

动态场景下基于激光雷达的 SLAM 算法

冯迎宾¹, 李亚玮¹, 王天龙², 曾城锋¹

(1. 沈阳理工大学 自动化与电气工程学院, 沈阳 110159; 2. 中国科学院 机器人与智能制造创新研究院, 沈阳 110016)

摘要: 使用激光雷达在动态场景下实现精确的位姿估计与地图映射是同时定位与建图(simultaneous localization and mapping, SLAM)研究领域的重要内容之一,但动态环境中物体移动会导致 SLAM 算法精度下降,为此提出一种低成本且可有效剔除动态影响的激光雷达 SLAM 算法。首先引入深度图投影,通过检测相邻时刻深度图之间的像素值波动,筛选并去除动态点云;然后进行地面点云分割,利用非地面点云的特征实现位姿估计和地图映射,利用地面点云的特征施加地面约束,限制高度漂移;最后引入回环检测矫正全局姿态。实验结果表明,与 LOAM、LeGO-LOAM 和 SuMa 算法相比,本文算法可更有效剔除动态目标,提供更优秀的定位建图效果和鲁棒性能。

关键词: 同时定位与建图;激光雷达;动态剔除;全局优化

中图分类号: TP391.4 **文献标志码:** A **DOI:**10.3969/j.issn.1003-1251.2025.05.001

LiDAR-based SLAM Algorithm for Dynamic Environments

FENG Yingbin¹, LI Yawei¹, WANG Tianlong², ZENG Chengfeng¹

(1. Shenyang Ligong University, Shenyang 110159, China; 2. Institute for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China)

Abstract: To achieve LiDAR-based precise pose estimation and mapping in dynamic environments is a big challenge in the field of simultaneous localization and mapping (SLAM). Dynamic objects can significantly interfere with a robot's environmental perception and autonomous navigation. To address the decline in SLAM accuracy caused by moving objects in dynamic environments, this study proposes a low-cost yet effective LiDAR SLAM algorithm to eliminate dynamic interference. The algorithm introduces depth map projection to identify and remove dynamic point clouds by detecting pixel value fluctuations between consecutive depth maps. Ground point cloud segmentation is then performed, leveraging non-ground point cloud features for pose estimation and mapping. Ground point cloud features are utilized to impose ground constraints, mitigating height drift. Additionally, loop closure detection is incorporated to correct global pose drift. Experimental results demonstrate that, compared to LOAM, LeGO-LOAM, and SuMa, the proposed algorithm more effectively filters out dynamic objects, delivering superior localization and mapping performance with enhanced robustness.

Key words: simultaneous localization and mapping; LiDAR; dynamic removal; global optimization

同时定位与建图 (simultaneous localization and mapping, SLAM) 算法是智能机器人在未知环境下实现环境感知和自主导航的核心技术^[1]。与视觉 SLAM 技术^[2]相比,激光 SLAM 技术具有精度高、范围广、鲁棒性强、光照突变影响小等优势,在智能机器人和无人驾驶领域得到了广泛应用。

经过多年发展,激光 SLAM 技术取得了一定的研究成果,其中较为重要的研究内容是通过激光雷达点云特征配准获取机器人的实时位姿。文献[3]提出了一种并行的原始点云配准算法,通过引入一种新颖的关键帧筛选机制,提高了匹配效率,实现了高精度定位。文献[4]通过引入降采样和自适应阈值机制,优化了里程计计算方法,并增强了算法的通用性。上述基于直接匹配的 SLAM 方法通常需要借助降采样来保证匹配效率,导致建图质量不佳,为此学者们提出了基于特征提取的一系列方法。文献[5]首次提出根据每个点提取的曲率分类出边缘和平面特征,用于后续节点的姿态估计。文献[6]提出了基于点云特征的改进随机一致性采样 (RANSAC) 地面分割算法,分割出地面点云和非地面点云,在复杂地面条件下表现出较好的鲁棒性和计算效率。基于特征提取的 SLAM 方法通过提取点的各种特征减轻了噪声的影响,并确保了计算效率,但会损失一些有价值的信息。

基于直接匹配和基于特征提取的 SLAM 方法都以应用场景的静态假设为前提,并未关注实际场景中的动态对象^[7],如无人驾驶场景中的车辆和行人等。实际应用场景中的动态物体很容易造成环境退化问题,潜在地影响激光 SLAM 定位的精确性和映射的一致性。因此,适应动态环境的激光 SLAM 技术成为当前研究的热点^[8]。

基于光线追踪的 SLAM 方法通过将三维空间划分成若干个体素单元,根据每个体素的占用状态判断是否存在动态物体。文献[9]提出一种基于体素的地图构建方法,利用追踪光束的路径标记出体素占用状态,通过统计体素的占用率来判断动态目标。该类方法可以有效应用于动态环境,但需要消耗大量内存和计算资源。基于深度学习的 SLAM 方法通过神经网络标注与识别动态物体并将其去除,如文献[10]基于语义信息来感知动态目标点,利用语义分割和几何信息提高定位与建图的精度。但深度学习需要使用大量的数据进行预训练和动态属性标注,增加了系统成本和复杂性,使得基于深度学习的算法泛化能力

变弱。

针对以上问题,本文提出一种适应动态环境的激光雷达 SLAM 方法,在深度空间中对点云信息进行处理,采用动态阈值筛选策略有效识别并过滤掉环境中的动态点云,减少其对匹配精度和建图一致性的影响。此外,通过地面分割算法分割出地面点云和非地面点云,对去除地面点云后的非地面点云进行特征提取和匹配,以提高激光里程计的计算速度,对分割出的地面点云提取出高度信息,以优化位姿估计,约束 z 轴漂移。引入回环检测机制,以实现全局优化,当检测到回环发生时,添加回环闭合约束因子,校正全局姿态。

1 动态环境下高精度 SLAM 算法

1.1 系统框架

本文激光雷达 SLAM 系统整体框架如图 1 所示。首先将激光雷达的原始点云按照球面投影的方式投影为深度图,基于深度图识别出动态点云和静态点云,并剔除动态点云;然后对静态点云进行分类,其中非地面点云输入特征提取节点进行线面特征提取,地面点云存储于容器;再通过特征提取获得非地面点云的面特征和边缘特征,据其进行相邻扫描帧的特征匹配,得到相邻扫描帧的最优位姿估计,由此构成激光里程计;最后由防止地面点云漂移的高度约束、校正全局姿态的回环检测和激光里程计共同构建出因子图优化算法,以获取更加精确的位姿估计和全局地图。

1.2 深度图投影

激光雷达的当前帧 (k) 三维原始点云表示为 $P_k = \{p_i(x, y, z) | i = 1, 2, \dots, n\}$, 其中 (x, y, z) 为原始点云中点 p_i 在雷达坐标系下的坐标, n 为点云中点的数量。将原始点云投影变换为深度图中的图像坐标 (u, v) , 变换公式为

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{2} [1 - \arctan(y, x) \pi^{-1}] \times w \\ [1 - (\arcsin(zr^{-1}) + f_{up})f^{-1}] \times h \end{pmatrix} \quad (1)$$

式中: w 和 h 分别为深度图像的高度和宽度; f 表示激光雷达的垂直视角范围, 为激光雷达的上仰角 f_{up} 和下俯角 f_{down} 之和, 即 $f = f_{up} + f_{down}$; r 为点 p_i 到雷达中心的距离, $r = \|p_i\|_2$ 。

经过深度投影后得到宽度为激光雷达垂直视角、长度为激光雷达水平视角、像素深度值表示点云距离信息、像素坐标与点云空间位置一一对应的一张深度图。将三维点云信息投射到二维深度

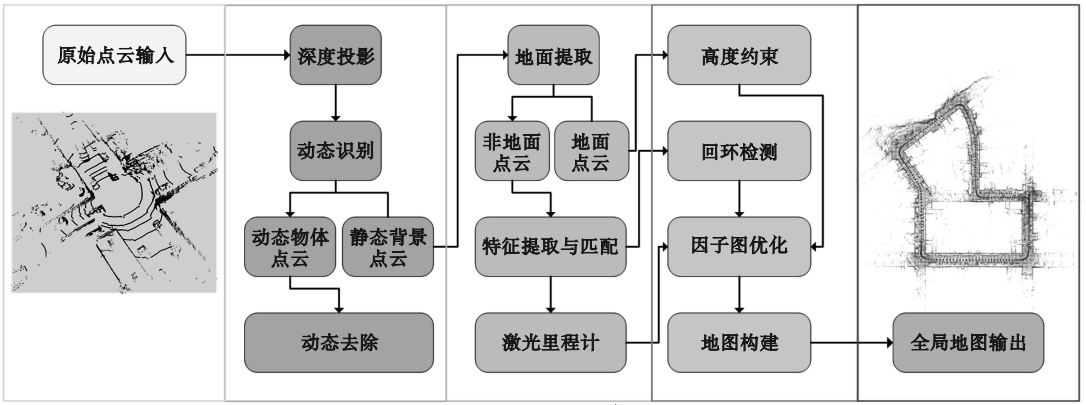


图 1 系统框架

Fig.1 System framework

图像中可以提高系统运行效率,且更直观地呈现场景内物体的空间形态特征,同时可以应用图像领域的理论和方法实现高效的目标识别与分割,为后续节点筛选和分类操作提供方法支持。

1.3 动态目标检测

动态目标在连续帧数据的配准过程中会造成空间位置的不一致,导致机器人对环境感知出现偏差。有效的动态目标滤除算法可为后续的位姿估计和地图映射提供可靠的感知数据,同时对于提升机器人导航系统的鲁棒性和精确度具有重要意义。

利用深度图进行动态点云剔除的方法称为可见性分析法。该方法认为,若某一激光点的传播路径被其他点所遮挡,则遮挡点可判定为动态目标。动态去除原理示意如图 2 所示,在某一时间段 $[t, t + 1]$,图中汽车发生了运动,原本无法采集的深色球位置的激光点云可以在 $t + 1$ 时刻采集到,采集的点云投影到深度图后,两个相邻时刻采集帧之间的同一位置呈现出不同的深度值,被动态目标遮挡位置具有较小深度值,动态目标移开后该位置具有较大深度值,则前面的遮挡物体被定义为动态物体。

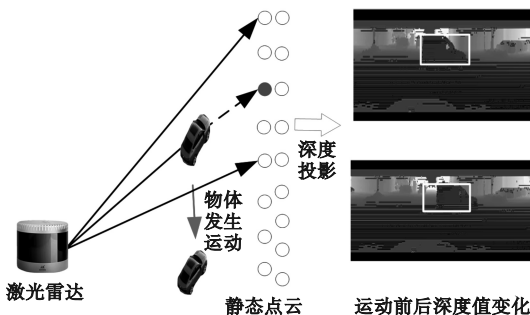


图 2 动态去除原理示意图

Fig.2 Dynamic removal principle diagram

由以上原理可知,动态物体的相对运动会在深度图中引发深度值的显著波动。因此,本文通过当前帧深度图像和前一帧深度图像的深度值相减,得到绝对差值深度图像,其深度值为 I_k^{diff} ,使用公式(2)进行动态点云判断,实现静态目标与动态目标的筛选分类。

$$\begin{cases} P_k^D = \{P_k \mid I_k^{diff} > \tau\} \\ \tau = \gamma \cdot r \end{cases} \quad (2)$$

式中: P_k^D 为筛选后的动态点云; τ 为阈值; γ 为算法设定的距离敏感度参数。通过 γ 与 r 相乘构成动态阈值 τ ,可有效避免将远处的静态物体误判为动态物体,从而提高筛选效果。动态点云与静态点云分割的可视化图如图 3 所示,框中为动态点云,框外为静态点云。

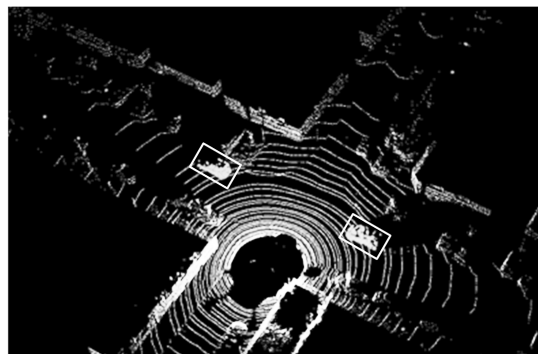


图 3 动态-静态分割可视化图

Fig.3 Dynamic-static segmentation visualization diagram

1.4 地面点云提取

地面点云作为激光雷达感知数据的关键组成部分,其特征可为 SLAM 系统提供可靠的空间约束,有效抑制建图过程中高度的累积误差^[11]。将地面点云分离出来还可以加快特征提取节点的运算速度,提升点云处理的效率。

地面点云分割原理如图4所示。设两束激光与障碍物的交点分别为 A 和 B , AB 连线与水平面的夹角为 β ,过 B 点的垂线与过 A 点的水平线交点为 C 。根据激光雷达的机械特性可计算出两束激光的发射角 α_1 和 α_2 ,利用飞行时间测距法(ToF)可计算出 O 点至 A 点和 B 点的距离(分别用 R_A 和 R_B 表示),则 β 的计算式为

$$\beta = \arctan \frac{|R_A \sin \alpha_1 - R_B \sin \alpha_2|}{|R_B \cos \alpha_2 - R_A \cos \alpha_1|} \quad (3)$$

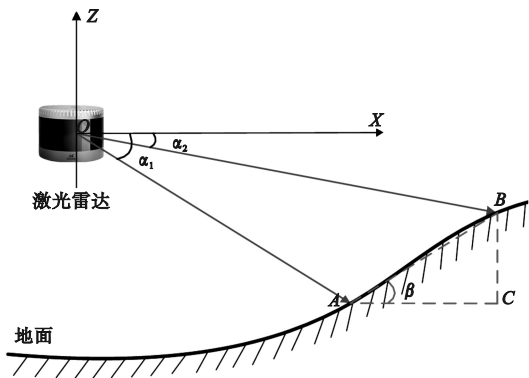


图4 地面点云分割原理示意图

Fig. 4 Ground point cloud segmentation principle diagram

由于多线激光雷达相邻光束之间的高度差异较小,所以地面点云的角度 β 会很小(近似于零),但非地面点云会因为光束落点的突变表现出较大的角度 β 。利用该特性,本文通过设置合适的阈值分割地面点云和非地面点云。

1.5 点云特征提取

点云特征提取是点云配准过程的关键步骤^[12],直接影响位姿估计的精度。首先对输入点云数据进行运动畸变校正,以消除传感器运动带来的误差;然后计算点云中每个点的局部曲率特征值,并依据预设的曲率阈值对特征点进行分类。当某点的曲率超过阈值时,将其归类为边缘特征点;反之,则为平面特征点。点云特征提取算法流程如图5所示。

基于曲率的特征提取方法能够有效保留点云中显著的几何特征,为后续点云匹配和位姿解算提供可靠的特征对应关系。曲率 c 的计算公式为

$$c = \frac{1}{|S| \cdot \|r_i\|} \sum_{j \in S, j \neq i} \|r_j - r_i\| \quad (4)$$

式中:集合 S 为同一条激光扫描线上点 i 的相邻点集合; i 和 j 分别代表 S 中的不同采样点; r_i 和 r_j 分别表示点 i 和点 j 对应的深度值向量。本文将集合 S 的规模设定为10个连续点。

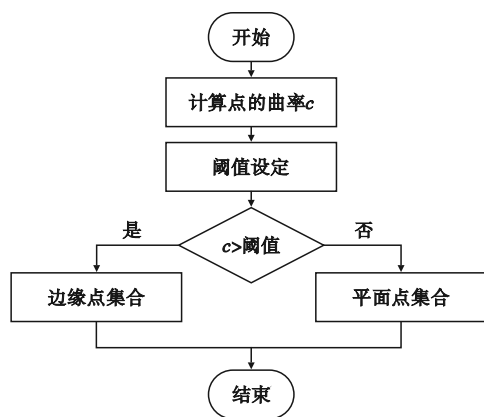


图5 点云特征提取算法流程图

Fig. 5 Point cloud feature extraction algorithm flowchart

1.6 激光雷达的位姿估计

SLAM的核心任务是位姿估计,在惯性测量单元(IMU)提供的初始位姿信息基础上,本文通过帧间特征匹配进行相邻时刻激光点云的空间变换求解,实现对机器人运动变换的精确估计。利用提取的边缘点集和面点集,分别构建点云的线特征匹配算法与面特征匹配算法,并通过最小二乘法优化计算,得到前后两帧点云之间的坐标变换关系。

空间中任意点的坐标可通过平移和旋转转换到另一个坐标系。设前一帧的点为 p_i ,当前帧的点为 p_j ,则可通过式(5)将前一帧的点转换到当前帧。

$$p_j = R \cdot p_i + t \quad (5)$$

式中: R 为旋转矩阵; t 为平移向量。如果以空间中两点的欧氏距离表示匹配的好坏,则可将匹配问题转化为求解欧氏距离最小值。目标函数定义为

$$f_e(X_{(k+1,i)}^L, T_{(k+1,i)}^L) = d_e \quad (6)$$

式中: f_e 表示线特征函数; d_e 表示线到线的距离; $X_{(k+1,i)}^L$ 表示点 i 在 $k+1$ 时刻雷达坐标系(用上角标 L 表示)下的坐标; $T_{(k+1,i)}^L$ 表示点 i 的旋转平移变换。

通过旋转平移变换使前一帧点云转换到当前帧后,与当前帧中边缘线之间的欧氏距离 d_e 最小,从而得到最优变换。 d_e 求解方法如式(7)所示。

$$d_e = \frac{|(\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,j)}^L) \times (\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,l)}^L)|}{|\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L|} \quad (7)$$

式中: $\bar{X}_{(k+1,i)}^L$ 表示上一帧中 i 点经过旋转平移变换到当前帧的位置; $\bar{X}_{(k,j)}^L$ 和 $\bar{X}_{(k,l)}^L$ 分别表示当前帧中第 j 个和第 l 个点的位置。

面特征匹配与线特征匹配类似,其目标是找到一个空间变换,使得当前帧中的所有点到前一

帧对应平面的距离平方和最小。目标函数表示为

$$f_{\text{g}}(\bar{X}_{(k+1,i)}^L, T_{(k+1,i)}^L) = d_{\text{g}} \quad (8)$$

式中: f_{g} 表示面特征函数; d_{g} 为点到面的距离。 d_{g} 求解公式为

$$d_{\text{g}} = \frac{|\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L \times (\bar{X}_{(k,j)}^L - \bar{X}_{(k,m)}^L)(\bar{X}_{(k+1,i)}^L - \bar{X}_{(k,j)}^L)|}{|(\bar{X}_{(k,j)}^L - \bar{X}_{(k,l)}^L) \times (\bar{X}_{(k,j)}^L - \bar{X}_{(k,m)}^L)|} \quad (9)$$

式中 $\bar{X}_{(k,m)}^L$ 为当前帧中 m 点坐标。

1.7 激光雷达的地图构建

地图构建节点以较低的频率对当前帧的点云进行线特征和面特征提取,并将其与地图中的点云进行匹配,优化位姿变换。同时,将点云扫描数据与全局地图关联,确保新观测的点云能够准确融入全局地图中,从而动态更新全局地图。

基于地面平坦且雷达与地面之间的高度变化在短时间内比较微小的假设,本文引入前端提取的地面点云来施加高度约束^[13],将当前帧的地面点云与地图中的地面模型进行匹配,计算点到地面的距离残差,将残差作为地面约束因子加入图优化中。通过最小化地面点云的残差,有效校正传感器在 z 轴方向上的漂移,更好地保持 z 轴坐标的一致性。

此外,本文引入回环检测进一步消除全局地图累积误差。在识别到回环闭合的情况下,首先运用迭代最近点(ICP)算法探寻当前特征集合与历史特征集合之间的对应关系,然后引入回环闭合约束因子,结合地面约束因子和雷达里程计因子构建出如图6所示的因子图以执行全局优化,从而校正整个地图构建和姿态估计效果。

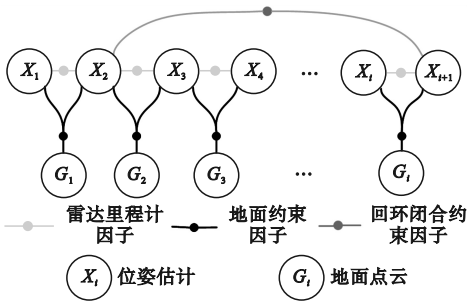


图6 优化节点的因子图

Fig.6 Optimized node factors graph

2 仿真实验与结果分析

本文采用自动驾驶的KITTI公开数据集和校

园环境内录制的Ours数据集进行验证实验,数据集包括大量的应用场景,可为算法提供强大的数据支撑。实验计算平台为装载16 G运行内存和Intel(R)Core(TM)i5-10210U CPU@1.6 GHz处理器的个人电脑,开发环境集成了ROS Melodic版本的Ubuntu 18.04 LTS操作系统,并采用RVIZ作为可视化工具。

2.1 动态剔除与建图效果分析

为验证算法的动态目标移除效果,本文选用KITTI数据集中的07序列子集作为数据输入,算法的深度图投影效果如图7所示。

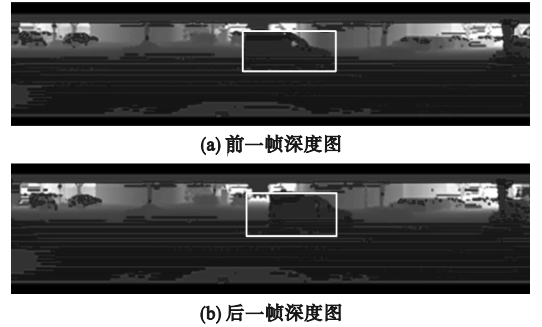


图7 相邻帧深度图

Fig.7 Adjacent frame depth map

由图7可知,深度图中车辆所在位置在前后两帧投影中发生像素的变化,直观反映出实际环境中车辆与雷达发生的相对运动。实验结果表明,本文算法可实时检测到环境中的动态物体,并将其点云集合抛弃,避免对后续里程计精度和建图的影响。

在Ours数据集的多个子集上对本文算法的动态目标剔除率进行验证,结果如表1所示。可见,本文算法对不同种类的动态目标均实现了有效剔除,漏剔除率为0%,很好地消除了动态目标的干扰。

表1 动态目标剔除结果

Table 1 Dynamic object removal results

子数据集	动态目标种类	动态数量	剔除数量	漏剔除率/%
Ours1	行人	11	11	0
Ours2	车辆	3	3	0
Ours3	自行车	6	6	0

本文算法通过动态剔除、地面约束和回环检测构建出更加精确的全局地图,更好地映射了真实的三维环境。为了更直观地表现全局地图改进效果,选用经典的LeGO-LOAM框架^[14]作为对比,LeGO-LOAM算法和本文算法采用相同算力

的计算平台,均采用 KITTI 数据集的 07 序列子集。对比结果如图 8 所示。

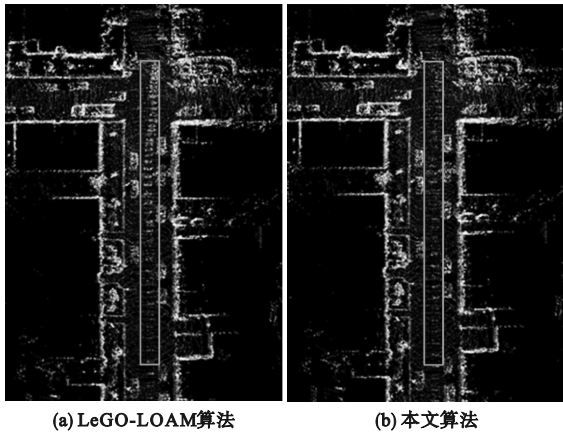


图 8 动态目标剔除效果对比

Fig. 8 Comparison of dynamic object removal effects

由于 LeGO-LOAM 算法对动态物体没有进行任何处理,导致算法构建的全局地图上出现长长的“拖影”,如图 8(a)所示。本文算法经过有效的动态物体处理后,解决了物体运动带来的负面影响,构建的全局地图不存在“拖影”现象,如图 8(b)所示,体现出更好的建图一致性。

2.2 里程计定位精度评估

采用 KITTI 数据集中的 00、05、06、07 序列子集和自制 Ours1、Ours2 序列子集验证本文算法在不同环境和条件下的性能表现,并与经典的 LOAM^[5]和 LeGO-LOAM 算法进行对比,使用均方根误差(RMSE)作为轨迹精度的评价指标,对比结果如表 2 所示。

表 2 不同算法的 RMSE 对比

Table 2 RMSE comparison of different algorithms

算法	RMSE/m					
	#00	#05	#06	#07	Ours1	Ours2
LOAM	26.77	11.21	4.63	2.80	6.44	10.11
LeGO-LOAM	20.09	6.84	2.28	1.19	5.03	8.23
本文算法	10.13	2.96	0.91	0.85	2.19	3.96

由表 2 可知,在不同序列的测试中,本文算法的 RMSE 均最小,比 LOAM 算法平均减少了 68.76%,比 LeGO-LOAM 算法平均减少了 50.55%,建图精度最高,在包含高速移动物体和多人密集交互场景的 Ours 数据集中,本文算法表现出更强的鲁棒性。实验结果表明,本文算法在复杂环境下仍具有优越的性能,能够有效提升轨迹估计的准确性。

2.3 算法实时性评估

在相同计算平台和开发环境下,对算法进行实时性测试。将 KITTI 数据集的 00、05、06、07 四个序列子集作为测试数据集,对动态物体剔除后,里程计姿态估计的频率可达到 10 Hz,建图模块的更新频率可达到 1 Hz。采集每个模块的运行时间并求取均值,得到平均运行时间。实验结果如表 3 所示。可见,本文算法可以实现实时运行。

表 3 KITTI 数据下各模块平均运行时间

Table 3 Average running time of each module on KITTI dataset

模块	平均运行时间/ms
预处理模块	47.21
里程计模块	16.84
建图模块	80.96
回环检测模块	31.52

使用 LeGO-LOAM 和 SuMa^[15]算法对同样的 KITTI 子集进行运算,采集里程计模块和建图模块的运行时间,并与本文算法对比,结果如表 4 所示。可见,LeGO-LOAM 和 SuMa 算法因未对动态环境进行提取,误将动态影响点云和静态背景点云同时处理,造成特征点的冗余,影响了各模块的运行效率。本文算法借助深度投影对动态目标处理后,减少了动态噪声的影响并精简了特征点数量,里程计模块和建图模块的运行时间最少,性能最优。

表 4 三种算法的各模块平均运行时间

Table 4 Average running time of each module for three algorithms

算法	平均运行时间/ms	
	里程计模块	建图模块
SuMa	25.11	124.07
LeGO-LOAM	22.05	116.81
本文算法	16.84	80.96

2.4 特征缺失场景下时间复杂度和收敛性分析

当环境特征缺失的情况发生时,会明显引起算法时间复杂度的变化。预处理模块的时间复杂度会因有效点减少由 $O(N)$ 变为 $O(N')$ ($N' < N$), N 和 N' 表示单帧点云数量;特征匹配模块的时间复杂度 $O(d)$ 会因匹配对减少导致 d 趋于 0,从而时间复杂度趋于 0, d 为匹配对数;地图构建模块的时间复杂度会因新贡献点云数量减少由

$O(M)$ 变为 $O(M')$ ($M' < M$), M 和 M' 表示地图点数;在图优化中,时间复杂度因为约束减少由 $O(s^{1.5})$ 上升至 $O(s^2)$, s 为约束数量。

特征退化不仅导致运行时间总体上升,还会使算法无法正常收敛,定位建图效果衰退。因此,本文通过改造 KITTI 数据集和 Ours 数据集的特征变量,验证本文算法在特征缺失场景下的收敛性边界条件。通过移除特征点云、添加动态遮挡面积、引入非结构化特征等方法控制实验变量,进行收敛性检测实验,得到表 5 所示的收敛性边界条件。

表 5 特征缺失场景下的收敛性边界条件

Table 5 Convergence boundary conditions under feature-deficient scenarios

参数	失效现象	有效范围	边界条件
边缘点密度/(pts·(°) ⁻¹)	旋转发散	≥5	<2
地面拟合残差/m	高度漂移	<0.1	>0.3
动态遮挡比例/%	位姿跳变	<40	>60
匹配重叠率/%	定位缺失	>30	<15

3 结论

本文对激光雷达输入的原始点云进行处理,通过深度投影实现了动态物体的识别与剔除,并利用地面特征、回环检测和里程计构建了因子图优化算法,生成了更一致的全局位姿和更精确的地图。与当前主流算法在公开数据集及自制数据集上的对比实验结果表明,本文方法在多种应用场景下均可识别并剔除动态物体干扰,表现出更高的定位和建图精度,实现了动态环境下 SLAM 系统性能的提升。

参考文献 (References):

[1] BAI Y N, SHAO S L, ZHANG J, et al. A review of brain-inspired cognition and navigation technology for mobile robots [J]. *Cyborg and Bionic Systems*, 2024, 5: 0128.

[2] 郑义桀, 罗健欣, 陈卫卫, 等. 单目视觉 SLAM 综述 [J]. *软件导刊*, 2022, 21(12): 242 - 251.

ZHENG Y J, LUO J X, CHEN W W, et al. Review of monocular visual SLAM [J]. *Software Guide*, 2022, 21(12): 242 - 251. (in Chinese)

[3] CHEN K, LOPEZ B T, AGHA-MOHAMMADI A A, et al. Direct LiDAR odometry: fast localization with dense point clouds [J]. *IEEE Robotics and Automation Letters*, 2022, 7(2): 2000 - 2007.

[4] VIZZO I, GUADAGNINO T, MERSCH B, et al. KISS-ICP: in defense of point-to-point ICP: simple, accurate, and robust

registration if done the right way [J]. *IEEE Robotics and Automation Letters*, 2023, 8(2): 1029 - 1036.

[5] ZHANG J, SINGH S. LOAM: Lidar odometry and mapping in real-time [C]// *Robotics: Science and Systems*. Berkeley, CA, USA: [s. n.], 2014, 2(9): 1 - 9.

[6] 隋心, 王思语, 罗力, 等. 基于点云特征的改进 RANSAC 地面分割算法 [J]. *导航定位学报*, 2024, 12(1): 106 - 114.

SUI X, WANG S Y, LUO L, et al. Improved RANSAC ground segmentation algorithm based on point cloud features [J]. *Journal of Navigation and Positioning*, 2024, 12(1): 106 - 114. (in Chinese)

[7] 刘砚菊, 晏佳华, 冯迎宾. 动态场景下基于视觉的 SLAM 技术研究 [J]. *半导体光电*, 2024, 45(2): 327 - 335.

LIU Y J, YAN J H, FENG Y B. SLAM based on deep learning and optical flow constraints in dynamic scenes [J]. *Semiconductor Optoelectronics*, 2024, 45(2): 327 - 335. (in Chinese)

[8] 王柯赛, 姚锡凡, 黄宇, 等. 动态环境下的视觉 SLAM 研究评述 [J]. *机器人*, 2021, 43(6): 715 - 732.

WANG K S, YAO X F, HUANG Y, et al. Review of visual SLAM in dynamic environment [J]. *Robot*, 2021, 43(6): 715 - 732. (in Chinese)

[9] 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.

[10] 兰凤崇, 田小强, 陈吉清, 等. 融合语义信息与物体级几何特征的实时动态激光 SLAM 算法 [J]. *汽车工程*, 2024, 46(11): 2028 - 2038.

LAN F C, TIAN X Q, CHEN J Q, et al. Real-time dynamic laser SLAM algorithm combining object-level geometric features and semantic information [J]. *Automotive Engineering*, 2024, 46(11): 2028 - 2038. (in Chinese)

[11] 杨颖, 王立勇, 孙鹏, 等. 动态环境下激光雷达地图构建 [J]. *计算机工程与设计*, 2023, 44(7): 2125 - 2131.

YANG Y, WANG L Y, SUN P, et al. LiDAR map construction in dynamic environment [J]. *Computer Engineering and Design*, 2023, 44(7): 2125 - 2131. (in Chinese)

[12] 杨娜, 程磊. 基于高效回环检测的大场景下的 SLAM 算法 [J]. *沈阳理工大学学报*, 2024, 43(2): 45 - 51.

YANG N, CHENG L. SLAM algorithm in large scenes based on efficient loop detection [J]. *Journal of Shenyang Ligong University*, 2024, 43(2): 45 - 51. (in Chinese)

[13] 孙伟, 曾豪霆, 张小瑞, 等. 基于点云强度和地面约束的大范围激光 SLAM [J]. *电子测量与仪器学报*, 2024, 38(4): 66 - 75.

SUN W, ZENG H T, ZHANG X R, et al. Large scale based on point cloud strength and ground constraints Lidar SLAM [J]. *Journal of Electronic Measurement and Instrumentation*, 2024, 38(4): 66 - 75. (in Chinese)

[14] SHAN T, ENGLOT B. LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain [C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE, 2018: 4758 - 4765.

[15] BEHLEY J, STACHNISS C. Efficient surfel-based SLAM using 3D laser range data in urban environments [C]// *Robotics: Science and System XIV*. Pittsburgh, PA, USA: MIT Press, 2018: 1 - 10.

(责任编辑: 宋颖韬)