

# 基于伪占用区在线点云去除的激光雷达 SLAM 算法

曾庆轩<sup>1</sup>, 李镛<sup>1\*</sup>, 聂为之<sup>2</sup>

<sup>1</sup>天津大学微电子学院, 天津 300072;

<sup>2</sup>天津大学电气自动化与信息工程学院, 天津 300072

**摘要** 提出一种使用点云记忆区来移除动态点云的激光雷达同时定位与建图(SLAM)算法——DOR-LOAM算法。该算法首先引入先验地图伪占用区域的概念,细化了动态点云记忆区的划分,从而提高了对动态点云的识别精度;然后,通过网格化每块伪占用区域,并设置点云概率密度阈值,增强算法对动态小目标的识别能力;最后,使用动态点云移除率优化滑窗方式,建立轻量级的先验地图,加快系统整体的处理速度。在KITTI开放数据集上的验证结果显示,DOR-LOAM算法移除动态点云的 $F_1$ 分数为95.52%,平均相对平移误差和相对旋转误差分别为0.81%和 $0.0033(^{\circ})\cdot\text{m}^{-1}$ ,该算法能实时、高效地消除环境中动态因素的影响。

**关键词** 遥感; 同时定位与建图; 动态物体去除; 激光雷达; 伪占用区域; 点云记忆区

中图分类号 TP242 文献标志码 A

DOI: 10.3788/AOS230839

## 1 引言

同时定位与建图(SLAM)技术已经在机器人和自动驾驶领域得到了广泛研究,并不断推动着SLAM算法的进步<sup>[1-3]</sup>。激光雷达在环境传感方面的优势,使其在SLAM领域受到广泛关注<sup>[4]</sup>。激光雷达SLAM中最重要的步骤就是点云匹配,现有的工作主要分为两类:原始点云匹配和特征点云匹配<sup>[5]</sup>。

在原始点云匹配中,迭代最近邻点(ICP)<sup>[6]</sup>是最常用的方法,ICP通过找到欧几里得空间中的最近邻点来迭代最小化对应点之间的残差距离,将两个点云之间的位姿转换收敛到最终位置。Deschard<sup>[7]</sup>提出的IMLS-SLAM,通过隐式移动最小二乘法计算相应点云之间的位姿,为每一个点云分配权重,但是这种方式需要庞大的计算量。

特征点云匹配方法主要是提取点云中的线、面特征,然后将这两种特征点云进行匹配。Zhang等<sup>[8]</sup>提出了LOAM系统,先根据不同激光束中的平滑度来获取原始点云中的线、面特征点,再构建线特征点到直线的匹配对和面特征点到平面的匹配对,从而实现位姿估计。Shan等<sup>[9]</sup>在特征提取之前进行了地面优化,通过提取地面点中的平面特征和非地面点的线条特征,滤除不相关的噪点,从而减少特征点的数量,实现更快速

的特征匹配。随着对点云特征处理方式越来越多样化,减少大量无效特征点不仅提升了系统的匹配速度,还提高了点云匹配的精度<sup>[10]</sup>。Wang等<sup>[11]</sup>提出一种更轻量的F-LOAM来连续提取较高平滑度的边缘特征和较低平滑度的平面特征,这些特征点在匹配过程中占据更高的权重,不仅减少了匹配点数量,还进一步提高了精度。

目前,去除动态物体是激光雷达SLAM领域中需要解决的重要问题,已经有很多学者对这个问题进行了深入研究<sup>[12-13]</sup>。张磊等<sup>[14]</sup>提出一种基于动态特征剔除点云与图像融合的位姿估计方法,通过自适应加权策略,有效融合了位姿结果。Chen等<sup>[15]</sup>设计了一种新颖的动态和静态对象分类器,同时考虑了对象的时间和空间一致性,在建图过程中有效地去除了动态对象,从而降低了激光里程计漂移。Chen等<sup>[16]</sup>通过构建语义地图可靠地移除动态对象,并增加语义约束优化扫描匹配,大幅提升了激光里程计的精度。这些方法在不同方面对激光雷达SLAM中去除动态物体的问题进行了探索,并取得了一定的成果。

为了消除环境中动态点云对SLAM系统建图定位精度的影响,本文提出一种基于伪占用区在线动态点云去除的激光雷达SLAM算法——DOR-LOAM(dynamic object removal-LOAM)。该算法使用当前

收稿日期: 2023-04-18; 修回日期: 2023-05-09; 录用日期: 2023-05-19; 网络首发日期: 2023-06-28

基金项目: 国家自然科学基金(61471263, 61872267)、天津市自然科学基金(16JCZDJC31100)、天津大学自主创新基金(2021XZC-0024)

通信作者: \*liqiang@tju.edu.cn

帧点云和先验地图匹配伪占用区域,通过对伪占用区域的判断,减少了逐块面元匹配带来的时间消耗,并且过滤只包含地面点云的面元区域,增强了提取动态点云的鲁棒性,并加快了提取速度;使用网格化伪占用区域的方法代替传统使用整块面元的匹配方法,增强了系统对同时包含动、静态物体区域的识别精度,在动态小目标场景中的识别效果尤为突出;引入基于动态移除率的滑动窗口方式来构建先验地图,大幅减少匹配动态点云带来的时间消耗;提出动态点云记忆区的概念,进一步减少重复点云匹配带来的资源消耗,使得动态点云移除工作能够实时运行,并集成到SLAM系统中。

## 2 DOR-LOAM算法原理

### 2.1 系统框架

所提出的DOR-LOAM算法整体框架如图1所示,主要分为4个阶段:特征提取、伪占用区域匹配记

忆区、构建滑动窗口和后端优化。

在特征提取阶段(图1中feature extraction模块),先通过连续提取周围点的集合计算曲率,再分别选出曲率大的边缘点和曲率小的平面点作为线特征和面特征。在伪占用区域匹配记忆区阶段(图1中pseudo-occupied region模块到dynamic memory area模块),首先通过面元来划分当前帧点云和地图点云的伪占用区域,然后提取地面点云并消除其对动静态物体判断的干扰,最后通过伪占用率来筛选动态点云并构造动态点云记忆区。在构建滑动窗口阶段(图1中build sliding window模块),首先引入滑动窗口形成轻量级的先验地图,然后设置动态点云移除率再次降低当前帧点云和先验地图匹配的时间消耗。在后端优化阶段(图1中matching static feature模块到pose optimize模块),通过对上述提取的静态点云进行特征筛选,完成后端位姿优化。

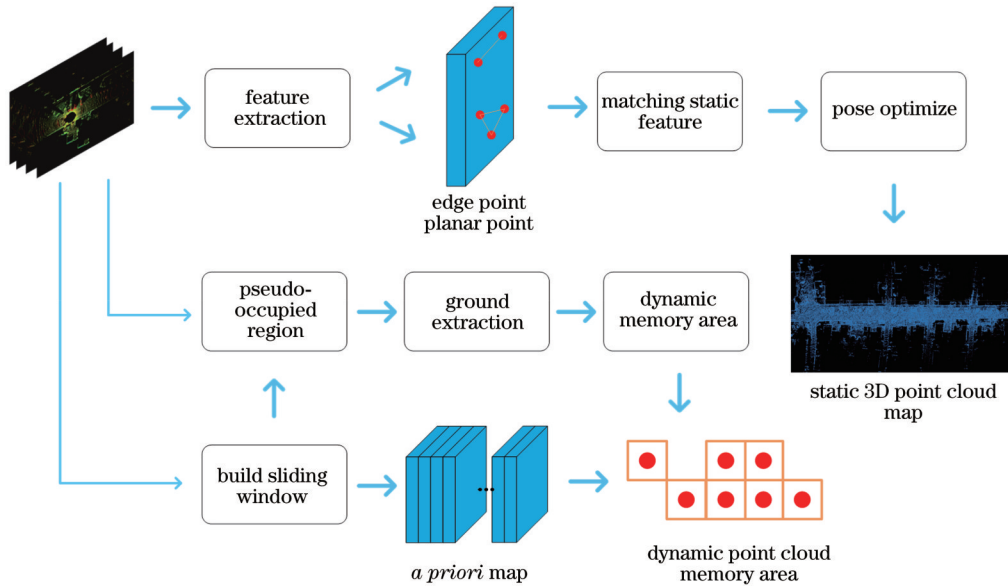


图1 DOR-LOAM系统整体框架

Fig. 1 Overall framework of DOR-LOAM system

### 2.2 伪占用区域划分与地面点云提取

用  $P_t = \{p_1, p_2, p_3, \dots, p_n\}$  表示  $t$  时刻单帧激光雷达扫描的一系列点云集合。在笛卡儿坐标系下,每一个点云的位置表示为  $p_k = \{x_k, y_k, z_k\}$ 。现将单帧点云  $P_t$  划分为  $N_s \times N_a$  个面元区域,其中  $N_s$  以每个主面元  $S_k$  的距离划分,  $N_a$  则把整个点云划分为多个主面元  $S_k$ , 由此可以得到面元的位置公式

$$B_{ij} = \left\{ p_k | p_k \in P_t, i = \frac{\text{atan} 2(y_k, x_k)}{\Delta\alpha}, j = \frac{N_s \cdot \sqrt{x^2 + y^2}}{D_{\max}} \right\}, (1)$$

式中:  $i, j$  表示面元所在的位置坐标;  $\Delta\alpha = 2\pi/N_a$  为相邻主面元  $S_k$  的偏移角度;  $D_{\max}$  为设定激光雷达能够探测的最远距离。

由于部分面元区域只包含地面点云,逐块面元匹配会带来多余的时间消耗,并且还会错误地提取地面点云作为动态物体。本文在面元划分的基础上提出伪占用区域的概念,减少逐块面元匹配的时间消耗和点云误匹配。伪占用区域的划分示意图如图2所示,图中方形为静态物体,三角为动态物体,定义面元未包含任何物体的区域为非占用区,面元存在物体的区域为伪占用区。伪占用区域中的点云可以分为3类:动态点云、静态点云和地面点云。由于地面点云会干扰算法对动态点云和静态点云的判断,因此准确提取伪占用区域的地面点云能够显著提升动态点云的识别精度。同时,复杂多变的地面会造成地面点云高度不一致,因此准确提取地面点云并识别动、静态点云是一大挑战<sup>[17]</sup>。

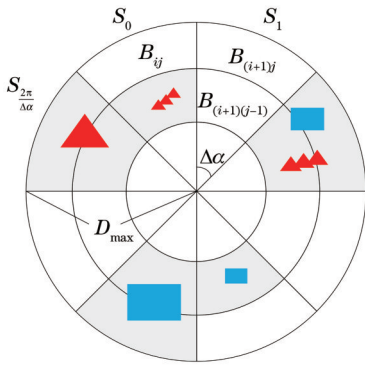


图 2 伪占用区域划分示意图

Fig. 2 Schematic of pseudo occupancy region division

用一维  $B_l$  来代替二维  $B_{ij}$  的位置,其含义为第  $l$  个伪占用区域。为了降低高度值较大的物体对伪占用区域进行地面提取的影响,设置一个和高度均值相关的阈值  $g(\bar{z}_l) \cdot h_g$  来限制单帧点云中地面点的高度,降低地面提取误差。最终可以根据阈值提取出含有地面点云的集合,表达式为

$$M_l = \{ \mathbf{p}_k | \mathbf{p}_k \in B_l, z_k < \bar{z}_l + g(\bar{z}_l) \cdot h_g \}, \quad (2)$$

式中:  $\bar{z}_l$  表示在  $B_l$  中点云的高度均值;  $z_k$  为  $\mathbf{p}_k$  点云中每个点云的高度;  $g(\cdot)$  为一个值域在  $[0, 1]$  区间的线性递减函数;  $h_g$  为线性递减函数  $g(\bar{z}_l)$  的可变系数。

假设一个点云集合  $M_l$  在单位空间中的协方差矩阵为  $C \in \mathbb{R}^{3 \times 3}$ , 那么其 3 个特征值  $\lambda_k$  及相应的特征向量  $\mathbf{v}_a$  的计算公式为

$$C\mathbf{v}_k = \lambda_k \mathbf{v}_k, \quad (3)$$

式中:  $k = 1, 2, 3$ 。计算后具有最小特征值的特征向量最有可能代表地面的法向量。

令特征值最小的法向量为  $\mathbf{n}_l = [a_l, b_l, c_l]^T$ , 其平面系数为  $d_l = -\mathbf{n}_l^T \bar{\mathbf{p}}_l$ , 因此提取地面点云的计算公式为

$$M_g = \{ \mathbf{p}_k | \mathbf{p}_k \in M_l, d_l - d_k < \tau_a \}, \quad (4)$$

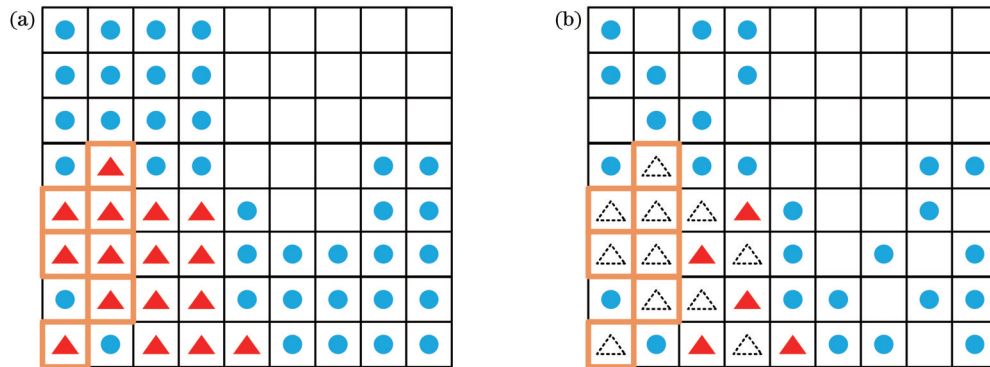


图 3 点云匹配形成动态点云记忆区示意图。(a)先验地图;(b)当前帧

Fig. 3 Schematic of dynamic point cloud memory area formed by point cloud matching. (a) *A priori* map; (b) current frame

当前帧点云稀疏的弊端导致不足以划分动态点云,可以假设当前帧伪占用区域的非地面点云为潜

式中:  $d_k = -\mathbf{n}_l^T \mathbf{p}_k$  表示每个点云的平面系数;  $\tau_a$  为点云平面边界的阈值。重复多次地面提取过程,即可获得准确度高的地面点云。同理,非地面点云为伪占用区域点云和地面点云的差集。

### 2.3 动态点云记忆区构建方法

为了定位当前帧和先验地图中的动态点云,需要将当前帧和先验地图的伪占用区域进行关联匹配。通过提取伪占用区域可以获得两类非地面点云:静态点云和动态点云。为了使当前帧和先验地图的非地面点云处于同一参考系,本文将世界坐标系(起始帧)作为点云的统一参考坐标。先验地图中每一伪占用区域的非地面点云的提取公式为

$${}^w M_l^m = {}^w B_l^m \cap {}^w T_m M_u^m, \quad (5)$$

当前帧每一伪占用区域的非地面点云为

$${}^w M_l^k = {}^w B_l^k \cap {}^w T_k M_u^k, \quad (6)$$

式中:  ${}^w M_l$  表示第  $l$  伪占用区域的动态点云集合;  $m, k$  分别代表先验地图和当前帧;  $M_u^m$  为先验地图的非地面点云;  $M_u^k$  为当前帧的非地面点云;  ${}^w T_k$  为当前帧到世界坐标系的变换矩阵;  ${}^w T_m$  为先验地图到世界坐标系的变换矩阵;  ${}^w T_k, {}^w T_m \in SE(3)$ ,  $SE(3)$  为特殊欧氏群。

在同一坐标系中,通过先验地图可以得到物体的丰富点云信息,但当前帧的物体信息不够丰富。图 3(a)所示为先验地图的非地面点云分布情况,图 3(b)所示为当前帧非地面点云的分布情况,其中实心三角点为动态点云,实心圆点为静态点云,三角点周围的粗线段表示动态点云记忆区。从数量和分布上看,先验地图的点云数量远远大于当前帧,并且动态点云分布也相对集中,这是因为动态移动物体具有时空不一致性,导致动态点云会在先验地图上留下一段轨迹,将这种现象称为“鬼迹效应”(ghost tracks)<sup>[18]</sup>。基于这一现象,本文提出一种网格化伪占用区域的方法来确定先验地图和当前帧点云的网格缺失密度,判断点云类别,提取动态点云形成动态点云记忆区域。

在动态点云。图 3 所示的虚线三角部分代表当前帧缺失的动态点云,而相应位置的先验地图动态点云



不存在缺失。不仅是动态点云,静态点云也存在缺失的情况,这是由点云稀疏化造成的,缺失的密度会受到限制,因此可以根据缺失密度来判断动静态点云,得到

$$P_m = \{p_k \in {}^w M_i^m | \rho({}^w M_i^m \oslash {}^w M_i^k) > \gamma_p\}, \quad (7)$$

$$\rho(x_i) = \frac{1}{\sqrt{2\pi} \sqrt{|\Sigma_m|}} \exp\left[-\frac{(x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m)}{2}\right], \quad (8)$$

式中: $P_m$ 为动态记忆区的点云集合; $\rho(x_i)$ 为网格内点云概率密度计算函数; $\oslash$ 表示两个点云集合 ${}^w M_i^m$ 和 ${}^w M_i^k$ 的差集运算; $\gamma_p$ 为伪占用率; $\mu_m$ 和 $\Sigma_m$ 分别为先验地图每个网格的均值和协方差。由于动、静态点云概率密度有一定差异,可以通过设定伪占用率阈值判断点云类别。

## 2.4 基于动态点云移除率的滑窗方式

Lim等<sup>[19]</sup>提出全量地图构建先验信息,以匹配动态异常点,这会带来巨大的时间消耗。在当前帧与地图匹配的过程中,并不会使用到全部过程的点云,所以使用全量地图查找只会增加时间消耗。本文提出一种滑动窗口的方式连续、实时地加入每一帧点云,并在一定数量后去除最后帧的点云。这种滑窗式的先验地图能够消除构建离线先验地图带来的时间损耗,并且不会降低动态物体移除精度。

在形成一系列的点云滑窗前,这些点云经过初始的点云匹配可以获得初始位姿。本文根据初始的位姿可以将滑窗内的一系列点云拼接成一幅先验地图。如图4所示,当有新的帧加入,当前帧就会与先验地图进行伪占用区域匹配,获取动态点云位置,形成动态点云记忆区。在移除动态点云后则过滤掉最老的一帧,加入最新帧,并等待下一帧的输入。

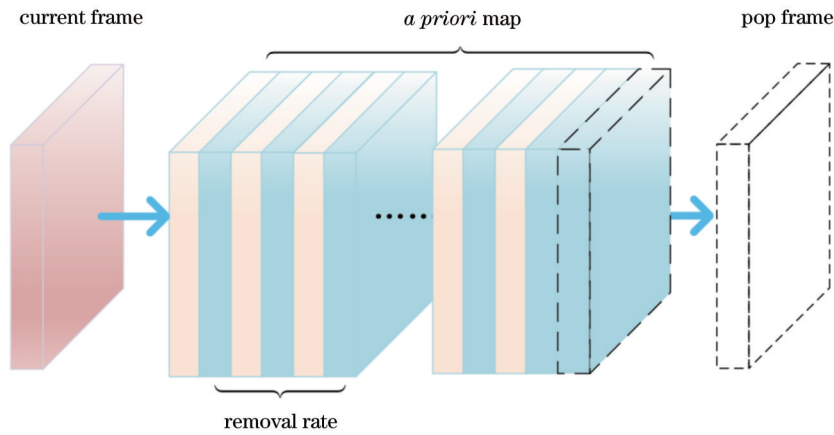


图4 先验地图的滑窗方式示意图

Fig. 4 Schematic of sliding window mode of a priori map

在这一过程中,激光雷达输出点云的频率在10 Hz内,一段时间后先验地图的点云有重复信息,冗余的点云会增加性能消耗。为了进一步减少时间消耗,本文提出了动态点云移除率 $\eta$ ,这一参数,通过在点云帧加入先验地图的步长来减少先验地图的点云数量,以提升计算速度。经过实验验证,在先验地图的构造过程中,引入动态点云移除率可以提高系统的性能。

## 2.5 后端点云匹配优化

采取LOAM中的后端点云匹配ICP算法,通过匹配先验地图和当前帧点云来获取优化后的位姿,并对当前帧和先验地图都进行了处理,其中包括动态点云移除。在匹配过程中只添加静态点云作为优化项目,而不加入动态点云,可以构建出ICP误差函数公式,即

$$e = \arg \min_{\xi} \left[ \sum_{i=0}^n \|p_i^s - \exp(\hat{\xi}) p_i^s\|_2^2 + \sum_{i=0}^n \|p_i^c - \exp(\hat{\xi}) p_i^c\|_2^2 \right], \quad (9)$$

式中: $\xi$ 表示待优化的位姿; $\hat{\xi}$ 表示预估计的位姿; $p_i^s$ 表示线特征点; $p_i^c$ 表示面特征点; $p_i^s$ 和 $p_i^c$ 表示与其匹配的线、面特征点。由于模型中涉及两次匹配,帧对帧匹配过程和帧对地图匹配过程中均使用上述方法,加入静态边缘点和静态平面点的特征来优化最终位姿,这两种匹配过程分别选取其相应的静态特征点。

## 3 实验与分析

### 3.1 实验环境与数据准备

在公共数据集KITTI odometry benchmark<sup>[20]</sup>上验证所提算法。KITTI是在自动驾驶平台上录制的,该平台配备了4台立体摄像机、Velodyne HDL-64E激光雷达和高精度OXTS RT 3003 GPS/INS定位系统。其中摄像机、激光雷达和定位系统已经完成标定和同步。数据集包含11个训练序列,并且所有传感器的数据均已采集。本文根据计算的里程计和KITTI提供的里程计真值来计算所有序列的相对平移和相对旋转误差。此外,本文使用移除率 $R_R$ 、保留率 $R_P$ 和 $F_1$ 分数

定量评估动态点云移除的准确度。

实验在配备了 Interl i7-1165G7 CPU(2.80 GHz)、16 GB RAM 和 Ubuntu18.04 操作系统的微型主机上运行。激光雷达的数据发送频率为 10 Hz,相邻帧的时间间隔在 100 ms 以内,能够保证实时处理。

### 3.2 评价指标

#### 3.2.1 动态物体移除精度评估标准

动态物体移除精度主要通过  $R_p$ 、 $R_R$  和  $F_1$  分数 3 个指标表征,计算公式为

$$\begin{cases} R_p = \frac{\omega_s}{\eta_s} \\ R_R = 1 - \frac{\omega_d}{\eta_d} \\ F_1 = \frac{2 \times (R_p \times R_R)}{R_p + R_R} \end{cases}, \quad (10)$$

式中: $\omega_s$ 为保留的静态点云数; $\eta_s$ 为静态点云总数; $\omega_d$ 为保留的动态点云数; $\eta_d$ 为动态点云总数。本文采用 0.2 的体素率来处理所有点云并进行比较。

#### 3.2.2 激光里程计精度评估标准

采用 KITTI metric<sup>[21]</sup>对不同建图算法的精度进行评价,KITTI metric 分别计算了相对平移误差和相对旋转误差,计算公式为

$$E_{\text{trans}} = \frac{1}{|f|} \sum_{(i,j) \in f} \frac{\mu_{\text{trans}}[(\hat{p}_j \odot p_i) \odot (p_j \odot p_i)]}{d_{\text{length}}(i,j)}, \quad (11)$$

$$E_{\text{rot}} = \frac{1}{|f|} \sum_{(i,j) \in f} \frac{\mu_{\text{rot}}[(\hat{p}_j \odot p_i) \odot (p_j \odot p_i)]}{d_{\text{length}}(i,j)}, \quad (12)$$

式中: $d_{\text{length}}(i,j)$ 表示  $i$ 、 $j$  两帧之间的固定距离; $\mu_{\text{rot}}(\cdot)$ 和  $\mu_{\text{trans}}(\cdot)$ 分别为  $p \in SE(3)$  的旋转量和平移量; $|f|$ 表示  $i$ 、 $j$  之间的帧数; $\odot$ 用于计算两帧之间的相对位姿; $\hat{p}_j$ 和  $p_j$ 分别表示位姿的估计值和真实值。为了反映激光里程计在不同距离的精度,计算了 100~800 m 距离的平移误差和相对旋转误差的平均值。

### 3.3 结果与分析

#### 3.3.1 动态点云移除精度

Removert<sup>[22]</sup>是移除动态点云的经典开源算法,其采用删除-还原的机制,先尽可能地删除动态点,再通过多分辨率深度图像来建立当前帧和地图之间的关联以恢复误删点,这种方式容易误删地面点,存在一定的局限性。ERASOR 算法<sup>[19]</sup>先对当前帧和全局地图进行面元划分,再通过逐块计算二者的面元高度差比率来判断伪占用区和动态区域,这种方式将同时包含动态和静态点云的区域简单地划分为一类,并且由于需要预先构建全局地图,其只能离线运行。本文算法(DOR-LOAM 算法)弥补了 ERASOR 算法的缺陷:首先,通过滑动窗口构建先验地图来代替全局地图,保证算法的实时性能;然后,采用点云概率密度来区分点云类别,并非 ERASOR 算法中对区域的简单划分;最后,

通过点云记忆区来进一步加强对动态点云的筛选。

本文选取了 KITTI 数据集中的 5 段序列,对 DOR-LOAM、Removert 和 ERASOR 算法的  $R_p$ 、 $R_R$  和  $F_1$  分数进行定量比较。如表 1 所示,DOR-LOAM 算法的  $R_p$  和  $F_1$  分数平均值分别为 94.13% 和 95.52%,相比于最新的开源算法 ERASOR,所提算法的  $R_p$  和  $F_1$  分数分别提升了 4.00% 和 1.81%。

表 1 不同算法的动态点云移除精度对比

Table 1 Comparison of dynamic point cloud removal accuracy for different algorithms unit: %

Sequence	Evaluation index	Removert	ERASOR	DOR-LOAM
00	$R_p$	85.50	93.98	<b>97.29</b>
	$R_R$	<b>99.35</b>	97.08	97.06
	$F_1$	91.90	95.50	<b>97.18</b>
01	$R_p$	94.22	91.49	<b>96.45</b>
	$R_R$	93.61	<b>95.38</b>	94.87
	$F_1$	93.91	93.40	<b>95.65</b>
02	$R_p$	76.32	<b>87.73</b>	84.27
	$R_R$	96.80	97.01	<b>97.30</b>
	$F_1$	85.35	<b>92.10</b>	90.32
05	$R_p$	86.90	88.73	<b>95.33</b>
	$R_R$	87.88	<b>98.26</b>	97.53
	$F_1$	87.39	93.30	<b>96.42</b>
07	$R_p$	80.69	90.62	<b>97.30</b>
	$R_R$	<b>98.82</b>	99.27	98.80
	$F_1$	88.84	94.80	<b>98.04</b>
Average	$R_p$	84.726	90.51	<b>94.13</b>
	$R_R$	95.292	<b>97.40</b>	97.11
	$F_1$	89.478	93.82	<b>95.52</b>

本文选取序列 00 和 07 来比较不同算法的动态点云去除结果,如图 5 所示,其中图 5(a1)~(d1)为序列 00 的结果,图 5(a2)~(d2)为序列 07 的结果。可以看到,Removert 算法会残留大部分动态点云,ERASOR 和 DOR-LOAM 算法能较为干净地滤除动态物体,但在小区域 ERASOR 算法没有达到好的效果,这是因为 ERASOR 算法直接处理整块伪占用区域的潜在动态点云,将同时包含动、静态物体的伪占用区域简单地划分为静态或动态中的一类,导致小区域的动态点云被划分为静态点云。DOR-LOAM 算法计算了伪占用区域中不同物体的概率密度,并设置合适的阈值来建立动态点云记忆区,能够有效地清除图 5 小方框中的动态点云。

DOR-LOAM 算法对  $R_p$  指标的提升显著。图 6 所示为序列 05 的动态点云提取结果,从上到下的 3 幅图分别表示静态点云图、静态点云细节图、动态点云图。通过和真值对比,ERASOR 算法的静态点云细节图中地面点云缺失严重,这是因为 ERASOR 算法对每一块



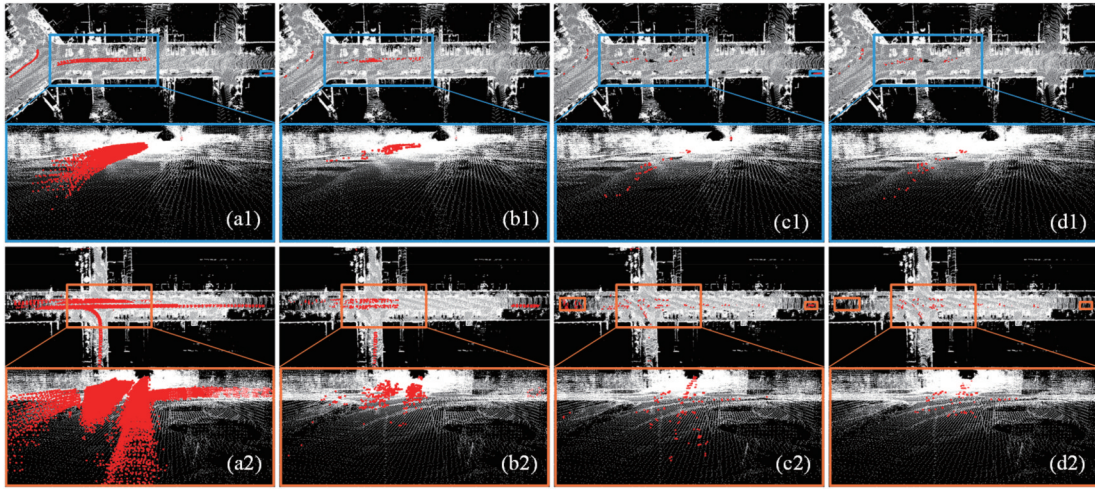


图 5 不同算法对序列 00 和序列 07 动态点云的移除效果。(a)原始图;(b) Removert 算法;(c) ERASOR 算法;(d) DOR-LOAM 算法  
 Fig. 5 Sequence 00 and sequence 07 dynamic point cloud removal effect for different algorithms. (a) Original iamges; (b) Removert algorithm; (c) ERASOR algorithm; (d) DOR-LOAM algorithm

面元都匹配筛选,导致点云提取错误。DOR-LOAM 算法通过先划分面元区域再确定伪占用区,之后提取

地面点云,较完美地减少了对无效面元区域地面点云的误判。

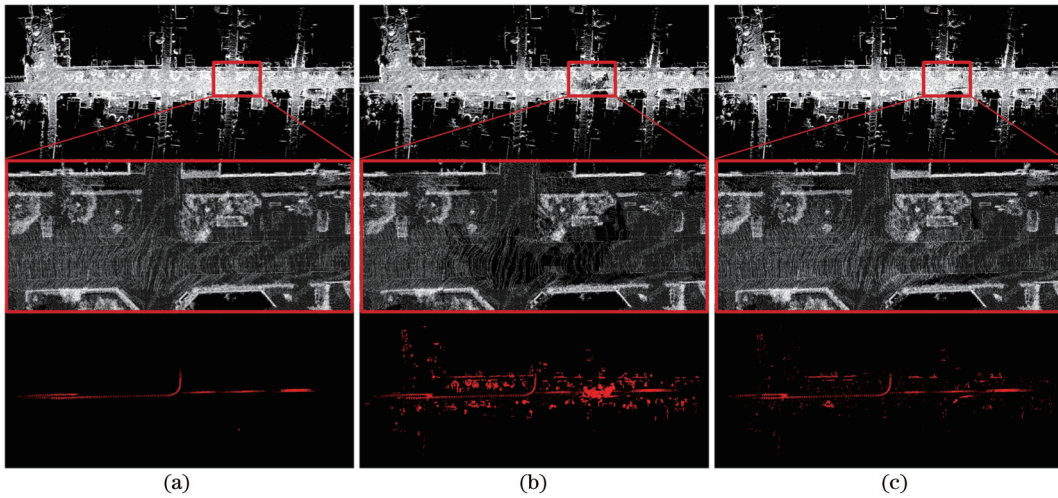


图 6 不同算法对序列 05 的点云移除效果细节图。(a)真值;(b) ERASOR 算法;(c) DOR-LOAM 算法  
 Fig. 6 Detail of point cloud removal effect of sequence 05 for different algorithms. (a) Ground truth; (b) ERASOR algorithm; (c) DOR-LOAM algorithm

### 3.3.2 激光里程计精度

将上述动态物体移除算法运用到后端点云匹配上,以提高点云匹配的精度。为了验证 DOR-LOAM 算法的精度和效率,在 KITTI 数据集上进行实验,分别对比了 E-LOAM<sup>[23]</sup>、T-LOAM<sup>[10]</sup>、NDT-LOAM<sup>[24]</sup>、PSF-LO<sup>[15]</sup> 和 SuMa++<sup>[16]</sup> 等算法在 100~800 m 的平均相对平移误差  $\bar{E}_{trans}$  和平均相对旋转误差  $\bar{E}_{rot}$ 。其中:E-LOAM、T-LOAM 和 NDT-LOAM 均为传统的纯激光雷达 SLAM 算法,也是最新开源的纯激光雷达 SLAM 算法;PSF-LO 和 SuMa++ 是基于深度学习网络提取点云语义信息的 SLAM 算法,其不仅通过语义信息来移除动态物体,还增加了语义约束来提高定位精度。本文采用 E-LOAM<sup>[23]</sup>、T-LOAM<sup>[10]</sup>、NDT-

LOAM<sup>[24]</sup>、PSF-LO<sup>[15]</sup> 和 SuMa++<sup>[16]</sup> 的实验数据,与 DOR-LOAM 的计算结果进行对比,结果如表 2 所示。

从表 2 可以看出:相比于 E-LOAM、T-LOAM 和 NDT-LOAM 算法,DOR-LOAM 算法的平均相对平移误差分别降低了 0.75、0.12、0.09 百分点;PSF-LO 和 SuMa++ 算法在移除动态物体的基础上,增加了语义约束,其整体性能要高于 DOR-LOAM 算法。其中,SuMa++ 算法在训练深度学习模型的过程中,使用序列 08 作为验证集,剩余序列作为训练集,故 SuMa++ 算法在序列 08 上的性能要低于本文算法。此外,PSF-LO 和 SuMa++ 算法需要使用额外的 GPU 资源进行计算,因此在通用性上 DOR-LOAM 算

表 2 不同算法精度的对比情况  
Table 2 Accuracy comparison of different algorithms

Sequence	E-LOAM		T-LOAM		NDT-LOAM		PSF-LO		SuMa++		DOR-LOAM	
	$\bar{E}_{trans} / \%$	$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$	$\bar{E}_{trans} / \%$	$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$	$\bar{E}_{trans} / \%$	$\bar{E}_{trans} / \%$	$\bar{E}_{trans} / \%$	$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$	$\bar{E}_{trans} / \%$	$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$	$\bar{E}_{trans} / \%$	$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$
00	1.17	0.0047	0.98	0.0060	0.79	<b>0.64</b>	<b>0.64</b>	<b>0.0022</b>	0.82	0.0028		
01	2.92	0.0062	2.09	0.0052	1.46	<b>1.32</b>	1.60	<b>0.0046</b>	1.83	0.0050		
02	2.36	0.0082	1.01	0.0039	1.09	<b>0.87</b>	1.00	0.0037	1.11	<b>0.0036</b>		
03	1.16	0.0068	1.10	0.0082	<b>0.65</b>	0.75	0.67	0.0046	<b>0.65</b>	<b>0.0043</b>		
04	1.39	0.0050	0.68	0.0068	<b>0.31</b>	0.66	0.37	<b>0.0026</b>	0.59	0.0038		
05	0.82	0.0036	0.55	0.0032	0.54	0.45	<b>0.40</b>	<b>0.0020</b>	0.54	0.0023		
06	1.34	0.0057	0.56	0.0031	0.56	0.47	<b>0.46</b>	<b>0.0021</b>	0.48	0.0023		
07	1.21	0.0066	0.50	0.0047	<b>0.27</b>	0.46	0.34	<b>0.0019</b>	0.37	0.0025		
08	1.63	0.0063	0.94	0.0033	1.04	0.94	1.10	0.0035	<b>0.93</b>	<b>0.0030</b>		
09	1.36	0.0057	0.80	0.0040	0.74	0.56	<b>0.47</b>	<b>0.0023</b>	0.71	0.0028		
10	1.84	0.0063	1.12	0.0061	1.12	<b>0.54</b>	0.66	<b>0.0028</b>	0.92	0.0037		
Average	1.56	0.0059	0.93	0.0049	0.90	0.74	<b>0.70</b>	<b>0.0029</b>	0.81	0.0033		

法具有更大的优势。

为了更加直观地显示 DOR-LOAM 算法的计算结果和真值之间的差异,绘制了序列 00~09 这 10 个序列的轨迹图。从图 7 和表 2 可以看出,在动态物体点云较

多的场景中,如序列 03、序列 04 和序列 07 等, DOR-LOAM 算法的平均相对平移误差和平均相对旋转误差较低,能够较好地贴合真值。

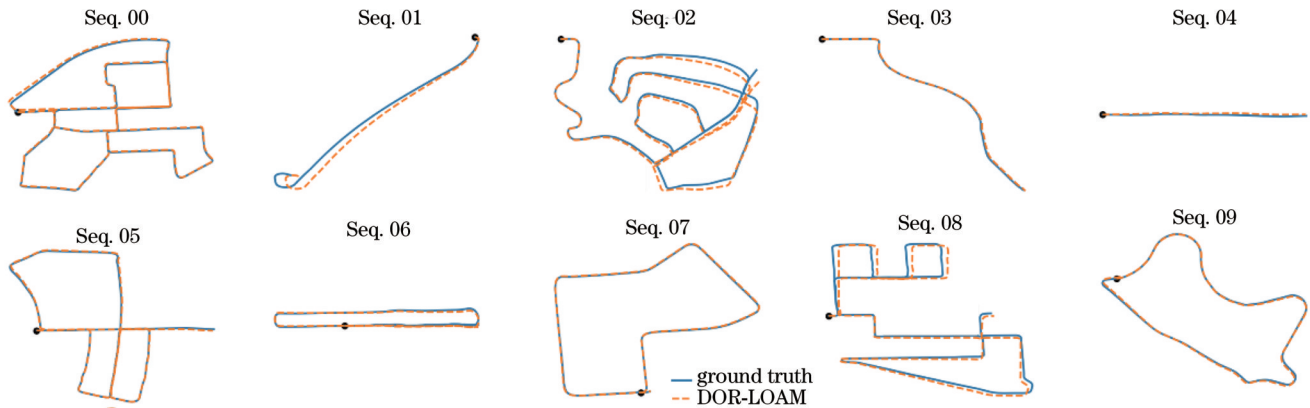


图 7 序列 00 到序列 09 的轨迹对比

Fig. 7 Comparison of tracks from sequence 00 to sequence 09

### 3.3.3 动态点云去除模块消融实验

为了探究动态点云移除模块 DOR 对激光里程计的影响,将移除 DOR 模块后的 DOR-LOAM 作为基准模型 Baseline,并与 E-LOAM、T-LOAM 和 NDT-LOAM 算法进行对比,得到表 3 所示的实验结果。从表 3 可以看出,Baseline 的平均相对平移误差和平均相对旋转误差分别为 1.28% 和 0.0044  $(\circ) \cdot m^{-1}$ ,在加入 DOR 模块后,虽然平均耗时增加了 11.15 ms,但是平均相对平移误差减小了 0.47 个百分点,平均相对旋转误差减小了 0.0011  $(\circ) \cdot m^{-1}$ ,证明了动态点云移除模块 DOR 能有效提升激光里程计的精度。此外,在耗时性能上,Baseline+DOR 的平均消耗时间低于 T-LOAM 算法,高于 E-LOAM 算法,均满足低于 100 ms 的实时

性能要求。

表 3 消融实验与不同算法对比情况

Table 3 Comparison of ablation experiments and different algorithms

Algorithm	Baseline	Baseline +DOR	E-LOAM	T-LOAM	NDT-LOAM
$\bar{E}_{trans} / \%$	1.28	0.81	1.56	0.93	0.90
$\bar{E}_{rot} / [(\circ) \cdot m^{-1}]$	0.0044	0.0033	0.0059	0.0049	—
Average time /ms	76.33	87.48	77.38	100	—

### 3.3.4 不同实验参数对比

为使 DOR-LOAM 算法达到最佳性能,设计了多



组实验来挑选出性能更好、实时性更佳参数。选取不同的动态移除率参数  $\eta_r = 1, 2, 4, 8$  作为候选值, 计算平均平移误差、平均旋转误差和平均每帧计算消耗时间, 实验结果如图 8 所示。

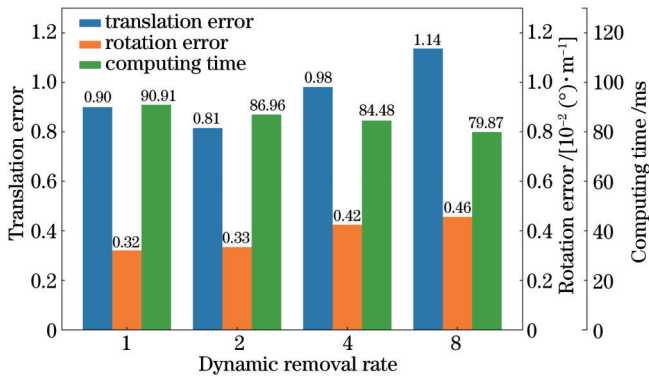


图 8 动态移除率参数对比

Fig. 8 Comparison of dynamic removal rate parameters

从图 8 可以看出, 动态移除帧率越大, 每帧计算消耗的时间越短, 平均旋转误差越大。待匹配的帧步长

越大, 系统对动态点云的位置越不敏感, 从而造成动态点云漏选, 影响里程计精度。平均平移误差呈现 V 字形, 在动态移除帧率为 1 时并未展现出最好的性能, 主要原因有如下两点: 1) 点云获取频率为 10 Hz, 逐帧匹配会过度选取点云, 导致  $R_p$  值降低; 2) 逐帧匹配会重复选取点云, 导致计算时间增加。当动态移除帧率设置为 2 时, 平均平移误差最小, 之后随着动态移除帧率增大, 平均平移误差也增大。实验结果表明, 动态移除帧率设置为 2 时, 既能保证较高的建图精度, 又能拥有更快的处理速度。

### 3.3.5 运行时间性能验证

DOR-LOAM 在测试阶段也取得了较好的实时性能。在整体系统方案中, 主要设置了 4 个线程来处理任务, 包括特征提取、位姿初优化、位姿后优化和动态点云滤除。本文随机选取 4500 帧点云的计算消耗时间, 绘制了图 9 所示的各模块时间消耗图, 从上到下分别为每帧消耗时间、平均每帧消耗时间、位姿优化消耗时间、动态点云滤除消耗时间和特征提取消耗时间。平均每帧消耗时间为 87.48 ms, 在图 9 中用点黑线表示, 满足小于 100 ms 的实时性能要求。

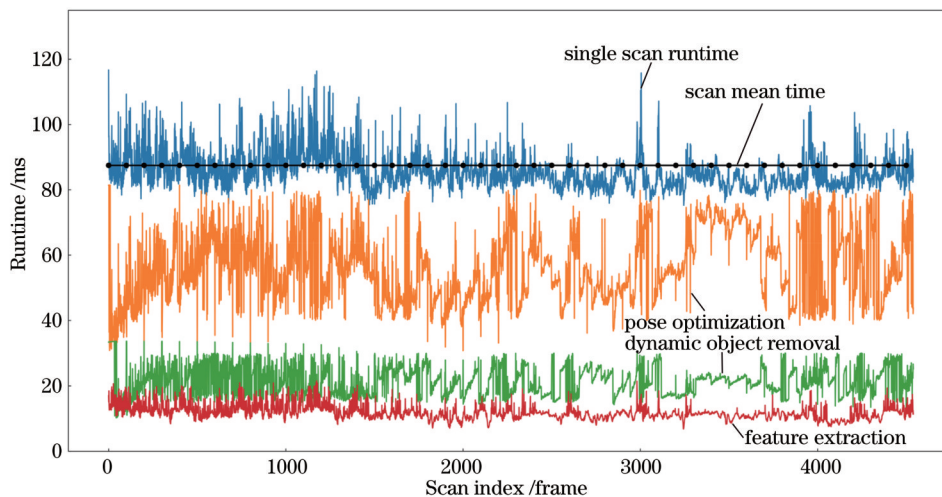


图 9 各模块运行时间比较

Fig. 9 Comparison of running time of each module

## 4 结 论

针对传统激光雷达 SLAM 算法在不同动态场景下提取点云线、面特征时, 错误提取非静态特征会影响点云匹配精度的问题, 提出一种在线移除动态点云的 DOR-LOAM 算法。该算法基于动态点云的时空不一致性特点, 滤除非相关动态点云, 旨在低成本、高效地消除其对激光里程计的影响, 以适应复杂多变的场景。在 KITTI 数据集上的验证结果显示, 动态点云平均移除率  $R_r$  和静态点云平均保留率  $R_p$  分别达到 97.11% 和 94.13%, 平均保留率  $R_p$  相比最新的开源算法提高了 4.00%。DOR-LOAM 算法不仅在动态点云的移除

上取得了较好的性能, 还提高了激光里程计的精度, 相比于 E-LOAM、T-LOAM 和 NDT-LOAM 算法, DOR-LOAM 算法的平均相对平移误差分别降低了 0.75、0.12、0.09 个百分点, 同时减少动态点云移除模块的计算时间消耗, 使得系统的实时性能得到保证。可见, DOR-LOAM 算法可以应用到更加复杂多变的场景中。

### 参 考 文 献

- [1] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): part II [J]. IEEE Robotics & Automation Magazine, 2006, 13(3): 108-117.
- [2] 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.
- [3] Fuentes-Pacheco J, Ruiz-Ascencio J, Rendón-Mancha J M. Visual simultaneous localization and mapping: a survey[J]. Artificial Intelligence Review, 2015, 43(1): 55-81.
- [4] 周治国, 曹江微, 邸顺帆. 3D 激光雷达 SLAM 算法综述[J]. 仪器仪表学报, 2021, 42(9): 13-27.  
Zhou Z G, Cao J W, Di S F. Overview of 3D lidar SLAM algorithms[J]. Chinese Journal of Scientific Instrument, 2021, 42(9): 13-27.
- [5] 王忠立, 李文仪. 基于点云分割的运动目标跟踪与 SLAM 方法[J]. 机器人, 2021, 43(2): 177-192.  
Wang Z L, Li W Y. Moving objects tracking and SLAM method based on point cloud segmentation[J]. Robot, 2021, 43(2): 177-192.
- [6] Besl P J, McKay N D. A method for registration of 3D shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239-256.
- [7] Deschaud J E. IMLS-SLAM: scan-to-model matching based on 3D data[C]//2018 IEEE International Conference on Robotics and Automation (ICRA), May 21-25, 2018, Brisbane, QLD, Australia. New York: IEEE Press, 2018: 2480-2485.
- [8] Zhang J, Singh S. Low-drift and real-time lidar odometry and mapping[J]. Autonomous Robots, 2017, 41(2): 401-416.
- [9] Shan T X, 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), October 1-5, 2018, Madrid, Spain. New York: IEEE Press, 2019: 4758-4765.
- [10] Zhou P W, Guo X X, Pei X F, et al. T-LOAM: truncated least Squares LiDAR-only odometry and mapping in real time[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-13.
- [11] Wang H, Wang C, Chen C L, et al. F-LOAM: fast LiDAR odometry and mapping[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 27-October 1, 2021, Prague, Czech Republic. New York: IEEE Press, 2021: 4390-4396.
- [12] 常耀辉, 陈年生, 饶蕾, 等. 动态环境下具有旋转和平移不变性的激光雷达点云描述子[J]. 光学学报, 2022, 42(24): 2401007.  
Chang Y H, Chen N S, Rao L, et al. Lidar point cloud descriptor with rotation and translation invariance in dynamic environment[J]. Acta Optica Sinica, 2022, 42(24): 2401007.
- [13] 徐陈, 周怡君, 罗晨. 动态场景下基于光流和实例分割的视觉 SLAM 方法[J]. 光学学报, 2022, 42(14): 1415002.  
Xu C, Zhou Y J, Luo C. Visual SLAM method based on optical flow and instance segmentation in dynamic scene[J]. Acta Optica Sinica, 2022, 42(14): 1415002.
- [14] 张磊, 徐孝彬, 曹晨飞, 等. 基于动态特征剔除的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.
- [15] Chen G B, Wang B S, Wang X L, et al. PSF-LO: parameterized semantic features based lidar odometry[C]//2021 IEEE International Conference on Robotics and Automation (ICRA), May 30-June 5, 2021, Xi'an, China. New York: IEEE Press, 2021: 5056-5062.
- [16] Chen X, Milioto A, Palazzolo E, et al. SuMa: efficient LiDAR-based semantic SLAM[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 3-8, 2019, Macau, China. New York: IEEE Press, 2020: 4530-4537.
- [17] Himmelsbach M, Hundelshausen F V, Wuensche H J. Fast segmentation of 3D point clouds for ground vehicles[C]//2010 IEEE Intelligent Vehicles Symposium, June 21-24, 2010, La Jolla, CA, USA. New York: IEEE Press, 2010: 560-565.
- [18] Pagad S, Agarwal D, Narayanan S, et al. Robust method for removing dynamic objects from point clouds[C]//2020 IEEE International Conference on Robotics and Automation (ICRA), May 31-August 31, 2020, Paris, France. New York: IEEE Press, 2020: 10765-10771.
- [19] Lim H, Hwang S, Myung H. ERASOR: egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2272-2279.
- [20] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: the KITTI dataset[J]. International Journal of Robotics Research, 2013, 32(11): 1231-1237.
- [21] Chen Y B, Huang S D, Zhao L, et al. Cramér-Rao bounds and optimal design metrics for pose-graph SLAM[J]. IEEE Transactions on Robotics, 2021, 37(2): 627-641.
- [22] Kim G, Kim A. Remove, then Revert: static point cloud map construction using multiresolution range images[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 24 2020-January 24 2021, 2021, Las Vegas, NV, USA. New York: IEEE Press, 2020: 10758-10765.
- [23] Guo H L, Zhu J K, Chen Y P. E-LOAM: LiDAR odometry and mapping with expanded local structural information[J]. IEEE Transactions on Intelligent Vehicles, 2023, 8(2): 1911-1921.
- [24] Chen S B, Ma H, Jiang C H, et al. NDT-LOAM: a real-time lidar odometry and mapping with weighted NDT and LFA[J]. IEEE Sensors Journal, 2022, 22(4): 3660-3671.

## Lidar SLAM Algorithm Based on Online Point Cloud Removal in Pseudo Occupied Area

Zeng Qingxuan<sup>1</sup>, Li Qiang<sup>1\*</sup>, Nie Weizhi<sup>2</sup>

<sup>1</sup>School of Microelectronics, Tianjin University, Tianjin 300072, China;

<sup>2</sup>School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

### Abstract

**Objective** When the robot or autonomous vehicle is driving, since the lidar will detect dynamic objects such as

pedestrians and cars, these dynamic point clouds are stacked in the constructed map, which affects the accuracy of point cloud registration. Therefore, removing the dynamic point clouds of lidar has profound significance for improving the mapping and localization accuracy of the SLAM algorithm. Currently, most of the algorithms for removing dynamic point clouds are offline, which needs to compare the differences between the current frame and the map to determine the dynamic point cloud. These methods need to store many point clouds at historical moments, which require high computing and storage resources. As a result, it is impossible to determine the point cloud category in real time, and the removal efficiency of dynamic point clouds is affected. We propose a lidar SLAM algorithm that can remove dynamic point clouds online, which can both remove dynamic point clouds and improve the accuracy of lidar odometry.

**Methods** Aiming at the problem that dynamic objects in the environment affect the mapping accuracy and efficiency, we propose a lidar SLAM algorithm DOR-LOAM to remove dynamic point clouds online. Firstly, the algorithm employs the current frame and the prior map to match the pseudo-occupancy area and divides the pseudo-occupancy area block by block for reducing the time consumption caused by the matching process. Secondly, it filters the bin regions that only contain ground point clouds, enhancing the robustness and speed of removing dynamic point clouds. Thirdly, it replaces the traditional method of utilizing whole-area matching and proposes a method of gridded pseudo-occupied areas. This enhances the recognition accuracy of the algorithm for areas containing both dynamic and static objects, and the recognition effect for small dynamic objects is particularly prominent. Fourthly, it adopts a sliding window method based on a dynamic removal rate to construct *a priori* map, greatly reducing the time consumption caused by matching dynamic point clouds. Meanwhile, the concept of dynamic point cloud memory area is proposed to further reduce the cost of repeated point cloud matching, making dynamic point cloud removal work in real-time.

**Results and Discussions** The DOR-LOAM algorithm is evaluated on the KITTI dataset, and we assess the algorithm's point cloud removal accuracy and lidar odometry accuracy respectively. In terms of point cloud removal accuracy, the static point cloud preserve rate  $P_R$ , dynamic point cloud removal rate  $R_R$ , and  $F_1$  score reach 94.13%, 97.11%, and 95.52% respectively. Compared with the latest open-source algorithm ERASOR, the  $P_R$  of DOR-LOAM improves by 4.00% (Table 1), and due to the utilization of gridded pseudo-occupied areas, the removal effect of small dynamic objects is better than other advanced algorithms (Fig. 6). In terms of lidar odometry, the relative translation error and relative rotation error are 0.81% and  $0.0033 (^{\circ}) \cdot \text{m}^{-1}$  respectively. Compared with the E-LOAM, T-LOAM, and NDT-LOAM algorithms, the relative translation error of DOR-LOAM is reduced by 0.75, 0.12, and 0.09 percent point respectively (Table 2). Additionally, an ablation experiment is carried out on the algorithm, which verifies that the dynamic point cloud removal module can improve the lidar odometry accuracy (Table 3). In terms of the time consumption performance of the system, the algorithm greatly reduces the time consumption of point cloud matching through the point cloud memory area and the construction of sliding windows. The average time consumption per frame is 87.48 ms, which can meet real-time performance (Fig. 9).

**Conclusions** The traditional lidar SLAM algorithm extracts point cloud line and surface features in different dynamic scenes, and the wrong extraction of non-static features will affect the point cloud matching accuracy. Thus, we propose a DOR-LOAM algorithm for the online removal of dynamic point clouds. Based on the spatiotemporal inconsistency of dynamic point clouds, the algorithm filters out non-correlated dynamic point clouds to eliminate its influence on lidar odometry at low cost and high efficiency and adapt to complex and changeable scenes. The verification results on the KITTI dataset show that the  $F_1$  score of the algorithm for removing dynamic point clouds is 95.52%, and the relative translation error and relative rotation error are 0.81% and  $0.0033 (^{\circ}) \cdot \text{m}^{-1}$  respectively. The proposed algorithm not only yields better performance in the removal of dynamic point clouds but also improves the accuracy of the lidar odometry.

**Key words** remote sensing; simultaneous localization and mapping; dynamic object removal; lidar; pseudo occupancy area; point cloud memory area