;然后,将矫正后的图像输入 RAFT-S 网络中,在获得低分辨率的稠密光流场后提取动态目标的掩码,并根据上一帧掩码中动态目标的位置和速度信息,对当前掩码中的动态区域进行跟踪和优化,从而提取动态目标在每一帧中的精确区域;最后,分离静态和动态特征点,通过最小化静态特征点的重投影误差,得到优化后的相机位姿,并结合轻量级语义分割网络 BiSeNetv2 提供的语义信息和相机提供的深度信息,构建无人的静态语义八叉树地图。公开数据集 TUM 上的测试结果表明,本文算法的绝对轨迹误差相对于 ORB-SLAM2 减少了 90% 以上,并能获取精确的动态区域掩码以及准确的语义地图,验证了该算法在复杂动态场景中具有良好的定位精度和鲁棒性。

关键词:同时定位与建图;动态场景;光流;语义信息;八叉树地图
中图分类号:TP242.6 TH74 **文献标识码**:A **国家标准学科分类代码**:510.80

Semantic RGBD-SLAM in dynamic scene based on optical flow

Liu Yusong[1], He Li[1], Yuan Liang[2], Qi Jichao[1]
(*1. College of Intelligent Manufacturing and Modern Industry, Xinjiang University, Urumqi 830047, China;*
2. School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: To address the problems of poor positioning accuracy and mapping error in the traditional simultaneous localization and mapping algorithms under the complex dynamic environments with dynamic objects, a semantic RGBD-SLAM algorithm in dynamic scenes is proposed, which is based on the optical flow. Firstly, the camera ego-motion is compensated by the optimized 2D perspective correction method based on adjacent frames. Secondly, by feeding the compensated perspective images into the RIFT-S network, the low-resolution dense optical flow field is obtained for extracting the current mask of the dynamic region. The dynamic regions in the current mask are tracked and optimized by using the position and velocity of the dynamic regions in previous mask. The accurate dynamic regions in each frame can be extracted. Finally, the static and dynamic features are separated, and the optimized camera pose is obtained by minimizing the reprojection error of the static feature points. The static semantic octree map without people is established by the depth data from camera and semantic information produced by the lightweight semantic segmentation network BiSeNetv2. Compared with ORB-SLAM2, the test results on the public data set of TUM indicate that the absolute trajectory error of the proposed algorithm is reduced by more than 90%, and the accurate masks of dynamic regions and an accurate semantic map also can be obtained. Results show that the proposed algorithm has a good positioning accuracy and robustness under complex dynamic scenes.

Keywords: SLAM; dynamic environment; optical flow; semantic information; octree map

收稿日期:2022-05-26 Received Date: 2022-05-26
*基金项目:国家自然科学基金(62063033)项目资助

0 引 言

同时定位与建图(simultaneous location and mapping,SLAM)技术自提出以来[1],已经取得了极大的发展,成为了很多应用领域的基础技术,如:无人驾驶汽车(un-manned motor vehicle,UMV)、机器人技术和增强现实(augmented reality,AR)[2],尤其是基于视觉传感器的 SLAM 算法,不断出现了很多优秀的算法,如 PTAM[3]、LSD-SLAM[4]、ORB-SLAM[5]等。但大部分算法都采用了环境静态假设,即便通过如随机抽样一致性(random sample consensus,RANSAC)等鲁棒算法,传统算法在高动态环境下依然会有建图不稳定和漂移现象。而在现实的室内环境中,机器人是在低动态和高动态结合的复杂动态环境下运行的,因此提高机器人在复杂动态环境下的运行效果受到了研究人员的广泛关注[6]。

随着深度学习在图像处理领域的发展,有研究者将语义信息融入 SLAM 算法前端以识别可能移动的目标,Yu 等[7]提出的 DS-SLAM 通过 SegNet[8]语义分割网络提取先验动态目标再计算特征的光流,利用特征的极线约束剔除运动不一致的特征点。Bescos 等[9]提出的 DynaSLAM 通过 MaskR-CNN[10]分割图像后利用多视图几何根据深度差和角度差信息进一步分割图像。但是基于语义分割的方法在某些边角案例下无法识别场景中的动态目标,并且网络中先验的静态物体发生移动也会使 SLAM 算法失效,所以还需要用其他辅助方法,导致计算量增大。Soares 等[11]提出的 Crowd-SLAM 通过改进的 YOLO Tiny 目标检测网络提高了在拥挤的人群环境中识别人的精度,通过滤除包围框中的特征点获得稳定静态特征。但由于人的包围框面积过大,导致人占据视角过大的情况下也会导致跟踪丢失的情况,且无法检测出非先验的动态目标。

基于深度学习的光流算法在检测图像动态像素上有着巨大的优势。Zhang 等[12]提出的 FlowFusion 通过 PWC-Net[13]估计连续两帧光流,同时根据深度信息估计相机运动,然后通过投影获得场景流,通过计算动态目标分割然后经迭代后重建静态地图。Zhang 等[12]提出的 VDO-SLAM[14]通过 Mask R-CNN 实例分割以及 PWC-Net 光流网络提取图像中的移动目标,以此来将相机位姿、动态和静态特征点、运动物体的位姿联合计算与优化,通过这套估计框架可以处理由于遮挡导致的分割失败的情况,使算法鲁棒性更强。以上两种方法都利用了光流检测动态区域,并构建了三维空间的场景流来对动态目标进行跟踪,精度较高但系统计算量大,导致实时性不足。

为提高稠密光流 SLAM 算法的实时性和精度,Canovas 等[15]提出的具有高速和内存使用效率的动态环境下稠密 SLAM。采用对深度图和彩色图透视变换来减小单个透视的误差以此达到动态目标分割的准确性。使用高效的稠密逆搜索光流算法[16],并通过自适应阈值对超像素上的流场大小和深度差异做补偿,极大提升了稠密 SLAM 算法的实时性和在动态环境下的精度。但是由于直接采用 RANSAC 算法获得透视变换矩阵,并没有对透视矩阵的提取策略做优化,所以只能在静态场景占主导的环境下运行。

本文针对室内复杂动态环境,提出了一种动态场景下基于光流的语义 RGBD-SLAM 算法。首先,为精确提取动态区域,采用优化的透视矫正方法来补偿相机的自我运动,而后通过光流场提取当前帧的动态掩码并对动态区域进行跟踪和优化。然后,提取稳定的动态区域,通过稳定的静态特征提升估计位姿的精度。最后,通过结合轻量级的语义分割网络构建静态语义八叉树地图。

1 SLAM 算法的流程

1.1 算法框架

算法选择 ORB-SLAM2[17]作为主体框架,在原有的跟踪线程、局部建图线程、回环检测线程外增加了基于递归所有场变换(recurrent all-pairs field transforms, RAFT)光流神经网络[18]的动态区域的提取和优化线程,和基于轻量级的 BiSeNetV2[19]的语义建图线程,流程如图 1 所示。

首先,系统将深度相机获得的彩色图同时输入 ORB-SLAM2 的跟踪线程、语义建图线程和动态目标的提取和优化线程中。待提取完当前帧的 ORB 特征(oriented FAST and rotated BRIEF, ORB)后,动态区域的提取和优化线程会结合上一帧掩码跟踪结果计算透视变换矩阵,对当前帧进行透视矫正。之后输入 RAFT-S 网络提取下采样后的光流场,并结合上一帧动态区域跟踪结果完成对当前帧动态区域的优化和跟踪操作。随后,剔除当前帧属于动态目标区域的动态特征点,利用静态点估计位姿,将位姿信息和关键帧信息传送至局部建图线程以及回环检测线程,结合光束平差法(bundle adjustment, BA)对地图点和位姿进行进一步的优化。最后,语义建图线程会根据位姿信息和关键帧信息,结合深度图像和 BiSeNetv2 输出的语义信息,由八叉树生成算法将点云信息转换为带有语义标签的八叉树地图[20]。

1.2 动态特征剔除及位姿估计

1)特征提取

本文基于 ORB-SLAM2 通过提取 ORB 特征对相机位姿进行计算。ORB 特征通过快速提取 FAST 角点和

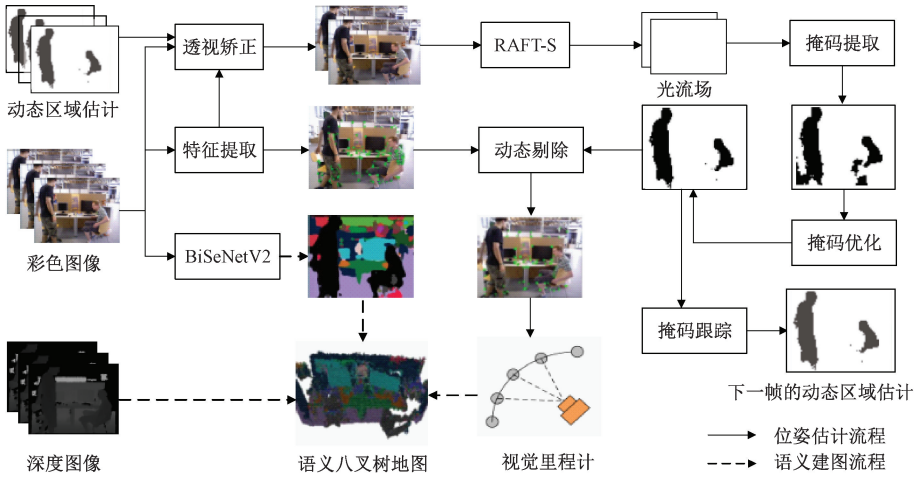


图 1　系统总体流程

Fig. 1　Overall system framework

BRIEF 描述子使得 ORB 特征在图像旋转、平移和缩放的情况下都有良好的表现。

2)相机运动补偿

由于相机在运动过程中,不可避免的会产生自我流,造成自我流和动态目标移动产生的光流叠加。本文算法通过建模单应矩阵 $\boldsymbol{H} \in SE(2)$ 作为二维透视的变换矩阵,如图 2 所示,设上标 k 为相机运行的当前时刻,通过对当前帧彩色图像 F^k 对上帧图像 F^{k-1} 估计单应变换 $\hat{H}^k$ 以补偿当前时刻相机自我运动 T^k,生成矫正后图像 F_w^k。F^k 中的黑色掩码 $\boldsymbol{D}_e^k$ 是由上一时刻优化后的动态实体的动态区域 $\hat{\boldsymbol{D}}_c^{k-1}$ 所估计而来。

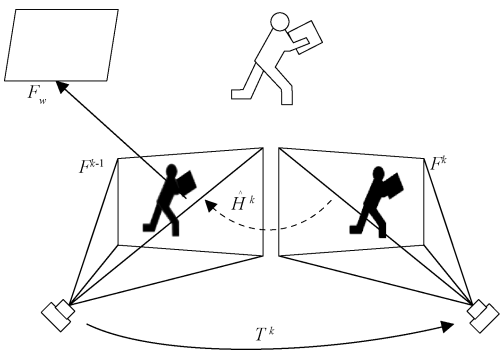


图 2　基于估计地静态特征求解单应矩阵

Fig. 2　Solving homography matrix by estimative static features

补偿相机运动的具体流程如下:

首先,选择非 $\boldsymbol{D}_e^k$ 区域内的特征点作为计算单应矩阵的点集。设 F^k 中二维特征点的集合为 $\boldsymbol{B}^k$,设 F^k 中有 n^k 个特征点构成的点集 $\boldsymbol{P}^k$:

$$\boldsymbol{P}^k = \{p_1^k, p_2^k, \cdots, p_{n^k}^k\} \not\subset \boldsymbol{D}_e^k \tag{1}$$

且 $\boldsymbol{P}^k \subseteq \boldsymbol{B}^k$,$\boldsymbol{P}^k$ 在 F^{k-1} 中有 n^k 个匹配点构成的点集 $\boldsymbol{P}^{k-1}$:

$$\boldsymbol{P}^{k-1} = \{'p_1^{k-1}, 'p_2^{k-1}, \cdots, 'p_{n^k}^{k-1}\} \not\subset \boldsymbol{D}_e^k \tag{2}$$

然而,由于采用单应矩阵平面透视,而真实环境空间关系更加复杂。因此算法采用最小中值法(least median of squares, LMedS)鲁棒算法遍历误差中值选取单应矩阵。相比于 RANSAC 算法,LMedS 算法可以减小通过平面透视复杂空间环境导致的光流场中的误差,提高动态掩码的精确度。

然后,LMeds 算法会对整个特征点集采样求误差中值。为了提高算法稳健性,设置 ε 表示正确匹配特征点置信度,设至少一次采样全是内点的概率 P_c^k 为:

$$1 - P_c^k = [1 - (1 - \varepsilon)^4]^{m^k} \tag{3}$$

其中,4 为每次采样点数,因为求解单应矩阵最少需要 4 对点;m^k 为采样一次全是内点的最小采样次数。可以通过给定 w 和 ε 的值进一步提高系统的鲁棒性。通过 m^k 次采样后可以得到至少一次采样全是内点。

因 LMedS 算法在有高斯噪声存在时效果较差,故采用加权最小二乘法以得到鲁棒的标准差估计:

$$\hat{\sigma} = 1.482\,6\left(1 + \frac{5}{n^k - 4}\right)\sqrt{M} \tag{4}$$

其中,常数 1.482 6 是在仅存在高斯噪声的情况下实现与最小二乘法相同效率的系数[21],$1 + 5/(n^k - 4)$ 是一个小样本修正因子,M 为整个点集的平方残差的中位数:

$$M = \underset{i=1,\cdots,n^k}{\operatorname{med}} r_i^2(\boldsymbol{H}, p_i^k) \tag{5}$$

其中,med 为取中值操作,r^2 为残差平方,$\boldsymbol{H}$ 为单应矩阵的估计值。之后对每一个 r_i^2 用式(6)进行筛选:

$$w_i = \begin{cases} 1, & r_i^2 \leqslant (2.5\hat{\sigma})^2 \\ 0, & \text{其他} \end{cases} \tag{6}$$

若 r_i^2 对应的 w_i 为0，则 r_i^2 为异常值，不进行下一步计算。通过对异常值的抛弃，可以稳健地构建出单应矩阵透视的最小二乘估计：

$$\hat{\boldsymbol{H}}^k = \arg\min_{\boldsymbol{H}} med_{i\in n^k} r_i^2(\boldsymbol{H}, p_i^k) \tag{7}$$

故 $\hat{\boldsymbol{H}}^k$ 即为 F^k 采用的单应矩阵，算法时间复杂度为 $O(n\log n)$。

而后，设 $\boldsymbol{P}^k$ 中点的坐标通过向量可以表示为 $\boldsymbol{p}_i^k = [x_i, y_i, 1]^{\mathrm{T}}, i = 1,2,\cdots,n^k$。可以得到矩阵 $\boldsymbol{P}_c^k = [\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, \cdots, \boldsymbol{p}_{n^k}^k]$，同理 $\boldsymbol{P}^{k-1}$ 中的点也可表示为 $\boldsymbol{P}_c^{k-1} = [\boldsymbol{p}_1^{k-1}, \boldsymbol{p}_2^{k-1}, \cdots, \boldsymbol{p}_{n^k}^{k-1}]$，和 $\hat{\boldsymbol{H}}^k$ 有对应关系为：

$$\boldsymbol{P}_c^{k-1} = \hat{\boldsymbol{H}}^k \boldsymbol{P}_c^k + \boldsymbol{\omega}^k \tag{8}$$

其中，$\boldsymbol{\omega}$ 为透视矫正后的误差项。根据式(8)求解 $\hat{\boldsymbol{H}}^k$：并设 $\hat{\boldsymbol{H}}^k$ 为：

$$\hat{\boldsymbol{H}}^k = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \tag{9}$$

最后，将 F^k 单应变换，得到当前帧的变换后图像 F_w^k 为：

$$F_w^k(x,y) = F^k\left(\frac{H_{11}x + H_{12}x + H_{13}}{H_{31}x + H_{32}y + H_{33}}, \frac{H_{21}x + H_{22}y + H_{23}}{H_{31}x + H_{32}y + H_{33}}\right) \tag{10}$$

故 F_w^k 即为补偿相机运动后的图像。

3)动态掩码优化

掩码优化算法的流程如图3所示，其中实线为当前掩码 M_c^k 的优化流程，虚线表示根据此时掩码估计 M_e^{k+1} 的流程。算法输入图像像素为640×480，将 F_w^k 和 F^{k-1} 输入RAFT-S中即可输出补偿相机自我运动后的水平位移场 f_u^k，和纵向位移场 f_v^k。

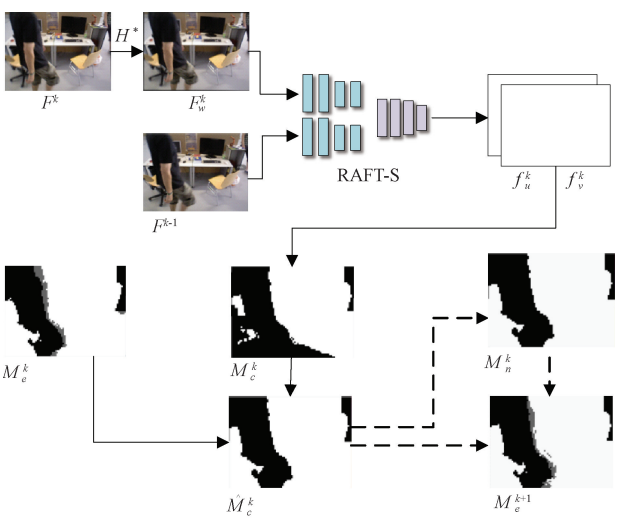


图3 掩码优化流程

Fig. 3 Mask optimization flowchart

RAFT-S是RAFT网络的轻量化版本，主要由特征编码器，一个相关层和一个基于GRU卷积层的循环更新器组成，并使用大量的更新模块来模拟一阶优化算法的步骤。RAFT-S通过Sintel[22]数据集训练，相对于传统的光流方法，它提取的特征子对稀疏纹理、光照等有更强的鲁棒性，上下采样的过程使得特征具有更大的感受野。RAFT-S相对与RAFT具有更少的权重参数和更快的输出速度，输出分辨率为RAFT的1/8，即80×60，因此 F_w^k 和 F^{k-1} 的关系可以表示为：

$$F_w^k(x,y) = F^{k-1}\left(x + 8\cdot \boldsymbol{f}_v^k\left(\frac{x}{8},\frac{y}{8}\right), y + 8\cdot \boldsymbol{f}_u^k\left(\frac{x}{8},\frac{y}{8}\right)\right) \tag{11}$$

通过对光流场强度设置阈值，即位移场的2个像素差，可以得到当前帧掩码 M_c^k 以及动态区域 $\boldsymbol{D}_c^k$：

$$M_c^k(x,y) = \begin{cases} 0, & |0.5\cdot \boldsymbol{f}_u^k(x,y)| > 1 \\ 0, & |0.5\cdot \boldsymbol{f}_v^k(x,y)| > 1 \\ 1, & \text{其他} \end{cases} \tag{12}$$

故 M_c^k 中为0的区域是估计的动态区域 $\boldsymbol{D}_c^k$。但是由于场景的深度误差、光流自身误差和相机运动过中图像模糊问题，光流场在相机运动时仍然会有大量噪声。为获得精确的动态目标区域，算法根据相邻帧中光流位移场中的像素的速度信息和位置信息，优化动态掩码，时间复杂度为 $O(n^2)$。通过对移动目标区域追踪，以减少因为光流误差导致错误被划为动态的静态区域，提高动态掩码对移动物体覆盖的精度和稳定性，最终提高定位精度。

M_c^k 中动态区域的优化流程为，若 $\boldsymbol{D}_c^k$ 大于800 pixel，则先结合当前的 M_e^k 中的估计的动态区域 $\boldsymbol{D}_e^k$ 与 $\boldsymbol{D}_c^k$，$\boldsymbol{D}_e^k \cap \boldsymbol{D}_c^k = \hat{\boldsymbol{D}}_c^k$ 即为优化后的动态区域，并得到优化后掩码 $\hat{M}_c^k$。而下一帧的 M_n^k 是通过 $\hat{M}_c^k$ 的动态区域 $\hat{\boldsymbol{D}}_c^k$ 结合 $\boldsymbol{f}_v^k$，$\boldsymbol{f}_u^k$ 的位移信息估计出的掩码。$\boldsymbol{D}_n^k$ 是以 $\hat{\boldsymbol{D}}_c^k$ 中像素在位移场的移动速度，移动两个时间间隔后的目标位置。$\boldsymbol{D}_n^k \cup \hat{\boldsymbol{D}}_c^k$ 即为更新后的估计的下一帧动态区域 $\boldsymbol{D}_e^{k+1}$。

考虑到移动目标的出现也可以瞬时由静到动的，为保证掩码的实效性和连续性，算法限制了优化进程的输入条件。若 $\boldsymbol{D}_c^k$ 小于800 pixel，则直接有 $\hat{M}_c^k = M_c^k$。得到 $\boldsymbol{D}_e^{k+1}$ 后，将在计算 $\hat{\boldsymbol{H}}^{k+1}$ 时提供约束，即 F^{k+1} 特征点集合 $\boldsymbol{P}_c^{k+1}$ 中，将不包含 $\boldsymbol{D}_e^{k+1}$ 区域内的特征点。

对掩码进行持续优化示例，如图4所示为TUM数据集中从第 n 帧开始的连续3帧图像以及对应的掩码。其中 M_e^k 中灰色部分为估计出的属于下一帧的动态区域 $\boldsymbol{D}_n^k$ 但不包含于 $\boldsymbol{D}_c^k$ 的区域，$\boldsymbol{f}c^k$ 为第 k 帧光流场的可视化图像。通过观察 F^{n+2} 中实际的动态区域，M_c^{n+2} 圆圈处有错

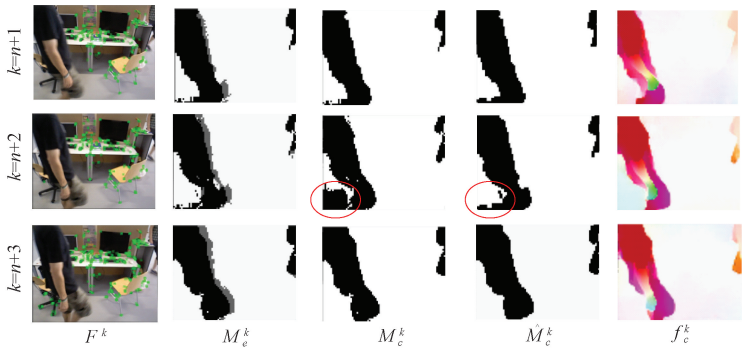


图 4　动态区域追踪和优化

Fig. 4　Dynamic area tracking and optimization

误划分的区域。经算法优化后，$\hat{M}_c^{n+2}$ 中只保留了稳定的动态区域，并估计第 $n+3$ 帧中动态目标的稳定范围 M_e^{n+3}。

4)基于静态特征的位姿估计

通过剔除带有动态掩码标识的动态特征点，跟踪线程根据保留的静态特征点求取位姿。设 $\boldsymbol{F}^k$ 中静态二维特征点集合为 $\boldsymbol{U}^k$，有 $\boldsymbol{U}^k \cap \hat{\boldsymbol{D}}^k{}_c = \varphi$，且其在三维空间的点集合为 $\boldsymbol{C}^k$。三维坐标点 $\boldsymbol{c}_i = [X_i, Y_i, Z_i]^{\mathrm{T}}$ 和其对应的投影像素坐标为 $\boldsymbol{u}_i = [u_i, v_i]^{\mathrm{T}}$，并且 $\boldsymbol{c}_i \in \boldsymbol{C}^k$，$\boldsymbol{u}_i \in \boldsymbol{U}^k$，两者之间的对应关系为：

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{K}\exp(\boldsymbol{\xi}_i^{\wedge}) \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} + \boldsymbol{e}_i \tag{13}$$

其中，S_i 为对应特征点深度，$\boldsymbol{K}$ 为相机内参矩阵，$\xi \in se(3)$ 为相机位姿 $\boldsymbol{R}, \boldsymbol{t}$ 的李代数形式，$\boldsymbol{e}$ 为误差项。写成矩阵形式为：

$$s_i \boldsymbol{u}_i = \boldsymbol{K}\exp(\boldsymbol{\xi}_i^{\wedge})\boldsymbol{c}_i + \boldsymbol{e}_i \tag{14}$$

将误差求和构建最小二乘问题并使它最小化，即最小化重投影误差：

$$\hat{\boldsymbol{\xi}}^k = \arg\min_{\xi} \frac{1}{2}\sum_{i=1}^{n} \left\| \boldsymbol{u}_i - \frac{1}{s_i}\boldsymbol{K}\exp(\boldsymbol{\xi}^{\wedge})\boldsymbol{c}_i \right\|_2^2 \tag{15}$$

其中，n 为静态特征点数，$\hat{\xi}^k$ 故即为动态环境下最优的相机位姿。

1.3　静态八叉树地图构建

静态八叉树地图[23]是一种易于压缩的和更新的地图类型。在语义建图线程中，本文使用 BiSeNetv2 语义分割网络，采用 COCO-stuff 数据集训练，其使用双路径结构，通过语义分支和细节分支可以用更少的通道和快速下采样策略实现更高的效率。相较于文献[20]中使用的 PSPnet[24]，本文算法建图线程提高约一倍的地图更新速度。

算法首先将不同时刻的位姿和深度图像进行融合生成不同时刻相机位姿下的三维点云图，并且将语义信息也存储在点云地图中；然后通过下采样过滤点云后将剩下的点插入到八叉树中；最后通过更新节点的概率来对八叉树地图进行可视化。

八叉树节点分为占据和空白两种属性，假设 $t = 1, \cdots, T$ 时刻，观测的数据为 z，某个节点为 n，那么第 n 个叶子节点的概率 P_n 为：

$$P_n(n \mid z_{1:T}) = \left[1 + \frac{1 - P(n \mid z_T)}{P(n \mid z_T)} \frac{1 - P(n \mid z_{1:T} - 1)}{P(n \mid z_{1:T} - 1)} \frac{P(n)}{1 - P(n)}\right]^{-1} \tag{16}$$

算法采用贝叶斯融合算法融合多视图的语义信息提高地图语义标签精度，并在保留原有背景的语义信息同时剔除人标签节点，构建仅有静态背景的语义地图。

2　实验与分析

实验采用 TUM 公开数据集[25]以及在实际室内环境下通过手持 KinectV1 深度相机进行测试。TUM 数据集是慕尼黑工业大学录制的室内场景下的公开的标准数据集，用来评价 SLAM 算法在室内环境下鲁棒性和稳定性的经典数据集，共包含 39 个不同的序列，拥有彩色图、深度图以及通过高精度传感器测量的真实相机位姿。数据集中 w(walking)代表由人物行走的高动态数据集序列，s(sitting)代表低动态数据集序列，而 xyz、halfsphere、rpy 和 static 代表数据集中相机的移动模式，分别为沿主轴移动、沿半球移动、沿主轴翻转、和几乎静止。算法运行环境为：Ubuntu16.04 操作系统，ROS 和 CUDA-11.3，CPU 为 Intel i7-1050，显卡为 NVIDIA RTX2060。

2.1　动态环境下定位精度分析

本文算法使用绝对轨迹误差(absolute trajectory error, ATE)来评估系统的整体性能，使用相对位姿误差

(relative pose error, RPE)来评估里程计漂移情况。通过对比不同算法并采用均方根误差(root mean square error, RMSE)作为评价的指标[26]。

ORB-SLAM2 与本文算法在 w-xyz 和 w-halfsphere 的轨迹的误差如图 5 所示。其中带方框的线为真实轨迹，带三角的线为估计轨迹，实线为真实轨迹和估计轨迹间的误差。ORB_SLAM2 由于没有复杂动态环境处理机制在开始阶段就有明显的轨迹偏移，并且随时间其误差不断增大。但本文算法的鲁棒机制可以在整个轨迹段内都可以保持轨迹稳定。

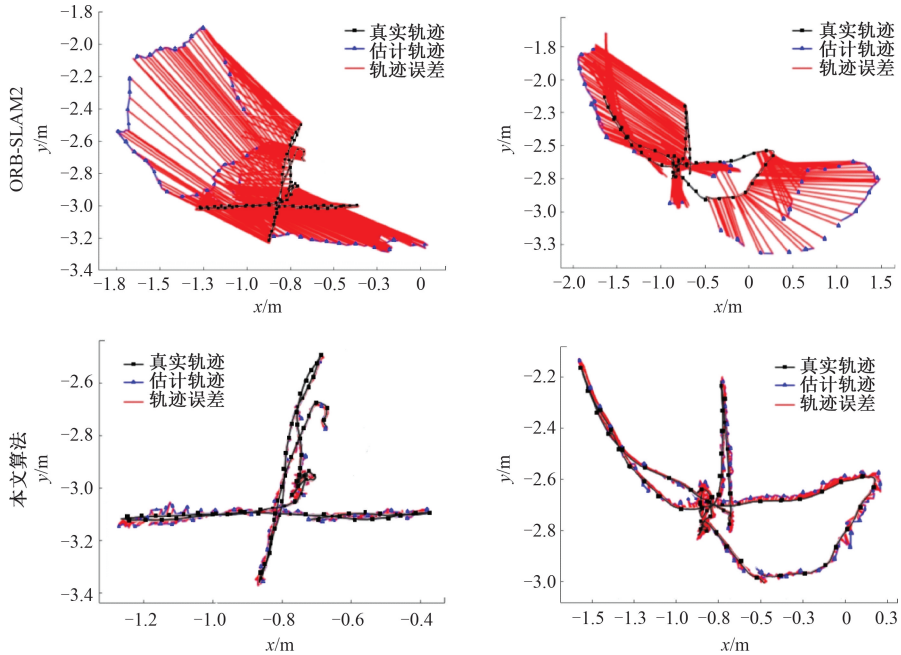


图 5 高动态环境下的轨迹对比

Fig. 5 Comparison of estimated trajectory and ground-truth between ORB-SLAM2 and our algorithm

本文算法与其他动态 SLAM 算法绝对轨迹误差如表 1 所示，本文算法在多个数据集的表现要优于基于语义分割网络的动态 SLAM 算法，如 Sad-SLAM 和 DS-SLAM。并在低动态数据集下没有出现精度下降，表现出了在低动态和高动态复杂环境下的适应性。

表 1 不同算法绝对轨迹误差对比

Table 1 Comparison of the absolute trajectory error among different algorithms m

序列	ORB-SLAM2	DS-SLAM	文献[15]	文献[27]	Detect-SLAM[28]	Sad-SLAM[29]	本文算法
w_xyz	0.770 3	0.024 7	0.021 0	0.019 4	0.024 1	0.016 7	**0.015 5**
w_static	0.425 4	0.008 1	-	0.011 1	-	0.016 6	**0.007 8**
w_rpy	0.829 9	0.444 2	-	0.037 1	0.295 9	**0.031 8**	0.060 1
w_half	0.601 0	0.030 3	**0.016 7**	0.029 0	0.051 4	0.025 7	0.026 9
s_xyz	0.009 5	-	**0.004 5**	0.011 7	0.020 1	0.012 4	0.008 6
s_static	0.008 7	0.006 5	-	-	-	**0.006 0**	0.008 5
s_rpy	0.019 7	-	-	-	-	0.028 8	**0.017 8**
s_half	0.021 1	-	-	0.017 2	0.023 1	0.015 1	**0.012 8**

相对位姿误差通过规定时间差来描述位姿精度，本文算法与文献[27]算法相对平移误差如表 2 所示，相对旋转误差如表 3 所示，均选择 1 s 时间差作为单位。本文算法在 3 个数据集中都更优，但在 w-rpy 数据集中，由于 20~25 s 中错误的局部地图点与当前特征点匹配，导致局部建图过程中出现了错误的优化，使得相对于其他数据集精度较低。综上所示，本文算法在室内动态环境具有较强的优势。

表 2　相对平移漂移误差对比

Table 2　Comparison of PRE in translational drift

(m/s)

序列	ORB-SLAM2	文献[27]	本文算法
w_xyz	0.412 4	0.023 4	**0.020 2**
w_static	0.216 2	0.011 7	**0.009 7**
w_rpy	0.424 9	**0.047 1**	0.066 8
w_half	0.355 0	0.042 3	**0.029 1**

表 3　相对旋转漂移误差对比

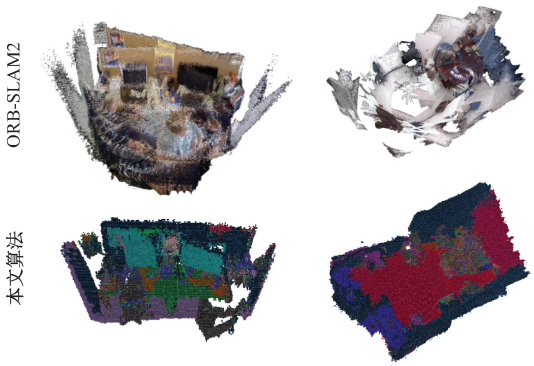
Table 3　Comparison of PRE in rotational drift

(°/s)

序列	ORB-SLAM2	文献[27]	本文算法
w_xyz	7.743 2	0.636 8	**0.599 9**
w_static	3.895 8	0.287 2	**0.254 8**
w_rpy	8.080 2	**1.058 7**	1.372 8
w_half	7.374 4	0.965 0	**0.720 5**

2.2　动态环境下三维语义地图构建结果对比

通过算法与 ORB_SLAM2 分别在 TUM 数据集和实际室内环境下测试以验证复杂动态环境下的建图效果。如图 6 所示,其中图 6(a)为 TUM 数据集的 w-static 序列,通过比较表明 ORB_SLAM2 在建图时由于里程计漂移导致明显建图错误,而本文算法在有动态人物的情况下精确构建了背景静态地图。图 6(b)为手持相机在封闭的办公室场景下构建的室内地图,动态人物在建图的过程中随机往返。通过封闭场所的实际建图实验验证,本文算法能在剔除动态目标的同时构建高精度静态地图,并且给予地图准确的语义信息。



(a) W-static序列　(b) 办公室场景

(a) W-static sequence　(b) Office scene

图 6　高动态场景下建图效果对比

Fig. 6　Mapping comparison in high dynamic scenes

2.3　复杂动态环境下掩码提取实验

为验证算法在复杂人群中的定位和建图效果,本文设置了低动态和高动态组合的复杂人群环境以及动态目标对相机视野大部分遮挡的环境,同时通过手持深度相机进行实验。低动态和高动态组合的复杂人群环境实验中有代表性的部分帧如图 7(a)和其对应的动态掩码 $\hat{M}_c^k$ 如图 7(b)所示。设置此实验的目的是当人或人群占图像面积过大的环境中,基于语义进行动态目标识别的 SLAM 方法如 Crowd-SLAM、Sad-SLAM 容易在此环境下失效,但本文算法可以精确的划分动态和静态区域并且不受动态目标个数和位置影响,同时建立语义静态地图,如图 8(b)所示。

(a) 剔除动态特征点　(b) 当前帧动态掩码

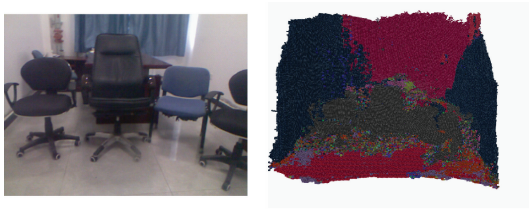
(a) Remove dynamic features　(b) Current frame dynamic mask

图 7　复杂人群环境下定位和建图

Fig. 7　SLAM in complex crowd environment

动态目标对相机视野大部分遮挡的环境中设置了移动目标通常为通常为静态的物品的非先验的语义对象,如纸箱等。由于相机距离较近,导致输入图像中,动态区域占据了图像的绝大部分面积的情况,如图 9 所示。

因本文算法优化了动态掩码的提取策略,相较于文献[15]使用了 RANSAC 作为提取透视矩阵的鲁棒算法,本文算法不需要设置静态面积占据主要区域的假设。经



(a) 真实环境
(a) Real environment

(b) 语义地图
(b) Semantic map

图 8 真实环境和语义地图

Fig. 8 Static scene and semantic octree map

(a) 剔除动态特征点
(a) Remove dynamic features

(b) 当前帧动态掩码
(b) Current frame dynamic mask

图 9 复杂动态环境下定位

Fig. 9 Positioning in complex dynamic environment

验证,算法可以实现在动态区域占据图像 4/5 以下面积,准确地提取动态区域,如图 9(b)所示。并且动态目标速度不限,在满足最少 20 个匹配点对情况下,算法就能在位姿不丢失的情况下实现鲁棒定位。

2.4 实时性评估

通过各模块运行时间评估算法实时性,如表 4 所示。运动补偿及光流计算和动态掩码提取及优化同线程运行,位姿跟踪及优化并行运行。语义建图线程不参与位姿估计,算法其他线程单帧平均运行时间约 160 ms。故本文算法的位姿鲁棒估计基本满足实时性的要求。

表 4 平均运行时间

Table 4 Average running time ms

场景	模块		
	运动补偿及光流计算	动态掩码提取及优化	位姿跟踪及优化
w_static	129	10	98
w_xyz	154	11	138
真实场景	139	13	79

3 结 论

本文针对室内复杂动态环境,提出了一种动态场景下基于光流的语义 RGBD-SLAM 算法。算法首先通过优化透视矩阵的提取策略矫正相机自我运动。然后结合 RAFT-S 提取实时的低分辨率光流场,通过相邻帧掩码的优化,提高了动态掩码质量和效率,并大幅提高了算法在复杂环境下的鲁棒性和精度。最后,使用 BiSeNetV2 轻量级网络提高了语义分割速度,使用语义八叉树地图减少了地图容量,在提高了建图效率的同时增加了地图的可读性。在 TUM 数据集和实际建图的验证中可以看出,本文算法在高动态的室内环境下有高精度和高鲁棒性。但是语义信息并没有利用到定位过程中,在以后的工作中,将考虑语义信息利用于辅助定位。

参考文献

[1] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on Robotics, 2016, 32(6): 1309-1332.

[2] 潘林豪,田福庆,应文健,等. 单目相机-IMU 外参自动标定与在线估计的视觉-惯导 SLAM[J]. 仪器仪表学报, 2019,40(6):56-67.
PAN L H, TIAN F Q, YING W J, et al. A multi-focal length dynamic stereo vision SLAM[J]. Chinese Journal of Scientific Instrument, 2019,40(6):56-67.

[3] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces [C]. IEEE International Symposium on Mixed and Augmented Reality, Japan: IEEE, 2007: 1-10.

[4] CARUSO D, ENGEL J, CREMERS D. Large-scale direct SLAM for omnidirectional cameras[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany: IEEE, 2015: 141-148.

[5] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: A versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.

[6] 冯明驰,刘景林,李成南,等. 一种多焦距动态立体视

觉 SLAM[J]. 仪器仪表学报, 2021,42(11): 200-209.
FENG M CH, LIU J L, LI CH N, et al. A multi-focal length dynamic stereo vision SLAM[J]. Chinese Journal of Scientific Instrument, 2021,42(11): 200-209.

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

[8] 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.

[9] 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.

[10] HE K M, GKIOXARI G, DOLL′AR P, et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386-397.

[11] SOARES J C V, GATTASS M, MEGGIOLARO M A. Crowd-SLAM: Visual SLAM towards crowded environments using object detection [J]. Journal of Intelligent & Robotic Systems, 2021, 102(2): 1001-1010.

[12] ZHANG T W, ZHANG H Y, LI Y, et al. FlowFusion: Dynamic dense RGB-D SLAM based on optical flow[C]. IEEE International Conference on Robotics and Automation (ICRA), Piscataway, USA: IEEE, 2020: 7322-7328.

[13] SUN D Q, YANG X D, LIU M Y, et al. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Piscataway, USA: IEEE, 2018.

[14] ZHANG J, HENEIN M, MAHONY R, et al. VDO-SLAM: A visual dynamic object-aware SLAM system[J]. ArXiv Preprint, 2020, ArXiv: 2005.11052.

[15] CANOVAS B, ROMBAUT M, NEGRE A, et al. Speed and memory efficient dense RGB-D SLAM in dynamic scenes [C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA: IEEE, 2020: 4996-5001.

[16] KROEGER T, TIMOFTE R, DAI D, et al. Fast optical flow using dense inverse search [C]. European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, Springer , 2016, 9908: 471-488.

[17] MUR-ARTAL R, TARD′OS 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.

[18] TEED Z, DENG J. RAFT: Recurrent all-Pairs field transforms for optical flow[C]. European Conference on Computer Vision(ECCV), 2020: 402-419.

[19] YU C, GAO C, WANG J, et al. BiSeNetV2: Bilateral network with guided aggregation for real-time semantic segmentation [J]. International Journal of Computer Vision, 2021, 129(11): 3051-3068.

[20] XUAN Z. Real-time voxel based 3D semantic mapping with a hand held RGB-D camera [EB/OL]. (2019-09-25) [2022-12-13]. https://github.com/floatLazer/semantic_slam.

[21] ROUSSEEUW P J. Least median of squares regression[J]. Journal of the American Statistical Association, 1984, 79(388): 871-880.

[22] BUTLER D J, WULFF J, STANLEY G B, et al. MPI-Sintel optical flow benchmark: Supplemental material[M]. MPI-IS-TR-006, MPI for Intelligent Systems, Citeseer, 2012.

[23] HORNUNG A, WURM K M, BENNEWITZ M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots, 2013, 34(3): 189-206.

[24] ZHAO H, SHI J, QI X, et al. Pyramid scene parsing

network[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, 2017: 6230-6239.

[25] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems, Piscataway (IROS), USA: IEEE, 2012: 573-580.

[26] PROKHOROV D, ZHUKOV D, BARINOVA O, et al. Measuring robustness of visual SLAM[C]. 2019 16th International Conference on Machine Vision Applications (MVA), IEEE, 2019: 1-6.

[27] JI T, WANG C, XIE L. Towards real-time semantic RGB-D SLAM in dynamic environments[C]. IEEE International Conference on Robotics and Automation, IEEE, 2021: 11175-11181.

[28] ZHONG F, WANG S, ZHANG Z, et al. Detect-SLAM: Making object detection and SLAM mutually beneficial[C]. IEEE Winter Conference on Applications of Computer Vision, IEEE, 2018: 1001-1010.

[29] YUAN X, CHEN S. SaD-SLAM: A visual SLAM based on semantic and depth information[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), IEEE, 2020: 4930-4935.

作者简介

刘钰嵩,2019 年于西安科技大学获得学士学位,现为新疆大学硕士研究生,主要研究方向为视觉语义 SLAM。

E-mail:756346842@qq.com

Liu Yusong received his B. Sc. degree in 2019 from Xi′an University of Science and Technology. He is currently a master student in College of Xinjiang University. His main research interests include visual semantic SLAM.

何丽(通信作者),2008 年于河北科技大学获得学士学位,2013 年于新疆大学获得博士学位,现为新疆大学副教授,主要研究方向为移动机器人模式识别与智能控制技术。

E-mail:xju_heli@163.com

He Li (Corresponding author) received her B. Sc. degree in 2008 from Hebei University of Science and Technology, received her Ph. D. degree in 2013 from Xinjiang University. Now she is associate professor in Xinjiang University. Her main research interests include pattern recognition and intelligent control technology for mobile robots.