DOI: 10.19650/j.cnki.cjsi.J2311212

动态场景下基于实例分割和三维重建的多物体单目 SLAM*

冯 洲,续欣莹,郑宇轩,程 兰,李鹏越
(太原理工大学电气与动力工程学院 太原 030024)

摘 要:针对大多数 SLAM 系统在动态环境下相机位姿估计不准确与环境语义信息利用不充分的问题,提出一种基于实例分割的关键帧检测和贝叶斯动态特征概率传播的动态物体检测算法,并对环境中存在的静态物体三维重建,以此构建一个动态环境下的多物体单目 SLAM 系统。该系统对关键帧输入图像进行实例分割与特征提取,获取潜在运动物体特征点集合与静态物体特征点集合;利用非运动物体特征点集合获取帧间位姿变换,普通帧利用贝叶斯对动静态特征点进行概率传播,利用静态特征点集实现对相机位姿的精准估计;在关键帧中对静态物体进行联合数据关联,数据充足后进行多物体三维重建,构建多物体语义地图,最终实现多物体单目 SLAM。本文在 TUM 与 Boon 公开数据集上的实验结果表明,在动态场景下,相较于 ORB-SLAM2 算法,绝对位姿误差的均方根误差平均降低 54.1% 和 58.2%。

关键词: 多物体单目 SLAM;动态场景;实例分割;位姿估计;三维重建
中图分类号: TP242 TH74 **文献标识码**: A **国家标准学科分类代码**: 510.80

Multi-object monocular SLAM based on instance segmentation and 3D reconstruction in dynamic scene

Feng Zhou, Xu Xinying, Zheng Yuxuan, Cheng Lan, Li Pengyue
(*College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan 030024, China*)

Abstract: To address the problems of inaccurate camera pose estimation and insufficient utilization of environmental semantic information in most SLAM systems in dynamic environments, proposes a dynamic object detection algorithm based on the instance segmentation, keyframe detection, and Bayesian dynamic feature probability propagation, and three-dimensional reconstruction of static objects in the environment. To construct a multi object monocular SLAM system in a dynamic environment, the system performs instance segmentation and feature extraction on key frame input images, which could obtain a set of potential moving object feature points and a set of static object feature points. A set of non-moving object feature points is used to obtain inter frame pose transformation, Bayesian probability propagation of dynamic and static feature points are utilized for ordinary frames, and a set of static feature points is used to achieve accurate estimation of camera pose. Joint data association is performed on static objects in key frames, and after sufficient data is available, multi object 3D reconstruction is performed to construct a multi object semantic map. Finally, multi object monocular SLAM is achieved. The experimental results on TUM and Boon public dataset show that in dynamic scenarios, compared to the ORB-SLAM2 algorithm, the RMSE of APE decreases by 54.1% and 58.2% on average.

Keywords: multi-object monocular SLAM; dynamic scene; instance segmentation; posture estimation; three-dimensional reconstruction

收稿日期:2023-03-24 Received Date: 2023-03-24
*基金项目:国家自然科学基金(62073232)、国家青年科学基金(62203319)、虚拟现实技术与系统国家重点实验室(北京航空航天大学)开放课题基金(VRLAB2023A06)、山西省科技合作交流专项(202104041101030)资助

0 引 言

同步定位与建图(simultaneous localization and mapping, SLAM)是移动机器人领域的热点研究,被广泛应用于增强现实(Augmented Reality, AR)、虚拟现实(Virtual Reality, VR)、无人驾驶、无人机等领域。SLAM 使用的传感器主要有相机、激光雷达等。其中,相机具有低成本,信息丰富的优点,迄今为止,基于相机传感器的 SLAM 技术一直是移动机器人领域的研究热点。

Mur-Artal 等[1]提出的 ORB-SLAM2 是基于特征点的 SLAM 算法,可在大规模场景下长期运行,支持单/双目、RGB-D 相机,可实现地图重用、回环检测和重定位。Campos 等[2]在 ORB-SLAM2 的基础上提出的 ORB-SLAM3 系统,增加了多地图融合、视觉惯性里程计等功能并支持针孔相机和鱼眼相机。对于大多数 SLAM 系统都存在静态环境的前提假设,即假定机器人所处环境为静态,但这种假设在真实环境中往往难以成立。当环境中存在动态物体时,例如移动的行人与动物、行驶的车辆等,由于特征点分布在动态物体上,会导致 SLAM 系统在位姿估计时产生轨迹偏差,甚至导致定位漂移、跟踪丢失、系统崩溃等问题,严重影响系统的定位精度及稳定性[3]。针对动态场景下视觉 SLAM 定位问题,改进方法主要归纳为两类:一类是基于传统多视图几何及其改进方法[4];另一类是利用语义先验信息从背景分割对象以区分动态点和静态点的方法[5]。

传统多视图几何方法主要通过多帧图像的位姿约束,剔除误差较大的特征点。Dai 等[6]利用 Delaunay 三角剖分方法将序列图像中的特征点连结成多个三角形,顶点表示地图点,边表示相邻点的相关性,通过比较相邻帧三角形变化情况来判断特征点是否属于同一目标,将动态目标进行剔除,并将其嵌入到 ORB-SLAM2 系统的前端。Li 等[4]提出深度边缘点静态加权方法,采用静态权重表示特征点属于静态环境的概率,以此剔除关键帧中的动态特征点。Sun 等[7]基于相机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图像帧,并将其集成到 RGB-D SLAM 的前端。但当环境中运动物体移动较快或占据图像比例较大时,传统多视图几何方法会失效。

近几年随着深度学习的发展,利用图像的语义先验信息从背景分割对象以区分动态点与静态点的方法应运而生。DynaSLAM[5]在 RGB-D 模式下采用 Mask R-CNN(region-convolutional neural network)实例分割网络[8]和多视图几何方法结合筛选出动态区域,去除图像帧中的动态特征点,在单目模式下仅用 Mask R-CNN 剔除动态特征点,但由于 Mask R-CNN 算法运行频率仅能达到 5 Hz,故无法实现系统实时运行。Detect-SLAM[9]在 ORB-SLAM2 系统基础上结合 SSD(single shot multibox detector)目标检测神经网络[10]检测物体,通过特征匹配和增加影响区域的方式实现运动概率的传播,以此降低动态特征点产生的影响。但 SSD 网络检测一帧需要 310 ms,也难以实时运行。针对实例分割网络难以满足 SLAM 系统实时运行的问题,部分学者采用目标检测获取环境语义信息,但该方法存在静态特征点被误剔除的问题。Yang 等[11]将 YOLOv3 目标检测网络[12]应用于 ORB-SLAM2 的 RGB-D 模式进行高动态目标的检测与剔除,通过修改 ORB-SLAM2 系统框架下的特征提取环节,有效改善了运动模糊对特征匹配的影响,但其运行速度只能达到 9 Hz。DS-SLAM[13]在 ORB-SLAM2 的基础上,使用 SegNet 语义分割网络[14]获取图像中的语义信息,且为解决时间问题将语义网络单独放入一个线程中,结合运动一致性检测滤除掉每一帧图像中的动态特征点,以此提高系统在动态场景中的定位精度,但其与 Detect-SLAM 一样也难以实时运行。刘钰嵩等[15]融合光流法与实例分割算法剔除动态物体,但运行时间也仅为 160 ms/帧,难以实时运行。

对于环境中存在的行人、车辆等动态物体,当前主要有两种形式,第 1 种即上文所说,被视为异常值并从 SLAM 位姿估计部分中剔除,第 2 种为检测到动态目标后,采用多目标跟踪方法对其进行单独跟踪,多采用双目相机对移动的车辆追踪。DynaSLAM II[16]、VDO-SLAM[17]、DOT[18],三者系统均基于双目视觉,首先对环境实例分割获取语义信息,仅使用环境中的静态特征点估计相机位姿,对环境中存在的动态车辆追踪并估计其速度。DyOb-SLAM[19]首先利用实例分割算法获取动态车辆,通过使用光流和场景流算法对运动目标进行跟踪,并构建动态物体轨迹全局图。对移动车辆追踪的形式大多应用于自动驾驶场景,多采用双目相机获取点云精准位置,其计算量相对于单目相机较大。

综上所述,当环境中动态物体占据图像区域较大或速度较快时,系统无法获得可靠的观测数据,基于传统多视图几何的方法会失效[20];通过语义先验信息区分动静态物体的方法因实例分割网络无法实现实时检测,且会忽略部分语义先验信息(仅剔除先验动态的物体)。因此,针对基于实例分割方式剔除环境动态特征点无法实时运行的问题与实例分割后环境语义信息使用不充分的问题,本文在 ORB-SLAM2[1]基础上,提出一种动态场景下基于实例分割和三维重建的多物体单目 SLAM 系统。主要贡献如下:

1)将 ORB-SLAM2 系统与 SparseInst[21]实例分割网络相结合,检测并滤除掉关键图像帧上动态物体掩膜内的特征点,利用静态特征点估计相机位姿;

2)基于静态特征点估计相机位姿,并利用贝叶斯概率传播,将动静态特征点移动概率传播到普通图像帧中,同时对特征点进行移动概率更新;

3)通过结合 DeepSDF[22] 三维重建网络,对环境中存在的物体进行特征数据关联,并在数据充足后进行多物体三维重建。

1　本文系统

1.1　SLAM 系统框架

基于特征点的 SLAM 算法采用特征匹配的方式估计相机位姿,但在特征提取时会提取到图像帧中所有的特征点,这将包含静态特征点与动态特征点,而动态特征点会对 SLAM 算法鲁棒性产生极大影响。如何检测并剔除动态特征点是其中关键研究问题。应用实例分割算法分割环境中存在的动态物体与静态物体时,该方法存在两个问题:1)实例分割算法应用于 SLAM 系统,难以实时运行;2)实例分割后仅剔除环境中存在的动态物体,对静态物体的利用不充分,即环境语义信息使用不充分的问题。为解决上述问题,本文提出动态场景下基于实例分割和三维重建的多物体单目 SLAM 算法,以提升系统在动态环境下的鲁棒性与环境地图的语义性。

本文提出的系统如图 1 所示,采用单目相机作为传感器。该系统基于开源系统 ORB-SLAM2[1],实线框内为 ORB-SLAM2 原有线程,实线框内标黄部分为本文所做改进。虚线框内为本文所增加的运动区域检测、动态概率传播与多物体三维重建。

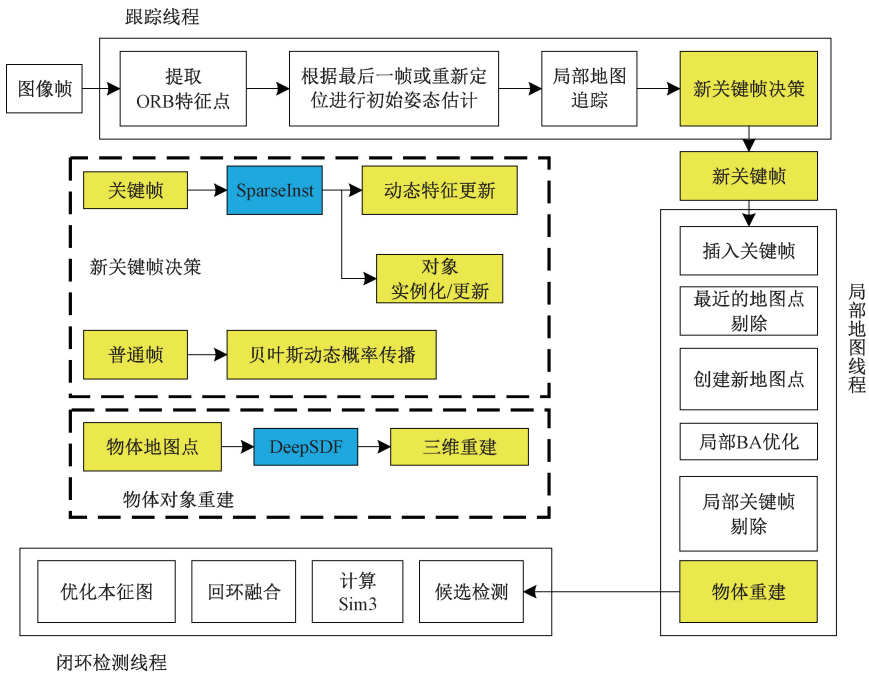


图 1　系统框架图

Fig. 1　System framework

系统处理流程为:1)初始位姿估计:当新一帧图像传入 Tracking 线程后,使用 ORB-SLAM2 系统原有特征检测与位姿估计得到当前图像帧系统初始位姿;2)动/静态物体分割提取:为确保系统时效性,本文采用关键帧检测策略,即仅在关键图像帧中进行实例分割,得到动态物体掩膜与静态三维重建物体掩膜集合;3)普通帧:通过上一帧静态特征点估计当前帧相机位姿,并将上一帧中动静态特征点使用贝叶斯概率传播进行更新;4)关键帧:将图像传入实例分割网络中,并将检测结果分为动态图像掩膜与多静态物体掩膜集合。将动态物体掩膜内特征点标记动态,其余特征点标记为静态,并初始化特征点移动概率;5)物体实例化与数据关联:根据关键帧中静态物体检测结果,对不同物体分别进行物体对象实例化与数据关联。

在 Local Mapping 线程中,根据观测到的物体数据判断,物体观测数据充足时系统会对其三维重建,即将关联的物体数据传入 DeepSDF 三维重建网络估计其隐式编码与优化后的物体位姿,最后对当前图像帧所有符合重建要求的物体进行三维重建与物体更新。

1.2　基于实例分割的动态区域检测

为得到图像中物体像素级语义标签,本文采用 Cheng 等[21] 于 2022 年提出的一种概念新颖、高效且完全

卷积的实时实例分割网络 SparseInst，采用 MS COCO 数据集训练。该网络以一组稀疏实例激活图（sparse instance activation map，IAM）作为新对象的表示，以此突出每个前景对象的信息区域。根据高亮区域聚合特征得到实例级特征，进行识别与分割。SparseInst 网络结构图如图 2 所示。

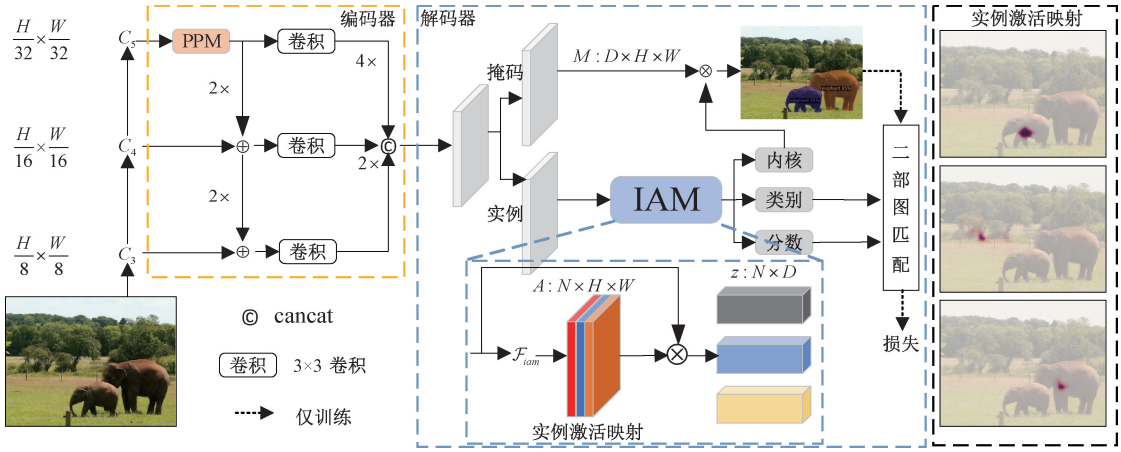
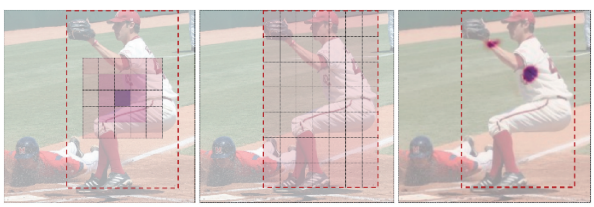


图 2 SparseInst 网络结构框图

Fig. 2 SparseInst network architecture

如图 3 所示为 3 种不同的对象表示法，大多数实例分割框架都依赖边界框（图 3(a)，例如 Mask R-CNN，依赖 Fast R-CNN）或密集中心（图 3(b) dense centers，例如 CondInst，依赖于一阶全卷积目标检测 FCOS），且都会产生大量冗余预测和计算负担，主要原因是在预测时会在特征图上产生大量稠密锚点（dense anchors）或中心（centers）。而 SparseInst 可以输出实例激活图（图 3(c) IMP）来表示每一个对象，以此突出前景对象的信息区域。在聚合高亮区域时得到实例级特征，基于二部匹配，以一对一方式预测对象，避免了后续的非极大值抑制。



(a) 基于中心 (a) Center-based　(b) 基于区域 (b) Region-based　(c) 实例激活图 (c) IMP

图 3 对象表示法

Fig. 3 Object representation

在编码器结构当中，SparseInst 网络为加快推理速度，重建了特征金字塔，在 C_5 之后采用金字塔池化模块（pyramid pooling module，PPM）来放大感受野，并将 P3～P5 融合输出，进一步增强输出的单水平特征。在解码器结构当中，掩码分支感知掩码特征，实例分支产生实例激活图和 N 个实例特征，用于识别和感知核。由于 SparseInst 具有实例激活图的简单而有效的设计，使得其推理速度非常快，在 COCO 数据集基准上可达到 40 fps 和 37.9 AP，其在速度与准确性方面明显优于其余实例分割网络。

本文将 SparseInst 网络增加于 ORB-SLAM2 的前端关键帧设置当中，用于提取实例掩膜。将先验动态物体掩膜结合，并将其掩膜内特征点标记为动态特征点，初始动态概率为 α，降低其特征点对系统位姿估计产生的影响。选取分割分数高于 θ 的静态物体掩膜保存，用于三维重建。

1.3 基于贝叶斯概率的动态传播

为确保可以剔除每一帧中的动态特征点，本文设计了一种基于贝叶斯概率的特征点动态概率传播。利用关键帧中得到的语义信息，根据相邻帧之间的位姿变换关系将上一帧中的动态信息传播到当前帧中。

1）移动概率初始化

移动概率初始化在每一帧关键帧中进行，初始化依据实例分割网络得到的分割结果。根据 SparseInst 网络得到的动态掩膜进行赋值，动态掩膜内特征点移动置信度初始化为：

$$bel(x_0) = \alpha \tag{1}$$

动态物体掩膜外特征点移动置信度初始化为：

$$bel(x_1) = 1 - \alpha \tag{2}$$

在本文中为确保初始动态物体置信度一致，α 设定为 0.95。

2）测量概率更新

为提升系统的鲁棒性与实时性，降低系统资源浪费，本文仅在关键帧中使用实例分割算法获取语义信息，在普通帧中采用贝叶斯概率传播降低当前帧中动态物体对

系统位姿估计产生的影响,采用地图中移动概率小于 α 的地图点(即为静态地图点) 与当前帧中的特征点进行匹配。此阶段估计出的位姿为 $\boldsymbol{R}_1$ 和 $\boldsymbol{t}_1$。

每一帧中特征点包括动态 (d)、静态(s) 与中间态 3 种状态,以最终传播得到的结果进行判定。根据得到的位姿,将上一帧中特征点投影到当前帧当中,并进行特征匹配。此时得到的特征点匹配像素差记为 d。根据像素差 d 采用式(3) 进行测量移动概率更新:

$$P(z_t = d \mid m_t = d) = \begin{cases} 1, & d < d_2 \\ \dfrac{d - d_2}{d_1 - d_2}, & d_2 < d < d_1 \\ 0, & d_1 < d \end{cases} \tag{3}$$

3)状态更新

根据初始移动概率与测量概率,将移动概率状态更新表示为贝叶斯滤波器,即式(4)中, $bel(m_t)$ 为当前帧中特征点移动概率。

根据贝叶斯规则与条件独立性可知,当前帧观测 z_t 仅依赖于当前状态 m_t,η 为归一化常数,$bel(m_{t-1})$ 为上一帧中对应特征点移动概率。其中状态预测 $\overline{bel(m_t)}$ 可由式(5) 得到。

$$\begin{aligned} & bel(m_t) = P(m_t \mid z_{1:t}, m_0) = \\ & \eta P(z_t \mid m_t, z_{1:t-1}, m_0) P(m_t \mid z_{1:t-1}, m_0) = \\ & \eta P(z_t \mid m_t) P(m_t \mid z_{1:t-1}, m_0) = \eta P(z_t \mid m_t) \overline{bel(m_t)} \end{aligned} \tag{4}$$

$$\begin{aligned} & \overline{bel(m_t)} = \int P(m_t \mid m_{t-1}, z_{1:t-1}) P(m_t \mid z_{1:t-1}) \mathrm{d}m_{t-1} = \\ & \int P(m_t \mid m_{t-1}) bel(m_{t-1}) \mathrm{d}m_{t-1} \end{aligned} \tag{5}$$

$$Status(m_t^i) = \begin{cases} dynamic, & P(m_t) \geqslant \theta_d \\ static, & P(m_t) < \theta_s \\ uncertain, & \theta_s \leqslant P(m_t) < \theta_d \end{cases} \tag{6}$$

为兼容物体由静态转为动态的情况,本文状态转移概率设定为 $P(m_t \mid m_{t-1}) = 0.95$。计算得到 $bel(m_t)$ 后,按照式(6)进行当前帧动静态点更新,并将动态特征点进行标记与剔除。

实验中通过阈值 θ_d 和 θ_s 来判断单个点是动态还是静态,θ_d 和 θ_s 分别设置为 0.7 和 0.4。

1.4　基于 DeepSDF 的多物体三维重建

为提升所构建地图语义性,本文将单幅图像二维物体掩膜检测扩展至单目多物体 SLAM 系统,实现多物体目标位姿估计与三维重建。二维图像掩膜检测采用 SparseInst 实例分割网络,将掩膜内特征点与物体地图点进行数据关联,当物体地图点数量足够时,使用 PCA 算法估计物体初始位姿。当前帧观测达到设定阈值时,实例化物体对象,包括物体二维掩膜、二维边界框、物体稀疏点云与初始物体位姿;在多帧关键帧观测且数据充足时将其放入预训练好的 DeepSDF 网络当中,经过优化得到单个物体的 32 维或 64 维形状编码与位姿。经多帧观测可实现对多物体的三维重建与位姿估计。

1)DeepSDF 网络预训练

DeepSDF 网络不同于传统 SDF 只能生成某一个形状的隐式表示,DeepSDF 采用深度学习网络生成某一类物体的连续 SDF 表示,即物体的隐式向量。且 DeepSDF 采用一种自解码网络结构来实现 SDF 估计,先随机定义物体形状的隐式向量,通过训练反向传播得到更精准的编码向量,这种解码网络结构相较于自编码更加鲁棒。DeepSDF 网络的编码形状如图 4 所示。

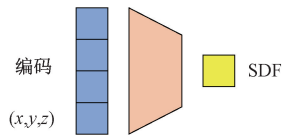


图 4　DeepSDF 编码形状

Fig. 4　Coded shape of DeepSDF

本文采用 ShapeNet 点云数据集[23] 对 DeepSDF 网络预训练,该数据集由普林斯顿大学、斯坦福大学和 TTIC 研究人员共同构建,为每个 3D 模型提供众多语义标注,可为 DeepSDF 网络训练提供充足而有效的数据。

2)物体数据关联与三维重建

为确保输入 DeepSDF 网络的静态物体有充足的数据,本文在 ORB-SLAM2 的基础上,结合 SparseInst 实例分割网络补充关键帧对象检测与特征点数据关联,仅在数据充足时进行三维物体重建。通过检测到的实例分割物体实例化对象,对象包含属性:图像掩膜 $\boldsymbol{M}$、二维边界框 $\boldsymbol{B}$、提取到对象的 ORB 特征点对应的三维稀疏点云 $\boldsymbol{P}$ 及深度观测值结合估计对象位姿 $\boldsymbol{T}_{co}$ 与隐式形状编码 z。

物体数据关联:在每一帧关键帧中,对检测到的对象进行匹配与数据关联,提取对象掩膜内 ORB 特征点及对应三维地图点与深度观测值,并保存在数据库中。当物体数据充足后,将数据输入 DeepSDF 预训练网络当中,得到物体形状隐式编码与优化后位姿。

物体数据关联在关键帧实例分割后进行,主要分为:实例化新对象和与现有对象数据关联。当新对象检测到时,对象 I 通过实例分割掩膜 $\boldsymbol{M}$、二维边界框 $\boldsymbol{B}$、三维稀疏点云 $\boldsymbol{P}$ 与初始对象位姿 $\boldsymbol{T}_{co,0}$ 进行实例化。

$$I = \{\boldsymbol{B}, \boldsymbol{M}, \boldsymbol{P}, \boldsymbol{T}_{co,0}\} \tag{7}$$

为更好进行物体三维重建,本文构建表面一致性损失,如式(8)所示。

$$L_{surf} = \frac{1}{|\Omega_m|} \sum_{u \in \Omega_m} G^2(\boldsymbol{T}_{oc} \boldsymbol{\pi}^{-1}(u, \boldsymbol{P}), z) \tag{8}$$

其中,Ω_m 为单目关键帧中实例分割掩膜内二维特征点对应三维地图点 $\boldsymbol{P}$ 集合,来源于 ORB-SLAM2 系统中

ORB 特征提取与稀疏重建结果。物体位姿 $T_{oc}=[s\boldsymbol{R}_{co},\boldsymbol{t}_{co};0,1]\in \boldsymbol{Sim}(3)$ 通过在物体三维地图点集合 Ω_m 上执行 PCA 得到。z 为 DeepSDF 重建稠密编码。在真实场景,单目视角掩膜内二维像素结合难以与物体完全重合,且存在物体部分被遮挡的情况,此时物体三维重建会存在偏差。为此,设定在一定距离 d_m 仅重建一个物体。最终目的是最小化表面一致性损失 L_{surf},得到每个对象的编码 z 与位姿 T_{co}。

2 实验结果与分析

2.1 实验条件

本文硬件实验平台为 AMD R7-5800H@3.2 GHz CPU,16 GB 内存,NVIDIA RTX 3060 6 GB GPU,系统环境为 Ubuntu 20.04。SLAM 系统使用 C++编写,实例分割动态区域检测使用 Python 编写,贝叶斯动态概率传播与多物体三维重建采用 C++编写。实验使用公开数据集 TUM RGB-D 动态数据集和 Boon RGB-D 动态数据集,采用单目序列对 SLAM 系统进行定量评估,并与主流 SLAM 系统进行定性对比。

TUM RGB-D 动态数据集由德国慕尼黑工业大学采集,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。数据集包含两类场景,即高动态场景和低动态场景。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;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 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、y 和 z 轴运动。

Boon RGB-D 动态数据集由波恩大学的 PRBonn 实验室采集,该数据集包含高度动态的序列。共 24 个动态序列和 2 个静态序列,动态序列中人执行不同的任务,例如操纵箱子或玩气球。本文中选取部分序列作为实验定位精度评估数据集,主要有 rgbd_bonn_ballon、rgbd_bonn_balloon_tracking、rgbd_bonn_crowd 和 rgbd_bonn_kidnapping_box,共 7 个序列。分别对应移动中的人向上拍气球、人投掷气球且相机追踪气球、多人移动且存在遮挡相机、纸盒子被移动的场景。

2.2 实验定位精度

本文采用绝对位姿误差(absolute pose error,APE)和相对位姿误差(relative pose error,RPE)作为算法定位精度的评价指标。表 1 和 2 为本算法与 ORB-SLAM2 算法在 TUM 动态数据集上的数据对比,表 3 为本算法相对 ORB-SLAM2 算法的提升,图 5 为本算法、ORB-SLAM2 算法与真实轨迹对比,采用 evo 工具评估里程计全局轨迹误差。APE 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,其计算公式如式(9)、(10)所示,式(9)为单个位姿真值与估计值之间的差异,单帧的 APE 计算为式(10)所示,整体绝对位姿误差 APE 计算的均方根误差如式(11)所示。

$$E_i=P_{est,i}\ominus P_{ref,i}=P_{ref,i}^{-1}P_{est,i}\in SE(3) \tag{9}$$

$$APE_i=\|\operatorname{trans}(E_i)\| \tag{10}$$

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}APE_i^2} \tag{11}$$

其中,$P_{est,i}$ 为第 i 帧系统估计位姿,$P_{ref,i}$ 为第 i 帧相机位姿真值,$\ominus$ 为逆合成算子,E_i 为第 i 帧二者之间差异。

RPE 用于评估视觉里程计的漂移量,与 APE 不同的是 RPE 计算的两个绝对位姿差异之间的差异,APE 计算的是两个绝对位姿之间的差异。其计算公式如式(12)~(14)所示。

$$E_{i,j}=\delta_{est_{i,j}}\ominus\delta_{ref_{i,j}}=(P_{ref,i}^{-1}P_{ref,j})^{-1}(P_{est,i}^{-1}P_{est,j})\in SE(3) \tag{12}$$

$$RPE_{i,j}=\|\operatorname{trans}(E_{i,j})\| \tag{13}$$

$$RMSE=\sqrt{\frac{1}{N}\sum_{\forall\ll\sim i,j}RPE_{i,j}^2} \tag{14}$$

每个误差表中采用平均值(mean)、中位数(median)和均方根误差(RMSE)作为评价指标。从表 1 与 3 中可看出,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 54.1%,并在静态场景中的定位精度也与 ORB-SLAM2 相当。

表 3 为本文算法与 ORB-SLAM2 算法的对比,平均绝对位姿误差的 RMSE 提升为 56.3%。

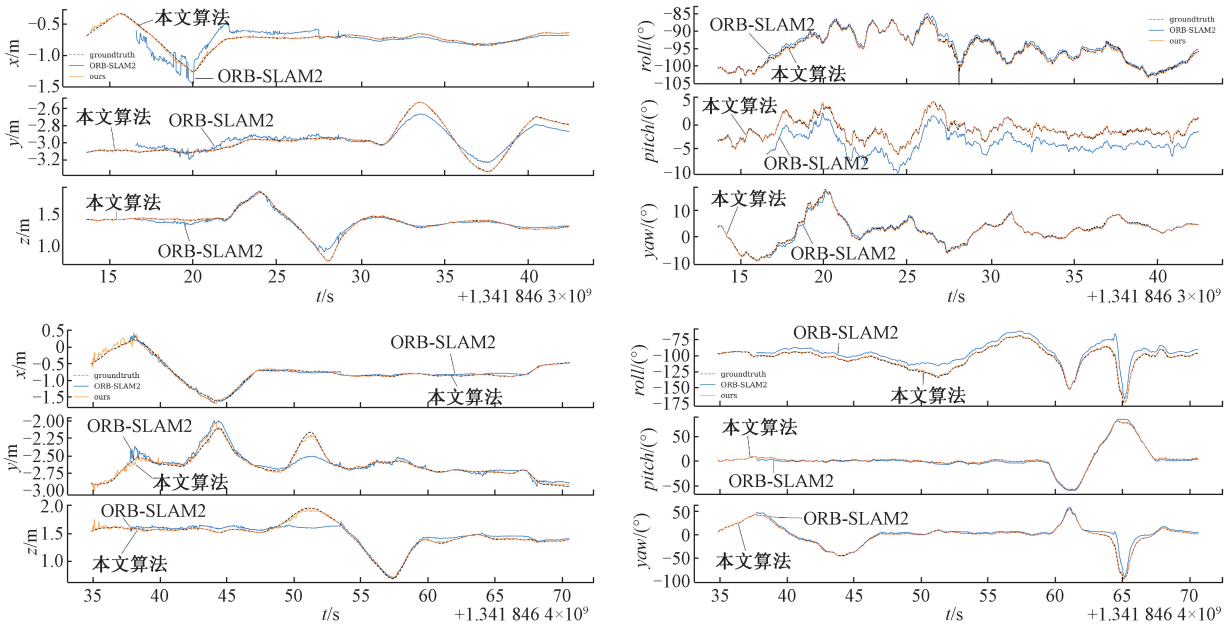
在 fr3_w_rpy 序列当中,本文算法在 ORB-SLAM2 算法基础上提升相对较小,主要原因为:1)fr3_w_rpy 序列相机旋转角度过大且出现特征点过少的场景,特征点剔除后会导致特征点数量更少;2)动态物体遮挡,系统剔除掉动态物体上特征点,可能会导致系统追踪丢失。在图 5 轨迹图中,可看出本文算法估计出相机估计与真实轨迹偏差很小,而 ORB-SLAM2 算法估计出相机与实际轨迹偏差较大。

如表 4 与 5 所示,为本文算法与目前领先的 SLAM 算法(DynaSLAM[5]、ORB-SLAM3[2])在 TUM 动态序列场景下的绝对位姿误差 APE 和相对位姿误差 RPE 对比,本文算法相较于 DynaSLAM 算法和 ORB-SLAM3 算法分别平均提升 30.1%、25.3%。在 w_xyz 序列当中,本文算法相对 ORB-SLAM2 算法、DynaSLAM 算法和 ORB-SLAM3 算法分别提升 85.7%、75.6%和 55.5%。

表 1　ORB-SLAM2 和本文算法的绝对位姿误差的对比

Table 1　Comparison of absolute pose error between ORB-SLAM2 and our algorithm　　m

序列	ORB-SLAM2 算法			本文算法		
	均方根误差	平均值	中位值	均方根误差	平均值	中位值
fr3_s_hs	0. 051 918	0. 045 580	0. 042 514	**0. 036 770**	0. 027 363	0. 019 548
fr3_s_rpy	**0. 049 442**	0. 045 546	0. 043 477	0. 050 570	0. 044 362	0. 039 255
fr3_s_static	**0. 017 821**	0. 014 187	0. 012 222	0. 019 132	0. 016 219	0. 012 322
fr3_s_xyz	0. 042 301	0. 038 048	0. 035 109	**0. 041 292**	0. 031 104	0. 021 878
fr3_w_hs	0. 135 035	0. 094 861	0. 061 219	**0. 050 565**	0. 035 550	0. 024 011
fr3_w_rpy	0. 151 751	0. 135 282	0. 108 058	**0. 109 479**	0. 097 911	0. 101 261
fr3_w_static	0. 019 122	0. 015 645	0. 013 442	**0. 012 342**	0. 011 212	0. 011 126
fr3_w_xyz	0. 132 151	0. 114 897	0. 100 04	**0. 018 917**	0. 014 751	0. 012 257



(a) *xyz*视角
(a) *xyz* perspective

(b) *rpy*视角
(b) *rpy* perspective

图 5　TUM 动态场景轨迹对比图

Fig. 5　Track comparison diagram in TUM dynamic scene

表 2　ORB-SLAM2 和本文算法的相对位姿误差的对比

Table 2　Comparison of relative pose error between ORB-SLAM2 and our algorithm　　m

序列	ORB-SLAM2 算法			本文算法		
	均方根误差	平均值	中位值	均方根误差	平均值	中位值
fr3_s_hs	**0. 013 63**	0. 009 259	0. 006 832	0. 014 370	0. 010 387	0. 007 828
fr3_s_rpy	**0. 005 912**	0. 004 182	0. 002 827	0. 009 081	0. 004 771	0. 002 266
fr3_s_static	0. 008 819	0. 003 411	0. 001 348	**0. 008 259**	0. 003 330	0. 001 483
fr3_s_xyz	**0. 016 314**	0. 0108 32	0. 007 710	0. 017 135	0. 010 271	0. 006 974
fr3_w_hs	**0. 023 441**	0. 014 466	0. 010 032	0. 032 182	0. 016 001	0. 009 239
fr3_w_rpy	0. 022 362	0. 008 848	0. 003 647	**0. 017 028**	0. 010 329	0. 006 270
fr3_w_static	0. 006 844	0. 002 339	0. 001 070	**0. 003 917**	0. 002 917	0. 002 243
fr3_w_xyz	0. 037 399	0. 016 823	0. 008 529	**0. 012 172**	0. 009 602	0. 007 491

表 3 TUM 动态场景下 APE 的提升

Table 3 Improvement of ATE for TUM dynamic scene

%

序列	均方根误差	平均值	中位值
fr3_w_xyz	85.7	87.2	87.7
fr3_w_static	35.5	28.3	17.4
fr3_w_rpy	27.9	27.6	6.3
fr3_w_halfsphere	62.6	62.5	60.8

表 6 为本文算法与 ORB-SLAM2 算法在 Boon RGB-D 动态数据集上 APE 测试的对比,从测试结果可看到本文算法在 Boon 动态数据集中 APE 显著优于 ORB-SLAM2 系统,在其基础上平均提升 58.2%。

2.3 三维重建实验

Freiburg Cars 数据集[24]是相机以车为中心移动的序列组成。本文选取 Freiburg Cars 数据集中部分序列与实际场景作为三维重建实验数据。图 6、图 7(a)和(c)分

表 4 TUM 动态序列 APE 测试对比

Table 4 Comparison of APE test results on TUM dynamic scene

序列	ORB-SLAM2 算法			DynaSLAM 算法			ORB-SLAM3 算法			本文算法		
	均方根误差/m	标准差/m	Traj/%	均方根误差/m	标准差/m	Traj/%	均方根误差/m	标准差/m	Traj/%	均方根误差/m	标准差/m	Traj/%
w_xyz	0.132 151	0.065 289	89.5	0.077 459	0.042 202	97.8	0.042 521	0.030 588	98.4	**0.018 917**	0.009 897	99.1
w_static	0.019 122	0.010 994	93.4	0.013 992	0.005 846	92.3	0.016 125	0.010 272	93.8	**0.012 342**	0.005 159	93.9
w_rpy	0.151 751	0.068 755	94.3	0.130455	0.083464	76.6	0.113892	0.045 066	96.2	**0.109 479**	0.0489 80	97.7
w_halfsphere	0.135 035	0.096 103	91.7	0.084 044	0.064 147	92.1	0.080 378	0.063 524	99.5	**0.050 565**	0.035 959	99.7

表 5 TUM 动态序列 RPE 测试对比

Table 5 Comparison of RPE test results on TUM dynamic scene

m

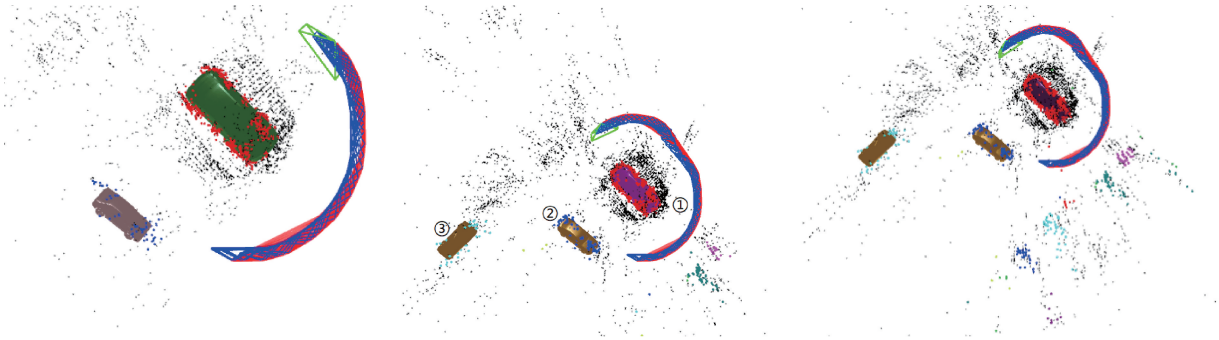
序列	ORB-SLAM2 算法		DynaSLAM 算法		ORB-SLAM3 算法		本文算法	
	均方根误差	标准差	均方根误差	标准差	均方根误差	标准差	均方根误差	标准差
w_xyz	0.037 399	0.033 402	0.023 683	0.020 02	0.020 192	0.015 628	**0.012 172**	0.007 480
w_static	0.006 844	0.006 431	**0.003 639**	0.002 498	0.008 662	0.007 957	0.003 917	0.002 613
w_rpy	0.022 362	0.020 537	0.130 455	**0.083 464**	0.008 795	0.007 688	0.017 028	0.013 537
w_halfsphere	**0.023 441**	0.018 445	0.044 704	0.040 207	0.058 046	0.053 546	0.032 182	0.027 923

表 6 Boon 动态序列 APE 测试对比

Table 6 Comparison of APE test results on Boon dynamic scene

m

序列	ORB-SLAM2 算法				本文算法			
	均方根误差	平均值	中位值	标准差	均方根误差	平均值	中位值	标准差
rgbd_bonn_balloon	0.119 808	0.110 689	0.103 961	0.045 847	**0.039 988**	0.035 058	0.027 677	0.019 236
rgbd_bonn_balloon2	0.172 595	0.161 052	0.160 583	0.062 06	**0.133 820**	0.120 034	0.123 403	0.059 159
rgbd_bonn_balloon_tracking	0.134 811	0.107 010	0.072 196	0.081 992	**0.025 897**	0.023 144	0.020 789	0.011 620
rgbd_bonn_balloon_tracking2	0.351 011	0.307 843	0.255 527	0.168 647	**0.072 616**	0.062 774	0.056 494	0.036 503
rgbd_bonn_crowd	0.067 053	0.059 778	0.051 592	0.030 376	**0.046 709**	0.042 552	0.039 114	0.019 262
rgbd_bonn_kidnapping_box	0.061 823	0.057 754	0.051 812	0.022 057	**0.020 267**	0.018 797	0.017 717	0.007 579
rgbd_bonn_kidnapping_box2	**0.058 591**	0.053 802	0.055 528	0.023 198	0.064 402	0.018 635	0.011 625	0.009 368



(a) 效果图a
(a) Rendering a

(b) 效果图b
(b) Rendering b

(c) 结果图
(c) Result graph

图 6　Freiburg Cars 三维重建效果图

Fig. 6　Rendering of 3D reconstruction in Freiburg Cars datasets

别为在 Freiburg Cars 数据集与真实场景中三维重建效果图,其中曲线代表相机位姿估计轨迹,不同点云代表已实例化的物体对象与已三维重建的物体对象,即图 7a 的①②③与图 7c 中的①②③④所表示的物体对象。图 7(b)和(d)分别对应图 7(a)和(c)当前帧实际场景图像。表 7 为三维重建结果评定表,选用 3 种指标进行评估:1)场景内中心对象重建准确度(accuracy of central object reconstruction, ACOR);2)正常区域内对象重建准确度(accuracy of object reconstruction in normal areas, AORNA);3)边缘对象是否重建(whether the edge object is reconstructed, WEOR)。

表 7　三维重建结果评定

Table 7　Evaluation of iterative reconstruction results

数据集	ACOR/%	AORNA/%	WEOR
Freiburg Cars	100	100	No
实际场景 1	100	100	No
实际场景 2	100	50	No
总计	100	83	No

如图 6(b)所示,在 Freiburg Cars 数据集三幅图中共重建 3 个对象,其中①为中心对象,②为正常区域内对象,③为边缘对象。由图 6(a)可以观测到中心对象①重建效果十分良好,原因在于所观测的帧数与地图点数量充足。对象②由于观测帧数与观测地图点数量不充足,导致三维重建效果相比于中心对象①较差。从效果图 6(b)中可以看出中心对象重建效果非常好,对象②在多帧观测后也正常进行重建。但在图 6(b)中看到边缘对象③在多帧数据后错误重建,主要原因为:1)数据观测量不足,车辆距离相对较远;2)存在多个车辆叠加,数据进行错误关联。最终重建结果图如图 6(c)所示。

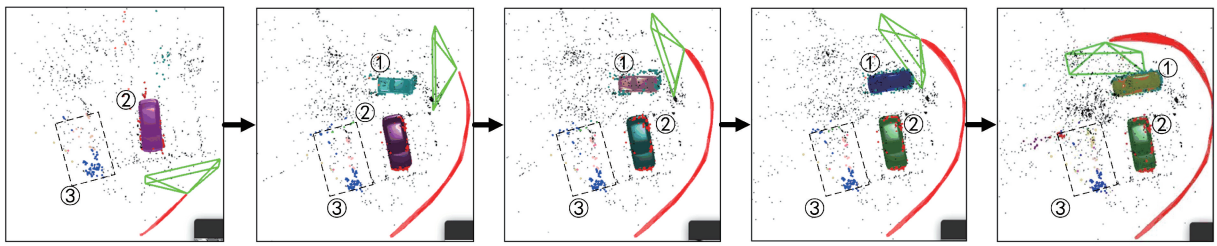
从图 7(a)与图 7(c)实际场景重建效果可知,图 7(a)中的中心对象①与②、图 7(c)中的中心对象①与②重建效果随观测帧数增大而提升;正常区域内对象根据观测帧数的不同,其重建效果也不同,如图 7(c)正常区域内部分对象③与④因观测帧数不足,其重建效果较差;在实际场景中的对边缘对象,即图 7(a)中使用虚线框标注的对象③可以被系统检测但并未重建。

如图 8 为 SLAM++算法[25]与 DSP-SLAM 算法[26]三维重建效果图。SLAM++采用 RGB-D 相机,其对室内场景中包含的桌椅进行重建,重建效果如图 8(a)所示。SLAM++同样需要预先建立物体数据库,但其需要结构先验,即所有物体同属于同一平面上。DSP-SLAM 系统同样采用 DeepSDF 对物体进行重建,不同的是其在多目或单目融合激光雷达时对多物体进行重建,单目情况下仅对首次检测到的物体进行重建,即只可重建一个物体,重建效果如图 8(b)。而本算法使用单目相机对环境内多个物体进行三维重建。为提高物体三维重建精度与相机位姿估计精度,后续我们会使用更鲁棒的多物体数据关联,并进行多物体之间、多物体与相机位姿之间、多物体与点云之间的联合优化。

2.4　实验算法运行时间

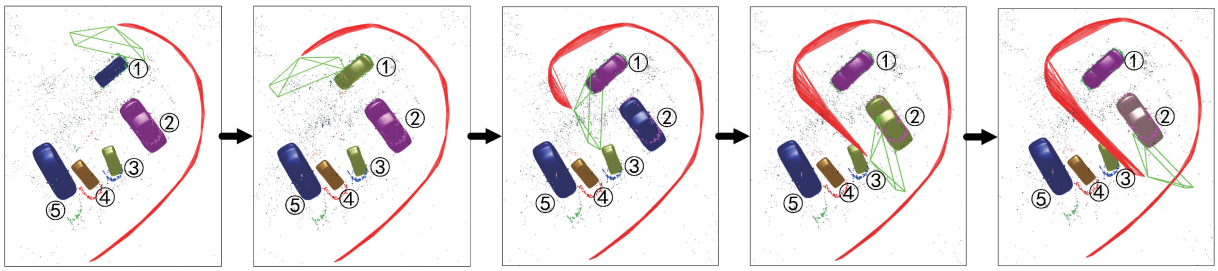
本文算法各个模块运行时间如表 8 所示,其中 SparseInst 实例分割网络与 DeepSDF 三维重建网络使用 CUDA(compute unified device architecture)并行计算加速。实例分割网络仅应用于关键帧,可达到 40 ms 左右,影响算法实时性的主要因素是 DeepSDF 多物体重建网络,当物体数量较多时消耗时间较长。

如表 9 所示为不同算法运行速度对比,其中 ORB-SLAM2 算法运行速度为 25 ms/帧;DynaSLAM 算法使用 Mask R-CNN 算法对每一帧图像进行实例分割,其消耗的时间约为 180 ms/帧,在本文设备环境下系统平均耗时



(a) 实际场景1效果图
(a) Rendering of reality scence 1

(b) 实际场景1
(b) Reality scence 1

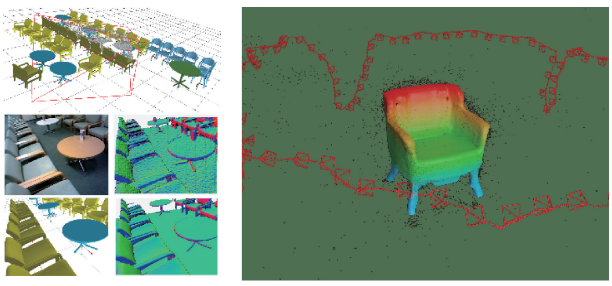


(c) 实际场景2效果图
(c) Rendering of reality scence 2

(d) 实际场景2
(d) Reality scence 2

图 7 实际场景三维重建效果图

Fig. 7 Rendering of 3D reconstruction in real scene



(a) SLAM++ (b) DSP-SLAM

图 8 其他算法三维重建效果图

Fig. 8 Other algorithms 3D reconstruction rendering

约为 2 s/帧;Detect-SLAM 采用 SSD 算法,每帧消耗时间约为 310 ms;因本文算法采用关键帧实例分割模块与普通图像帧贝叶斯概率传播上模块,二者耗时平均 15 ms/帧,系统总耗时为 40 ms/帧(不包含三维重建部分),故证明本文算法可以极大提升系统运行速度。

表 8 各模块运行时间

Table 8 Running time of each module

模块	运行时间	
SparseInst	40 ms/KeyFrame	平均 15 ms/Frame
物体数据关联	40 ms/KeyFrame	
贝叶斯概率传播	6 ms/Frame	
DeepSDF 三维重建	300 ms(局部映射线程)	多物体

表 9 不同算法运行时间对比

Table 9 Comparison of running time of different algorithms

算法	运行时间 ($ms \cdot Frame^{-1}$)	运行环境
ORB-SLAM2 算法	40	CPU
DynaSLAM 算法	2 000	RTX 3060 Laptop
DynaSLAM 算法	1 000	Titan X
本文算法(仅 Sparse and Bayesian)	40	RTX 3060 Laptop
本文算法(全部)	100	RTX 3060 Laptop

本文三维重建部分整体耗时包括:多物体数据关联、物体实例化对象、使用 DeepSDF 对物体进行三维重建。其中多物体数据关联在追踪(tracking)线程进行,耗时为 40 ms/关键帧;物体实例化对象与三维重建部分在局部映射(LocalMapping)线程中进行,物体实例化对象耗时约为 20 ms,单个对象三维重建耗时约为 100 ms。在每个物体重建完成后,后续只需根据输入的点云对物体数据进行维护更新。整体系统耗时平均为 100 ms 每帧(包含三维重建部分)。

3　结　　论

为提高环境中语义信息的利用率及 SLAM 系统精度,本文提出一种面向动态环境的基于实例分割和三维重建的多物体单目 SLAM 算法。该算法建立在 ORB-SLAM2 的基础上,基于实例分割网络与贝叶斯概率传播技术,检测图像帧先验动态物体,并降低其对位姿估计产生的影响。对关键帧进行实例分割网络的推理,并将动态物体上特征点依据贝叶斯概率传播将其传播到普通帧当中,并将其剔除。在 TUM 数据集与 Boon 数据集中,动态场景下本文算法与 ORB-SLAM2、ORB-SLAM3、DynaSLAM 算法相比,定位精度平均提升 54.1%、6.3%、22.6%。在 TUM 的 w_xyz 序列当中,本文算法相对 ORB-SLAM2、DynaSLAM 和 ORB-SLAM3 系统分别提升 85.7%、75.6%和 55.5%。实验结果验证了本文算法在动态场景下具有更好的定位精度和鲁棒性。在三维重建实验中本文在 Freiburg Cars 数据集与实际场景中对环境中存在的语义物体进行多物体重建,充分利用了环境中存在的语义信息,将地图语义化,且时间消耗相对较少。从三维重建结果可知,本文对中心对象的重建效果十分好,但对正常区域内存在的其他对象重建效果较差,具体原因为观测数不足与多帧后对象数据关联不充分导致,未来可增加对正常区域对象的鲁棒数据关联与物体二次观测后重定位数据关联。

参考文献

[1] 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.

[2] CAMPOS C, ELVIRA R, RODRÍGUEZ J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap slam [J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[3] SAPUTRA M R U, MARKHAM A, TRIGONI N. Visual SLAM and structure from motion in dynamic environments: A survey[J]. ACM Computing Surveys, 2018, 51(2): 1-36.

[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] 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.

[6] DAI W, ZHANG Y, LI P, et al. RGB-D SLAM in dynamic environments using point correlations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(1): 373-389.

[7] 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.

[8] HE K, GKIOXARI G, DOLLÁR P, et al. MASK R-CNN[C]. In Proceedings of the IEEE International Conference on Computer Vision, 2017: 2961-2969.

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

[10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]. Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[11] YANG S, FAN G, BAI L, et al. SGC-VSLAM: A semantic and geometric constraints VSLAM for dynamic indoor environments[J]. Sensors, 2020, 20(8): 2432.

[12] REDMON J, FARHADI A. Yolov3: An incremental improvement [J]. ArXiv Preprint, 2018, ArXiv: 1804.02767.

[13] 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), 2018: 1168-1174.

[14] 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.

[15] 刘钰嵩,何丽,袁亮,等. 动态场景下基于光流的语义 RGBD-SLAM 算法[J]. 仪器仪表学报,2022,43(12): 139-148.
LIU Y S, HE L, YUAN L, et al. Semantic RGBD-SLAM in dynamic scene based on optical flow[J].

Chinese Journal of Scientific Instrument, 2022, 43(12): 139-148.

[16] BESCOS B, CAMPOS C, TARDÓS J D, et al. DynaSLAM II: Tightly-coupled multi-object tracking and SLAM[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 5191-5198.

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

[18] BALLESTER I, FONTÁN A, CIVERA J, et al. DOT: Dynamic object tracking for visual SLAM[C]. IEEE International Conference on Robotics and Automation (ICRA), 2021: 11705-11711.

[19] WADUD R A, SUN W. DyOb-SLAM: Dynamic object tracking SLAM system[J]. ArXiv Preprint, 2022, ArXiv: 2211.01941.

[20] MICHAEL E, SUMMERS T, WOOD T A, et al. Probabilistic data association for semantic SLAM at scale[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022:4359-4364.

[21] CHENG T, WANG X, CHEN S, et al. Sparse instance activation for real-time instance segmentation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022:4423-4432.

[22] PARK J J, FLORENCE P, STRAUB J, et al. DeepSDF: Learning continuous signed distance functions for shape representation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019:165-174.

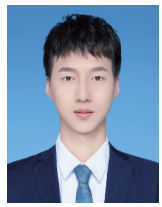
[23] CHANG A X, FUNKHOUSER T, GUIBAS L, et al. Shapenet: An information-rich 3d model repository[C]. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015:1912-1920.

[24] SEDAGHAT N, BROX T. Unsupervised generation of a view point annotated car dataset from videos[C]. IEEE International Conference on Computer Vision (ICCV), 2015:1314-1322.

[25] SALAS-MORENO R F, NEWCOMBE R A, STRASDAT H, et al. Slam++: Simultaneous localisation and mapping at the level of objects[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 1352-1359.

[26] WANG J, RÜNZ M, AGAPITO L. DSP-SLAM: Object oriented SLAM with deep shape priors[C]. International Conference on 3D Vision (3DV), IEEE, 2021: 1362-1371.

作者简介



冯洲,2021 年于太原理工大学获得学士学位,现为太原理工大学硕士研究生,主要研究方向为视觉语义 SLAM 和移动机器人位姿估计。

E-mail: fz990906@163.com

Feng Zhou received his B. Sc. degree from Taiyuan University of Technology in 2021. He is currently a master student at Taiyuan University of Technology. His main research interests include visual semantic SLAM and pose estimation.

续欣莹(通信作者),分别在 2002 年、2005 年和 2009 年于太原理工大学获得学士、硕士和博士学位,现为太原理工大学电气与动力工程学院教授,主要研究方向为计算机视觉和智能控制。

E-mail: xuxinying@tyut.edu.cn

Xu Xinying (Corresponding author) received his B. Sc., M. Sc. and Ph. D. degrees all from Taiyuan University of Technology in 2002, 2005 and 2009, respectively. He is currently a professor at Taiyuan University of Technology. His main research interests include computer vision and intelligent control.