

DOI: 10.19650/j.cnki.cjsi.J2513706

基于多阶段动态过滤的静态点云地图生成算法

范明泽, 徐晓苏

(东南大学仪器科学与工程学院 南京 210096)

摘要:在动态场景中,诸如行人、车辆等动态对象出现在传感器的观测范围内,对静态点云地图的构建带来了显著干扰。这些动态目标在地图的生成过程中常常留下“鬼影”,严重影响地图构建精度与完整性。为了解决该问题,提出了一种面向复杂动态场景的高精细静态点云地图生成算法,旨在保持地图构建精度的同时,有效剔除动态干扰点。首先,对点云进行序列化处理并结合区域化地面分割,以减少地面对动态判别的干扰。然后,设计了一种多阶段的动态点云离线滤除策略。在动态识别的前两阶段,分别采用两种分布式描述符(D-POD与D-PODV)来描述点云的空间占用和分布情况,并结合扫描比率测试(SRT)和径向比率测试(RRT),实现对强弱动态点云的精准识别。在第3阶段,利用基于区域密度的改进自适应DBSCAN聚类算法完善对不规则动态点云的滤除。在公开数据集SemanticKITTI上的实验结果表明,所提算法能够在多种复杂场景下准确滤除动态点并生成高精细静态点云地图。与当前主流动态点云滤除算法ERASOR和Removert相比,所提算法的平均保留率分别提升3.95%和14.56%,平均拒绝率分别提升13.44%和17.46%。通过对强弱动态点云进行分阶段滤除,该研究能够有效滤除各类动态目标,同时最大限度地保留原始点云中的静态信息,确保了全局地图的结构完整性,为实现多场景通用、高精度、高可靠性的静态点云地图构建提供了有力支撑。

关键词: 分布式描述符;多阶段动态滤除;静态地图构建;自适应DBSCAN聚类

中图分类号: TP242.6 TH701 文献标识码: A 国家标准学科分类代码: 520.60

Multi-Stage dynamic filtering-based static point cloud map generation algorithm

Fan Mingze, Xu Xiaosu

(School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: In dynamic environments, the presence of moving objects such as pedestrians and vehicles within the sensor's field of view causes significant interference in the creation of static point cloud maps. These dynamic objects often result in "ghost" artifacts, severely affecting the accuracy and completeness of the generated maps. To tackle this challenge, this paper proposes a high-precision algorithm for static point cloud map generation specifically designed for complex dynamic scenarios. The goal is to effectively eliminate dynamic interference points while preserving map construction accuracy. The method begins by serializing multi-frame point cloud data and applying region-based ground segmentation to minimize the impact of ground points on dynamic point identification. A multi-stage offline dynamic point cloud filtering strategy is then implemented. In the first two stages of dynamic point identification, two distributed descriptors (D-POD and D-PODV) are used to capture the spatial occupancy and distribution patterns of the point cloud. These descriptors are combined with the scan ratio test (SRT) and radial ratio test (RRT) to precisely identify both strongly and weakly dynamic points. In the third stage, an improved adaptive DBSCAN clustering algorithm, based on regional density, is employed to further refine the removal of irregular dynamic point clouds. Experimental results using the publicly available SemanticKITTI dataset show that the proposed algorithm effectively filters out dynamic points, resulting in high-precision static point cloud maps across diverse complex scenarios. Compared to state-of-the-art dynamic point cloud filtering algorithms, ERASOR and Removert, the proposed method achieves average improvements of 3.95% and 14.56% in static point retention, and 13.44% and 17.46% in dynamic point rejection, respectively. By employing staged filtering for both strongly and weakly dynamic point clouds, the proposed method successfully eliminates various types of dynamic objects while maximizing the preservation of static information in the original point cloud. This

ensures the structural integrity of the global map, providing robust support for the creation of high-precision and highly reliable static point cloud maps across a wide range of applications.

Keywords: distributed descriptors; multi-stage dynamic filtering; static map construction; adaptive DBSCAN clustering

0 引言

同步定位与地图构建 (simultaneous localization and mapping, SLAM) 作为无人驾驶领域的重要支撑技术^[1-3], 通过融合多源传感器数据以构建环境的三维地图。传统的 SLAM 技术大多针对静态环境设计, 在静态可控场景中的适应性较好。但在复杂室外场景下易受行人和车辆等动态物体的干扰, 这些动态物体在全局地图中留下的“幽灵痕迹”^[4] 不仅降低了地图质量, 而且严重影响智能车后续的导航和决策任务。

相机通过捕捉环境中丰富的视觉特征 (如颜色和纹理), 结合图像序列分析完成动态静态物体的区分。而激光雷达由于缺乏颜色和纹理信息, 仅依靠空间中的三维点云数据, 当动态与静态物体外形相似时, 容易产生混淆。尤其在点云帧数较少或帧间变化不足的情况下, 其动态物体识别能力受到明显限制, 这一局限性使激光 SLAM 在动态场景下的地图构建面临更大挑战。为实现复杂动态场景下的鲁棒定位与高精度地图构建, 激光 SLAM 需具备有效处理动态物体的能力。然而, 在建图过程中动态点往往被错误地关联到全局静态地图上, 导致建图精度下降, 不利于后续基于地图的导航定位^[5-6]。主流解决方法包括在线动态过滤和离线动态过滤。在线方法在地图生成过程中实时移除动态物体, 离线方法在地图完全生成后进行, 能够对地图进行更为深入的分析与处理, 达到理想的动态滤除精度。鉴于此, 研究工作聚焦于离线场景下的动态点云识别与过滤。

在线动态过滤^[7-9]是在点云数据采集的同时, 实时检测并滤除环境中的动态物体, 依赖于当前时刻的传感器数据。离线过滤方法是在地图构建完成后, 通过多帧点云对比以及特征分析实现动态点云的精准识别。由于不受实时性限制, 离线方法可以综合利用更多的历史点云数据进行全面分析, 从而显著提高动态物体检测的准确性和鲁棒性。以离线方法生成静态点云地图有 3 种主流方法: 光线投影法、视点可见性法以及分割方法。Schauer 等^[10]提出的基于光线投影的离线处理技术, 通过构建全局体素网格地图并判断栅格是否先被激光击中后再被穿过, 实现动态点云的识别, 但在遮挡环境中容易误判静态物体。Kim 等^[11]提出了基于视点可见性的 Removert 算法, 通过高分辨率深度图移除动态点, 随后降低分辨率恢复静态点来构建地图, 缺点是容易误杀地面点且对位姿精度敏感。Chen 等^[12]提出的 DORF (dynamic object

removal framework) 算法延续了 Removert 的恢复思想, 采用由粗到精的方式移除潜在动态点, 并在拥挤的行人环境中证实了鲁棒性。Pagad 等^[13]根据已配准的 3D 点云构建体素占用图并设计了占用率更新策略, 在城市道路上展现了良好的动态滤除效果。Milioto 等^[14]对原始激光点云进行语义分割并借助推断的标签实现动态过滤, 但此类方法依赖于监督标签和训练数据。Lim 等^[15]提出的 ERASOR (egocentric ratio of pseudo occupancy-based dynamic object removal) 算法利用伪占用描述符实现了对静态点云地图的精细化处理。该方法对剧烈运动和运动模糊具有鲁棒性, 但在动静物体重叠的场景下识别效果受限。

针对上述方法存在的不足, 受文献^[15]的启发, 在 ERASOR 算法的基础上设计了一种改进方案, 旨在更加有效地应对复杂动态场景并生成更加精细且准确的静态点云地图。设计了一种多阶段动态点云滤除策略: 第 1 阶段采用伪占用描述符结合扫描比率测试识别强动态点云; 第 2 阶段利用多维分布式描述符结合径向比率测试识别弱动态点云; 第 3 阶段设计了一种基于区域密度的改进自适应聚类方法, 用于进一步完善对不规则动态点云的识别。该方法是在基于密度的噪声应用空间聚类算法^[16-17] (density-based spatial clustering of applications with noise, DBSCAN) 的基础上改进得到的。最终, 构建高精度的全局静态点云地图。

1 基于多阶段动态过滤的静态地图构建

1.1 算法总体框架

所提出算法的整体框架如图 1 所示, 主要包括两个模块:

1) 地图点云预处理模块: 首先, 通过传感器订阅原始激光雷达点云并对点云进行去噪和降采样操作, 利用位姿变换融合连续帧点云, 以构建初始全局地图。首先, 对点云进行序列化处理, 借助文献^[18]的思想将其划分为等间隔的待处理段区。接着, 根据点云的三维坐标进行潜在动态区域划分。最后, 为了尽可能减少地面对动态对象识别的干扰, 引入“区域化”思想来实现精准的地面分割^[19]。

2) 动态点云滤除模块: 动态识别的核心思想是对查询帧点云和子地图点云中相应的段区进行对比。第 1 阶段的目的是筛选仅包含动态点云的强动态段区。采用分布式伪占用描述符 (distributed-pseudo occupancy descriptor, D-POD), 结合扫描比率测试 (scan ratio test, SRT) 来检查两个对应段区的点云差异。通过计算扫描

比率,进而筛选出仅包含动态点云的强动态段区。其余段区中则包含绝对静态段区和弱动态段区。第 2 阶段的目的是筛选出弱动态段区。设计了一种多维分布式伪占用描述符向量(distributed-pseudo occupancy descriptor vector, D-PODV),结合径向比率测试(radial ratio test, RRT)进行点云差异检测,通过计算径向比率,筛选得到

同时包含动静态点云的弱动态段区。第 3 阶段的目的是进一步完善弱动态点云。设计了一种基于区域密度的改进自适应 DBSCAN 聚类算法,以合并三维空间中足够接近的动态点,进而实现对不规则弱动态点云的识别与滤除。最后,构建不含动态痕迹的全局静态点云地图。

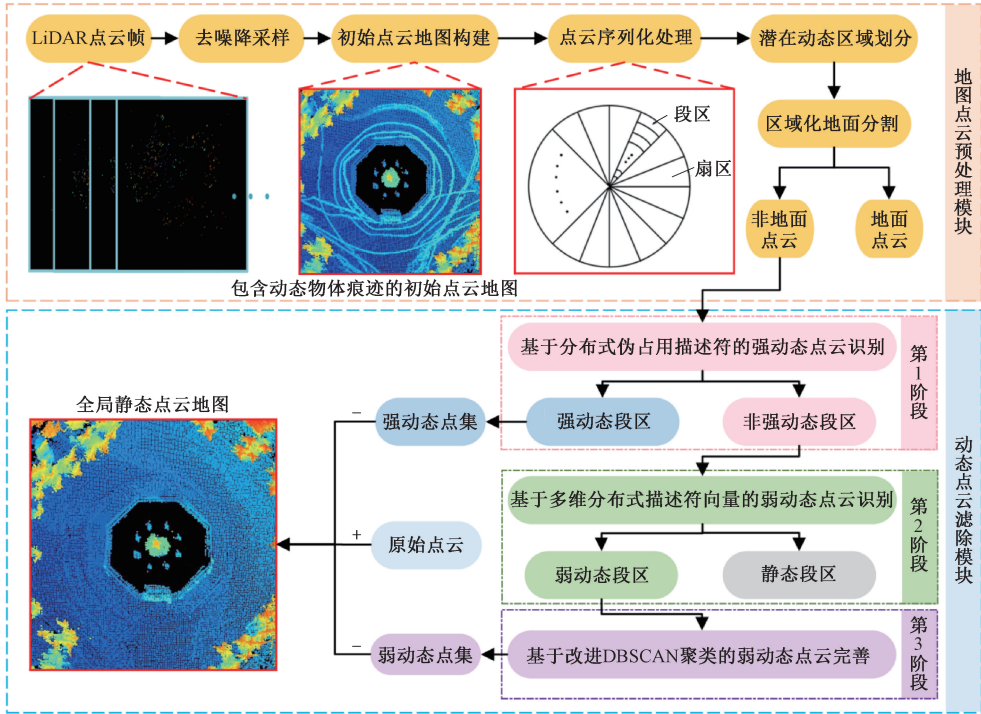


图 1 算法框架

Fig. 1 The algorithm frame

1.2 地图点云预处理

全局地图点云的预处理操作主要包括初始点云地图构建、点云序列化处理、潜在动态区域划分以及区域化地面分割 4 个部分。核心目的是提高原始点云处理效率,同时提升关键区域及目标点云的处理精度,有效提升后续动态目标的判定效率。

1) 初始点云地图构建:对原始点云数据进行点云去噪和体素滤波降采样后,将每一帧点云累积到全局点云地图中,生成待优化的初始点云地图,如式(1)所示。

$$M = \bigcup_{t \in T} {}^w_L T P_t^L \quad (1)$$

其中, M 表示待处理的初始点云地图, P_t^L 表示时刻 t 激光雷达坐标系下的点云帧, T 表示时间集合, ${}^w_L T$ 表示点云帧对应的 SE(3) 位姿。

2) 点云序列化处理:通过对初始点云进行划分,将空间中的无序点云划分为有序点云。首先,在初始点云地图中根据指定范围划定相对于查询帧点云 P_t^L 的子地图,

并统一坐标系。然后,对查询帧和相应的子地图点云进行结构化处理。

如图 2 所示,将点云投影到极坐标平面上,以极坐标的方式将空间划分为方位上的等间隔扇区,将扇区划分为径向上的等间隔段区。

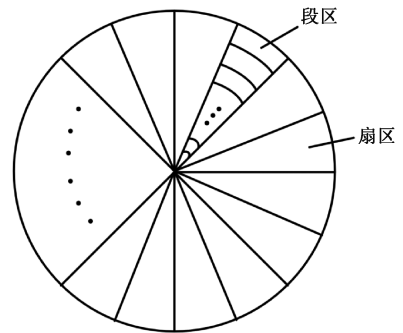


图 2 点云序列化示意图

Fig. 2 Diagram of point cloud serialization

计算点云 \mathbf{p}_k 对应的扇区索引 $segment_{p(k)}$ 以及段区索引 $bin_{p(k)}$, 计算公式如式(2)~(4)所示。

$$segment_{p(k)} = \frac{\arctan(y_k, x_k)}{L_\theta} \quad (2)$$

$$bin_{p(k)} = \sqrt{(x_k^2 + y_k^2)} / L_r \quad (3)$$

$$N_\theta = 2\pi / L_\theta, N_r = Li_{\max} / L_r \quad (4)$$

其中, (x_k, y_k, z_k) 为点云 \mathbf{p}_k 的三维空间坐标, 满足 $0 \cdot \sqrt{(x_k^2 + y_k^2)} \cdot Li_{\max}, Li_{\max}$ 为雷达最大探测距离, L_θ 表示角度分辨率, N_θ 表示扇区总数, L_r 表示径向分辨率, N_r 表示段区总数。

段区划分具体如式(5)所示。

$$B_t^X = \bigcup_{i \in N_r, j \in N_\theta} b_{(i,j),t}^X \quad (5)$$

其中, X 为查询帧和子地图的区分标志, $b_{(i,j),t}^X$ 表示时刻 t 下的第 (i,j) 个段区。

3) 潜在动态区域划分: 由于环境中大部分物体为静态物体, 如果处理扫描到的全部点云将大大造成计算资源的浪费。因此, 针对智能车行驶过程中可能遇到的动态对象类别, 对点云进行潜在动态区域划分, 集中资源处理感兴趣区域, 如式(6)所示。

$$V_t = \{ \mathbf{p}_k \mid \mathbf{p}_k \in P_t, \rho_k < R_{\max}, h_{\min} < z_k < h_{\max} \} \quad (6)$$

其中, V_t 为潜在动态区域的通用表示, P_t 表示在时刻 t 采集的点云帧, $\rho_k = \sqrt{x_k^2 + y_k^2}, R_{\max} = 80.0 \text{ m}, h_{\min} = -1.0 \text{ m}, h_{\max} = 3.0 \text{ m}$, 分别表示潜在动态区域的最大径向边界及最小最大高度边界。

4) 区域化地面分割: 由于动态物体通常与地面接触, 为提高对动态目标的识别精度, 需要对地面点进行有效滤除, 以降低地面点对动态识别过程的干扰。考虑到室外环境中的地面情况复杂多样(如坡度、台阶、不规则起伏等), 若仅依赖单一区域点云进行地面分割, 容易受到局部地形变化的影响, 导致分割精度下降。因此, 所提算法引入了“区域化”思想, 通过结合邻域段区信息, 增强地面分割的鲁棒性和精度。具体而言, 借助文献[20]的思想并结合“区域化”策略, 以待分割段区为中心, 结合周围8个邻域段区进行地面分割。结合中心段区及八邻域段区中具有最小 Z 向高度的点云, 对中心段区点云进行地面估计, 如式(7)所示。

$$b_{(i,j)}^X = b_{(i,j)}^C \cup b_{(i,j)}^{\bar{C}} \quad (7)$$

其中, $b_{(i,j)}^C$ 表示段区地面点集, $b_{(i,j)}^{\bar{C}}$ 表示段区分割点集。与全局地图的单一地面估计模型相比, 基于单段区的地面拟合更具有鲁棒性, 能够准确分割平缓地面以及坡面。

1.3 多阶段动态点云滤除

在完成对初始地图点云的预处理操作后, 设计了一种多阶段动态点云滤除策略, 旨在完成复杂场景下的动

态对象识别与滤除。具体步骤为:

阶段1: 强动态点云识别。第1阶段通过高度描述符对段区进行快速粗筛选, 旨在筛选出仅包含动态点云的强动态段区。采用分布式伪占用描述符 D-POD 来描述每个段区中点云的分布特征, 以段区内点云的平均最大 Z 向高程差作为描述符, 表示段区的空间点云伪占用率, 描述符如式(8)所示。

$$\text{D-POD: } \overline{\Delta h_{(i,j),t}^X} = \overline{Z_{(i,j),t}^{\max}} - \overline{Z_{(i,j),t}^{\min}} \quad (8)$$

其中, $\overline{\Delta h_{(i,j),t}^X}$ 表示第 (i,j) 个段区的平均 Z 向高程差, 将段区中的点云按 Z 向高度大小进行排序, $\overline{Z_{(i,j),t}^{\max}}$ 表示该段区中 Z 向高度值最大的5个点的平均 Z 轴值, $\overline{Z_{(i,j),t}^{\min}}$ 表示该段区中 Z 向高度值最小的5个点的平均 Z 轴值, X 为查询帧和子地图的区分标志。

采用 SRT 来检测查询帧点云和子地图点云中两个对应段区的点云差异, 扫描比率指两个对应段区之间的伪占用率比例, 基于该扫描比率可以从所有段区中筛选得到强动态段区, 即该段区内仅包含地面之上的绝对动态点云。扫描比率计算公式如式(9)所示。

$$S_r = \min \left\{ \frac{\overline{\Delta h_{(i,j),t}^O}}{\overline{\Delta h_{(i,j),t}^M}}, \frac{\overline{\Delta h_{(i,j),t}^M}}{\overline{\Delta h_{(i,j),t}^O}} \right\} \quad (9)$$

其中, S_r 表示扫描比率, 判定阈值如式(10)所示。

$$\begin{cases} S_r < \xi_{\text{threshold}}^D, & \text{强动态} \\ S_r > \xi_{\text{threshold}}^{ND}, & \text{非强动态} \end{cases} \quad (10)$$

其中, $\xi_{\text{threshold}}^D, \xi_{\text{threshold}}^{ND}$ 分别表示强动态阈值和非强动态阈值。

当扫描比率接近1时, 说明两个段区中的点云高程差几乎没有发生变化, 属于非强动态情况; 当扫描比率 $\ll 1$ 时, 说明两个段区中的点云高程差发生明显变化, 段区中仅包含绝对动态点云, 属于强动态情况。将强动态段区中的点云加入动态点集中, 当段区中仅有少量点云时, 跳过 SRT。

阶段2: 弱动态点云识别。经过第1阶段的粗筛选后, 未识别的弱动态段区通常是动静态点云的混合段区。为了更精准地识别弱动态点云, 需要依赖更高分辨率的细节信息和特征表达。因此, 将段区按径向距离和方位角划分为等间隔的子段区, 如式(11)所示。

$$b_{(i,j),t}^X = \bigcup_{d \in N_{bin}} subin_{d,t}^X \quad (11)$$

其中, $subin_{d,t}^X$ 表示第 d 个子段区, N_{bin} 表示划分的子段区数目, 根据段区内点云坐标将其分配到各个子段区内。

弱动态点云通常分布稀疏且动态特征不够明显, 仅依赖阶段一的单一描述符难以有效识别弱动态点云。为更全面地反映弱动态点云的局部分布特征, 设计了一种

多维分布式伪占用描述符向量 D-PODV 来描述子段区的点云占用率,如式(12)~(13)所示。

$$\text{D-PODV: SubBin}_{(i,j),t}^X = [\bar{\rho}_1, \bar{\rho}_2, \bar{\rho}_3, \dots, \bar{\rho}_{N_{bin}}] \quad (12)$$

$$\bar{\rho} = \frac{1}{N_d} \sum_{m=1}^N \rho_m \quad (13)$$

其中, $\text{SubBin}_{(i,j),t}^X$ 为第 (i,j) 个段区的分布式描述符向量, $\bar{\rho}$ 为对应子段区点云在极坐标平面上的平均径向距离, N_d 为点云总数, ρ_m 为点云 p_m 的径向距离。

采用 RRT 对查询帧点云和子地图点云中各个对应子段区进行点云差异检测。在同一视线方向上,仅能看到距离最近的物体,后方被遮挡的物体无法被直接探测。通过比较激光束所在射线方向上扫描到的平均点云径向距离差,能够有效判断对应子段区中是否存在动态点云。以子段区 $\text{subin}_{d,t}^X$ 为例,该子段区对应的径向比率 R_d 的计算公式如式(14)所示。

$$R_d = \min \left\{ \frac{\bar{\rho}_d^Q}{\bar{\rho}_d^M}, \frac{\bar{\rho}_d^M}{\bar{\rho}_d^Q} \right\} \quad (14)$$

其中, $\eta_{\text{threshold}}$ 为径向比率阈值,当径向比率 $R_d < \eta_{\text{threshold}}$ 时,说明该子段区内存在动态点云,属于弱动态段区,将该子段区中的动态点云加入动态点集合中。

段区 $b_{(i,j),t}^X$ 的整体径向比率向量如式(15)所示。

$$S_y = \min \frac{\text{SubBin}_{(i,j),t}^Q}{\text{SubBin}_{(i,j),t}^M} = \{R_1, R_2, \dots, R_{N_{bin}}\} \quad (15)$$

其中, $\min \cdot$ 表示取描述符向量 $\text{SubBin}_{(i,j),t}^Q$ 和 $\text{SubBin}_{(i,j),t}^M$ 各维度的径向比率最小值, S_y 表示包含两描述符在各个维度最小比率值的 N_{bin} 维向量。

阶段3:弱动态点云完善。考虑到弱动态点云通常具有复杂的形状或模糊的边界,前两阶段以段区为最小单元的识别方法可能存在少量动态点云漏检的情况,尤其对于体积较小的动态物体。此类动态物体由于点云稀疏或遮挡等原因,容易被忽略。为进一步提升弱动态点云的识别精度,所提算法采用 DBSCAN 聚类方法,旨在完善对部分遗漏动态点云的检测。由于实际场景中不同区域的动态点云密度可能稠密或稀疏,传统的固定参数聚类方法容易导致误检或漏检情况,并且难以处理复杂形状的点云簇。所提算法在传统 DBSCAN 聚类的基础上进行了改进,设计了一种基于区域密度的自适应 DBSCAN 聚类方法。根据点云的局部区域密度对聚类参数进行动态调整,该方法能够更准确地划分动态点云边界,在提升聚类精度的同时,有效减少误检与漏检情况的发生。

首先,基于阶段2中所有弱动态段区内被判定为动态的子段区,计算子段区内非地面点云的质心 $c_{(i,j)}^d = [\bar{x}_d, \bar{y}_d, \bar{z}_d]^T$, 计算公式如式(16)所示。

$$\begin{cases} c_{(i,j)}^d = \frac{1}{N_d} \sum_{m=1}^{N_d} p_m \\ p_m = [x_m, y_m, z_m]^T \end{cases} \quad (16)$$

其中, $c_{(i,j)}^d$ 为段区 $b_{(i,j),t}^X$ 中第 d 个动态子段区的点云质心, N_d 为子段区内点云总数, p_m 表示激光点坐标, \bar{x}_d , \bar{y}_d , \bar{z}_d 分别表示子段区内点云坐标平均值。

将所有动态子段区的质心点作为核心节点加入核心点列表中,并以核心点为中心进行基于区域密度的空间聚类,以合并三维空间中足够接近的动态点。自适应 DBSCAN 聚类算法通过分析核心点所在窗口区域的密度,动态调整邻域半径 Eps 和最小点数 minPts ,以适应不同密度的点云分布,同时有效处理复杂形状的动态点云簇。如图3所示,以核心点 A 所在段区为中心,设置窗口尺寸 $\text{size}_{\text{window}} = 3$,划分其所属的窗口区域。

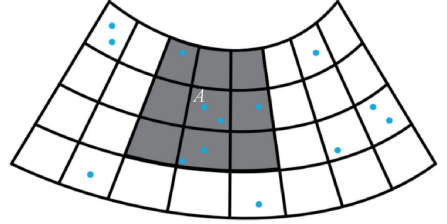


图3 核心点所属窗口区域示意图

Fig. 3 Diagram of the core point's window region

根据点云分布特性和具体场景设定基础邻域半径 Eps_{base} 并计算基础最小点数 $\text{minPts}_{\text{base}}$, 计算公式如式(17)所示。

$$\text{minPts}_{\text{base}} = \frac{1}{N_{\text{core}}} \sum_{i=1}^{N_{\text{core}}} P_i \quad (17)$$

其中, N_{core} 表示核心点总数, P_i 表示第 i 个核心点对应的邻域内包含的点云个数。

在此基础上对邻域半径 Eps 和最小点数 minPts 进行自适应调整,计算公式如式(18)和(19)所示。

$$Eps_i = Eps_{\text{base}} \times \alpha \times \beta \times \varepsilon \quad (18)$$

$$\begin{cases} \alpha = \left(1 + \frac{1}{1 + N_{\text{core}}^i} \right) \\ \beta = \left(1 + \frac{\sigma_i}{Eps_{\text{base}}} \right) \\ \varepsilon = (1 + |\nabla_{\text{density}}^i|) \end{cases} \quad (19)$$

其中, Eps_i 表示第 i 个核心点的邻域半径, α , β , ε 分别表示核心点密度修正系数、分布修正系数、局部密度变化系数, N_{core}^i 表示第 i 个核心点所属窗口区域内的核心点总数, σ_i 表示第 i 个核心点所属窗口区域内的核心点分布标准差,其计算公式如式(20)~(22)所示。

$$\sigma_i = \sqrt{\frac{1}{N_{\text{core}}^i} \sum_{j=1}^{N_{\text{core}}^i} (d_j - \bar{d})^2} \quad (20)$$

$$d_j = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (21)$$

$$\bar{d} = \frac{1}{N_{\text{core}}^i} \sum_{j=1}^{N_{\text{core}}^i} d_j \quad (22)$$

其中, $[x_i, y_i, z_i]$ 为核心点 i 的三维坐标, $[x_j, y_j, z_j]$ 为窗口区域内其余核心点的三维坐标, d_j 表示其余核心点到当前核心点的距离, \bar{d} 为平均距离。

局部密度变化系数 ε 中的 $\nabla_{density}^i$ 为局部密度梯度, 计算公式如式(23)所示。

$$\nabla_{density}^i = \frac{\max(Num_{bin}) - \min(Num_{bin})}{size_{window}} \quad (23)$$

其中, Num_{bin} 为窗口区域内子段区中的核心点数量, $size_{window}$ 为窗口尺寸。

第 i 个核心点对应邻域的最小点数 $minPts_i$ 的计算公式如式(24)~(26)所示。

$$minPts_i = \max[\minPts_{base}, X, Y] \quad (24)$$

$$X = \frac{\minPts_{base} \times N_{core}^i}{1 + \sigma_i} \quad (25)$$

$$Y = |\nabla_{density}^i| \times \mu_{density} \quad (26)$$

其中, $\mu_{density}$ 表示密度因子。

对于每个核心点, 根据其所属段区划分对应的窗口区域并计算邻域半径 Eps_i , 该半径内的所有点视为邻域点, 若邻域点数目超过阈值 $minPts_i$, 认为该核心点为核心对象, 核心点及其邻域点构成初始点云簇。对进行过 Eps 邻域判断的点标记为已访问, 依次遍历初始点云簇中未访问的点, 对其进行 Eps 邻域判断, 将满足阈值条件的点的邻域点加入点云簇中, 不断重复该步骤直至点云簇中不再存在未被访问的且满足阈值条件的点。重复上述步骤直至所有核心点均完成了 DBSCAN 聚类, 聚类完成后, 将点云簇中的非动态点标记为动态点并加入动态点集合中。最后, 通过剔除原始全局点云地图中的动态点集合, 得到不含动态物体痕迹的全局静态 3D 点云地图。

2 实验验证与分析

2.1 实验数据介绍

1) 标准数据集

SemanticKITTI^[21] 公开数据集是在原有 KITTI 数据集基础上, 针对 00-10 序列的点云数据进行了逐点语义标注, 提供了 28 种不同的语义标签, 包括道路、车辆、建筑物、行人等, 可用于动态点云滤除算法的定量分析与测试。

2) 校园实测数据

校园实测数据集由实验室的移动机器人采集得到, 通过运行 Fast-lio2^[22] 算法, 在录制点云数据的同时, 同步获取每一帧点云的位姿信息, 实验场景为东南大学群贤路和中心喷泉周围道路。研究使用的移动机器人如图 4 所示, 选用型号为 Ouster-64 的激光雷达采集点云信息, 工作频率为 10 Hz, 实验平台为具有 16 G 运行内存、AMD

Ryzen75800U 处理单元的个人电脑, 操作系统为 Ubuntu18.04, ROS 版本为 melodic。



图 4 移动机器人

Fig. 4 Mobile robot

2.2 基于标准数据集的静态地图生成实验

为了全面评估所设计的动态点云滤除算法在静态地图构建任务中的性能, 开展了系统的对比实验, 选取了当前在离线动态点云去除领域的主流算法 Remover^[11] 和 ERASOR^[15] 作为基准进行比较。这两种算法广泛应用于离线全局静态点云地图生成, 并在多个数据集上展现了较为优秀的性能。通过对比实验, 能够更全面地验证所提算法在复杂动态场景下进行动态点云移除与静态地图构建的效果和优势。实验基于 SemanticKITTI 数据集, 从中筛选了 3 个具有不同典型的典型场景(括号内为对应的区间起止索引), 涵盖多种复杂环境及多种动态对象(如行人、车辆、骑行者等)。算法参数设置见表 1。

表 1 算法参数设置

Table 1 Algorithm parameter settings

变量	数值
扇区划分数 N_{θ} / 个	108.0
段区划分数 N_r / 个	20.0
强动态阈值 $\xi_{threshold}^D$	0.2
径向比率阈值 $\eta_{threshold}$	0.6
基础邻域 Eps_{base} / m	0.6
基础最小点数 $minPts_{base}$ / 个	15.0
密度控制因子 $\mu_{density}$	1.5
核心点最低数目 Num_{cp}^{min} / 个	3.0
VOI 区域 Z 坐标上限 h_z^{max} / m	3.0
VOI 区域 Z 坐标下限 h_z^{min} / m	-1.0

场景 1: 选用 SemanticKITTI 00 (790~830) 序列中的小镇街道场景作为测试对象。该场景具有典型的城市街道特征, 街道两旁是静态的建筑物、停车的车辆和绿植等, 环境布局较为复杂。该场景的动态对象主要为在建筑物旁行走的行人, 且动静态点云存在一定的重叠区域, 进一步加大了动态识别的难度。场景的相机图像如图 5 所示, 原始点云地图及不同算法构建的静态点云地图如图 6 所示。

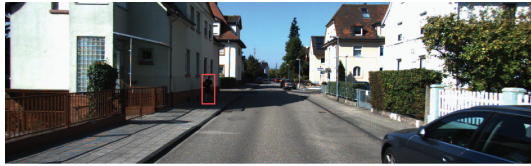


图 5 小镇街道相机图像

Fig. 5 Camera image of a town street

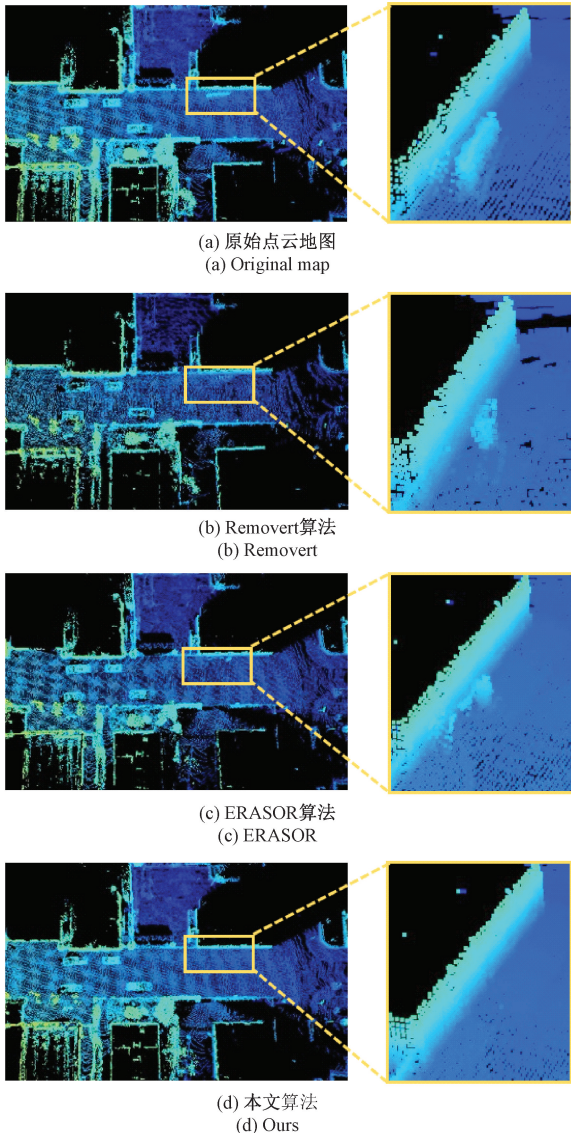


图 6 静态点云地图结果对比

Fig. 6 Comparison of static point cloud map results

场景 2: 选用 SemanticKITTI 01 (120~210) 序列中的高速公路场景作为测试对象, 用于验证算法在非完全空旷条件下处理多个高速运动物体时的动态识别能力。该场景为两条相向而行的空旷公路和一条右转弯匝道, 公路两旁是密集排列的树木, 动态对象主要为多个高速行驶的车辆。图 7 为该场景的相机图像, 不同算法构建的静态点云地图的对比结果如图 8 所示。



图 7 高速公路相机图像

Fig. 7 Camera image of the highway

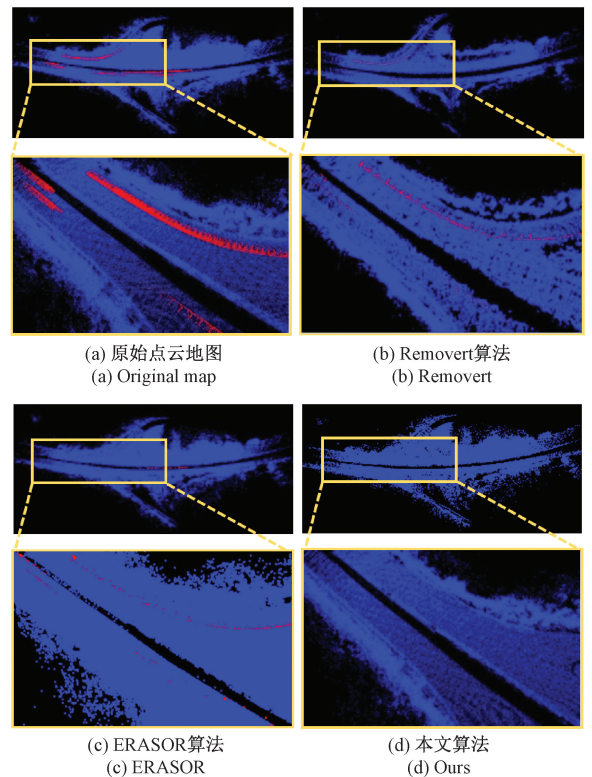


图 8 静态点云地图结果对比

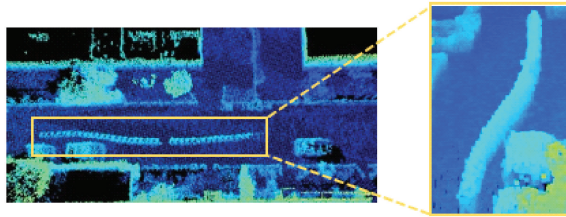
Fig. 8 Comparison of static point cloud map results

场景 3: 选用 SemanticKITTI 05 (1120~1160) 序列中密集住宅区场景作为测试对象, 用于验证算法对于复杂空间布局和弱动态点云识别的精度和鲁棒性。该场景的特点是房屋紧密排列、道路狭窄、公共设施及空间布局复杂。动态对象主要为街道一侧的骑行者, 此类对象通常形状不规则且边界模糊, 在复杂空间布局中极易与静态点云混淆。该场景的相机图像如图 9 所示, 不同算法构建的静态点云地图对比结果如图 10 所示。

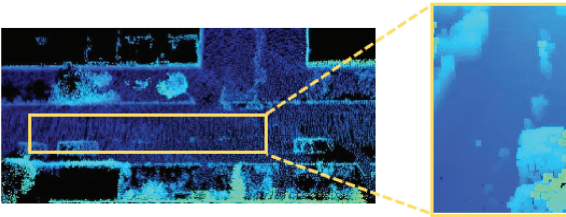


图9 密集住宅区相机图像

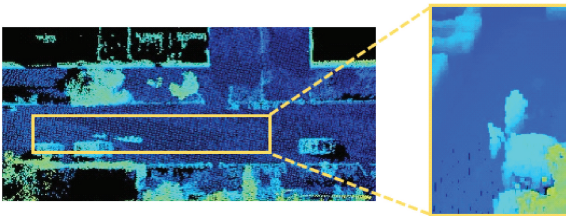
Fig. 9 Dense residential area image



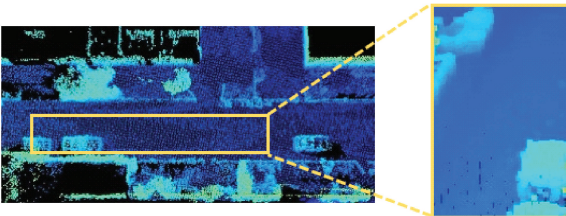
(a) 原始点云地图
(a) Original map



(b) Removert算法
(b) Removert



(c) ERASOR算法
(c) ERASOR



(d) 本文算法
(d) Ours

图10 静态点云地图结果对比

Fig. 10 Comparison of static point cloud map results

实验结果表明,ERASOR 算法在场景 2(高速公路)中表现较好。但在场景 1 和 3 的复杂环境中,由于动态点云存在重叠且动态对象形状不规则,ERASOR 算法难以凭借单一的高度描述子实现对此类弱动态点云的精准识别。Removert 算法采用批量移除动态点和逐步恢复静态点的策略,在动静态点云高度重叠场景中,不仅容易误杀边界或地面处的静态点,且对于弱动态点云的识别

能力有限。当面对高速运动或形状复杂的动态物体时,Removert 仍存在明显的痕迹残留。所提算法凭借多阶段动态滤除策略,精准识别复杂场景中的弱动态点云。阶段 2 有效解决了动静态点云重叠导致的漏检问题,阶段 3 通过处理弱动态点云形状复杂、边界模糊等难点,进一步提升了动态点识别精度。所提算法在滤除动态点云的同时,最大限度地保留了静态点云。在多个复杂场景下的地图构建精度优于现有主流算法,展现出更强的鲁棒性和适用性。此外,2.4 节将对该实验结果进行定量分析。

2.3 基于校园真实场景的静态地图生成实验

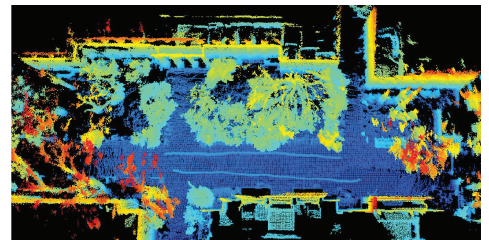
为验证所提算法在复杂场景下的动态点云滤除效果以及静态地图构建精度,选取校园中的两个典型场景作为实验区域,涵盖多样化环境特征,以评估所提算法在不同环境下的地图构建性能。

场景 4:选用东南大学群贤路路段,该路段环境特征复杂,包括两侧的建筑物、树木以及停放的车辆等静态设施。该路段作为校园主要通行区域之一,动态对象主要为经过的行人和车辆,具有流动性强且动静态点云交织重叠的特性,具体场景如图 11 所示。采用 Removert 算法、ERASOR 算法以及所提算法进行实验验证,实验结果如图 12 所示。

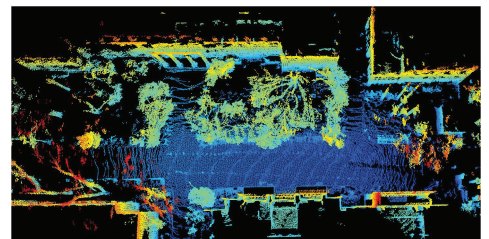


图11 校园真实场景图像

Fig. 11 Campus real scene image



(a) 原始点云地图
(a) Original map



(b) Removert算法
(b) Removert

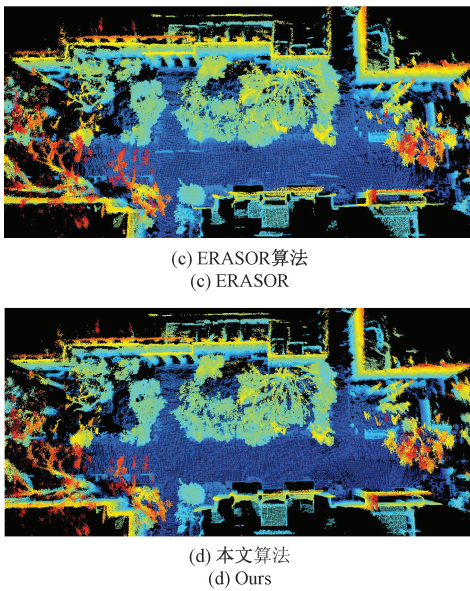


图 12 静态点云地图结果对比
Fig. 12 Comparison of static point cloud map results

场景 5: 选用中心喷泉周围路段, 该区域环境布局复杂, 周边密集分布着树木和建筑物。场景中包含大量行人、骑行者和车辆等动态对象, 动态对象种类丰富, 动态点云形状多样且分布复杂, 对算法的动态点云识别能力和鲁棒性提出了更高的要求。具体场景如图 13 所示, 实验结果如图 14 所示。

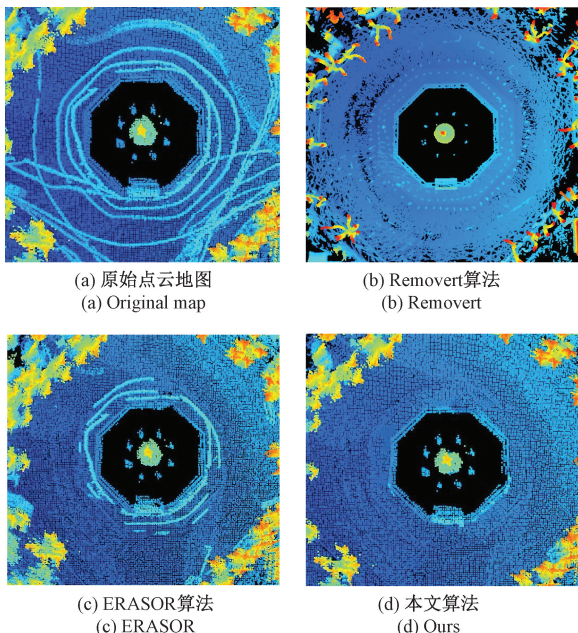


图 13 校园真实场景图像
Fig. 13 Real scene image of the campus



图 14 静态点云地图结果对比
Fig. 14 Comparison of static point cloud map results

Removeert 算法在面对复杂多样的动态对象以及较为密集的空间布局时, 无法完整滤除动态点云, 生成地图中均存在动态痕迹残留。此外, Removeert 算法存在明显的静态点误杀现象, 导致静态点云的完整性下降。此外, 2.4 节将对该实验结果进行定量分析。

2.4 静态地图构建精度对比实验及消融实验

利用 SemanticKITTI 数据集提供的语义标签对生成的静态地图进行定量分析, 数据集中类别 252~259 为动态类别, 此类点为需要移除的动态点。在此, 选用 3 个关键的评价指标来衡量静态地图的构建质量, 指标定义为:

静态点保留率 (preservation rate, PR) 指生成全局地图中被标记为静态的点占初始地图中总静态点的比例, 计算公式如式 (27) 所示。

$$PR = \frac{static_point_num \Big|_{result_map}}{static_point_num \Big|_{original_map}} \quad (27)$$

其中, 分子表示生成全局地图中的静态点数量, 分母表示初始点云地图中的静态点数量。

动态点拒绝率 (rejection rate, RR) 指初始地图中成功被移除的动态点所占比例, 计算公式如式 (28) 所示。

$$RR = 1 - \frac{dynamic_point_num \Big|_{result_map}}{dynamic_point_num \Big|_{original_map}} \quad (28)$$

其中, 分子表示生成全局地图中残留的动态点数量, 分母表示初始点云地图中的动态点数量。

F1 分数指静态点保留率和动态点滤除率的调和平均数, 用于综合评估算法在静态点和动态点的保留与拒绝性能, 计算公式如式 (29) 所示。

$$F1 = \frac{2 \times PR \times RR}{PR + RR} \quad (29)$$

通过上述指标对 ERASOR、Removeert 以及所提算法所构建的地图精度进行评估。首先, 对 2.2 节中选取的 3 个场景进行地图构建精度对比, 结果见表 2。为深入分析所提出算法中各模块对整体性能的贡献, 设计并开展了消融实验。其中, 变体算法 Ours-cluster 表示移除了自适应 DBSCAN 聚类模块后的版本, 用于评估该模块对动态点云滤除精度的影响。

实验结果表明, 在校园实测场景中, ERASOR 算法和

表2 地图构建精度对比

Table 2 Comparison of map construction accuracy

场景	序列	算法	评价指标		
			PR/%	RR/%	F1
场景 1	00	Removert	75.260	88.608	0.814
		ERASOR	92.504	88.853	0.906
		Ours-cluster	97.086	96.507	0.968
		本文算法	96.508	98.048	0.975
场景 2	01	Removert	78.540	80.287	0.794
		ERASOR	88.297	88.137	0.882
		Ours-cluster	94.428	97.160	0.957
		本文算法	94.171	98.471	0.963
场景 3	05	Removert	98.651	82.289	0.897
		ERASOR	97.430	83.100	0.896
		Ours-cluster	99.217	95.235	0.972
		本文算法	98.522	98.524	0.985

由表2可知,所提算法的动态点拒绝率在3个场景中均达到了98%以上,F1分数始终保持在0.96以上,平均PR和平均RR相比于ERASOR算法分别提升3.95%和13.44%,相比于Removert算法分别提升14.56%和17.46%。变体算法的拒绝率相对于完整算法略有下降,尤其在动态对象不规则且环境复杂度较高的场景中差距更为显著。表明自适应DBSCAN聚类模块对于滤除环境中的不规则动态点云发挥了关键作用,弥补了ERASOR算法容易遗漏部分动态点云的不足。

对2.3节中选取校园实测场景进行地图构建精度对比。由于缺乏校园场景的动态点云真值标签,采用基于点云处理的动态点标注方法。首先,通过随机采样一致性算法(random sample consensus, RANSAC)对点云进行地面分割;然后,对非地面点云进行欧式聚类并从中筛选动态聚类,如图15所示。由此生成校园实测场景中的仿真值标签,作为评估实验结果的依据,实验结果见表3。

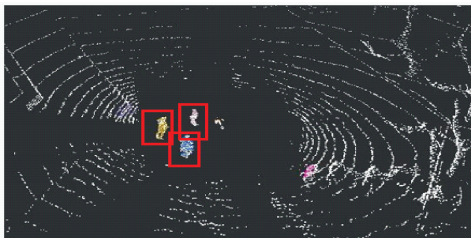


图15 动态聚类点云示意图

Fig. 15 Dynamic clustered point cloud image

由表3可知,在校园实测场景中,所提算法的动态点移除率均达到了97%以上,并且综合性能优于两种基

表3 实测地图构建精度对比

Table 3 Comparison of actual map construction accuracy

场景	算法	评价指标		
		PR/%	RR/%	F1
场景 4	Removert	80.179	86.301	0.831
	ERASOR	94.914	90.163	0.925
	本文算法	94.058	97.908	0.959
场景 5	Removert	77.694	85.469	0.814
	ERASOR	93.720	89.604	0.916
	本文算法	95.237	98.814	0.970

准算法,说明所设计算法在不同类型的校园场景下均能保持较高的动态滤除精度和静态点云完整性。

此外,进一步扩展了实验范围,对SemanticKITTI数据集中00~10共11个序列进行了静态地图构建精度的评估。在每个序列中选取包含动态对象的特定区间开展实验,综合分析各算法在不同环境、动态对象类别及其分布条件下的性能,实验结果见表4,括号内为区间所属的数据集序列。

另外,从SemanticKITTI数据集的00~10序列中截取了23个数据段,对Removert、ERASOR以及所提算法进行了23组对比实验并计算精度指标。如图16、17所示,直观地展示了3种算法的地图构建精度差异。

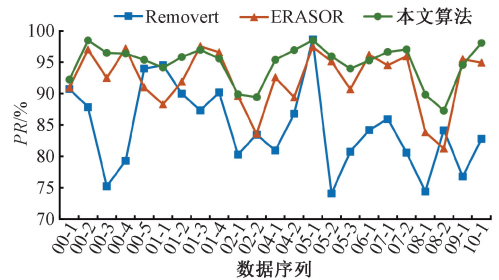


图16 保留率对比折线图

Fig. 16 Line chart of PR comparison

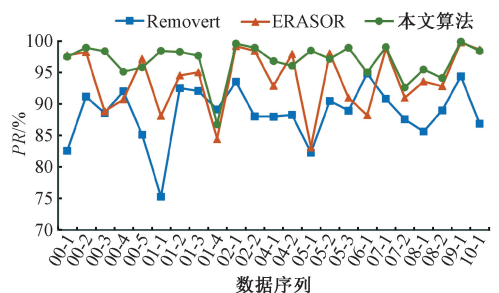


图17 拒绝率对比折线图

Fig. 17 Line chart of RR comparison

表4 地图构建精度对比

Table 4 Comparison of map construction accuracy

区间序列	算法	评价指标		
		PR/%	RR/%	F1
1(00)	Removert	87.865	91.187	0.895
	ERASOR	97.029	98.268	0.976
	本文算法	98.484	98.950	0.987
2(00)	Removert	79.285	92.046	0.852
	ERASOR	97.217	90.735	0.939
	本文算法	96.405	95.133	0.958
3(01)	Removert	87.363	92.108	0.897
	ERASOR	97.551	95.049	0.963
	本文算法	96.983	97.716	0.974
4(02)	Removert	83.474	88.027	0.857
	ERASOR	83.610	98.465	0.904
	本文算法	89.428	98.974	0.940
5(04)	Removert	80.956	88.016	0.843
	ERASOR	92.591	92.891	0.927
	本文算法	95.419	96.827	0.961
6(05)	Removert	80.774	88.971	0.847
	ERASOR	90.680	94.016	0.923
	本文算法	91.009	98.964	0.948
7(08)	Removert	74.372	85.657	0.796
	ERASOR	83.863	93.614	0.885
	本文算法	89.869	93.478	0.916
8(08)	Removert	84.174	89.000	0.865
	ERASOR	81.236	92.852	0.867
	本文算法	87.306	94.173	0.906

所提出的三阶段动态点云滤除策略充分考虑了环境布局的复杂性与动态对象的不规则性,在大部分数据序列中表现出色,静态点保留率始终维持在95%左右且波动较小,相较其他两种算法展现出良好的稳定性与鲁棒性。ERASOR算法采用简单的几何描述子(如高度描述子)来判别动态点云,能够快速高效地处理大规模点云数据,但因其对环境特征的敏感性,容易受到复杂场景或对象的影响,导致处在边界位置的点容易出现残留或误删。而Removert算法在简单场景中效果较好,但该算法依赖位姿估计精度,点云配准中的误差累积会削弱动态点云滤除效果,导致静态地图中仍残留部分动态物体的痕迹。此外,由于误删大量地面点导致在部分序列(00-3、05-2)中表现不佳。相较而言,所提算法在保留率和拒绝率之间实现了较好的平衡,既能够有效保留静态点云的完整

性并准确滤除动态点云,整体性能优于其他两种算法,适合作为复杂动态场景下的高精细地图构建首选方案。

2.5 算法复杂度分析

所设计算法通过3个阶段实现强弱动态点云的识别与完善。假设点云规模为 N ,阶段1的总时间复杂度为 $O(N)$,阶段2的总时间复杂度近似为 $O(N)$,阶段3的总时间复杂度 $O(M\log N)$,因此,所提算法的整体时间复杂度为 $O(M\log N)$ 。Removert算法的时间复杂度为 $O(N)$,ERASOR算法的时间复杂度为 $O(M\log N)$ 。在SemanticKITTI数据集01序列上,3种算法的单次迭代运行时间见表5。

表5 SemanticKITTI数据集01序列的单次迭代运行时间

Table 5 Runtime per iteration on Sequence 01 of the SemanticKITTI dataset

算法	运行时间/s
Removert	0.830 7
ERASOR	0.073 2
本文算法	0.216 8

Removert算法具有最低的时间复杂度,但该算法每个批量中的点云都要进行不同分辨率的重复计算,并为每个点分配静态标签,随着地图规模的增大,Removert算法的耗时也会相应增加。与之相比,所提算法与ERASOR算法在时间复杂度上相当,但由于额外增加了两个识别阶段,导致运行时间略有增加,但仍快于Removert算法。

所设计算法的核心目的是以离线动态滤除的方式,提高复杂动态场景下的全局静态点云地图构建精度。为此,该算法设计了更为精细的点云差异检测方法,通过增加两个额外的识别阶段,适用于处理更复杂的动态场景,如多个不规则动态目标或紧密布局的环境。该方法增强了算法在处理高动态、密集场景时的稳定性和准确性,同时能够构建更高精度的全局静态点云地图。因此,尽管增加了额外的计算步骤,所提算法在高精度地图构建任务中仍表现出了优异的性能。

3 结论

该研究提出了一种有效的高精细静态点云地图生成算法,具有无需依赖先验地图信息或预训练数据的优势。在点云预处理阶段,通过区域化地面分割方法降低地面点对动态识别过程的干扰。在动态点云滤除阶段,逐步引入两种分布式描述符D-POD与D-PODV,并结合SRT与RRT,实现了对强、弱动态点云的准确识别。最后,通过改进的自适应DBSCAN聚类进一步完善了对不规则动

态点云的滤除。实验结果表明,该算法在 SemanticKITTI 数据集和校园实测场景中均表现出良好的性能。与 ERASOR 和 Removert 等主流算法相比,所设计算法在动态对象复杂且环境布局多变的场景中展现出更强的适应性与鲁棒性,为多样化场景下的高精细静态点云地图构建提供了一种可靠且高效的解决方案。

参考文献

- [1] 周治国, 邸顺帆, 冯新. 语义信息增强的 3D 激光 SLAM 技术进展[J]. 仪器仪表学报, 2023, 44(3): 209-220.
ZHOU ZH G, DI SH F, FENG X. Advances in SIE 3D LiDAR SLAM technology [J]. Chinese Journal of Scientific Instrument, 2023, 44(3): 209-220.
- [2] 余联想, 郑明魁, 欧文君, 等. 多传感器融合的移动机器人室外激光 SLAM 算法优化与系统实现[J]. 电子测量与仪器学报, 2023, 37(2): 48-55.
YU L X, ZHENG M K, OU W J, et al. Optimization and system implement of outdoor LiDAR SLAM algorithm for mobile robots based on multi-sensor fusion[J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(2): 48-55.
- [3] 陈熙源, 戈明明, 姚志婷, 等. 雨雪天气下的激光雷达滤波算法研究[J]. 仪器仪表学报, 2023, 44(7): 172-181.
CHEN X Y, GE M M, YAO ZH T, et al. Research on Lidar filtering algorithm for rainy and snowy weather[J]. Chinese Journal of Scientific Instrument, 2023, 44(7): 172-181.
- [4] 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.
- [5] 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 CH, 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.
- [6] 闫河, 王旭, 雷秋霞. 动态场景结合稀疏场景流和加权特征的视觉 SLAM 方法[J]. 中国惯性技术学报, 2024, 32(9): 891-897.
YAN H, WANG X, LEI Q X. Visual SLAM method combining sparse scene flow and weighted features in dynamic environment [J]. Journal of Chinese Inertial Technology, 2024, 32(9): 891-897.
- [7] PARK S, WANG SH Y, LIM H, et al. Curved-voxel clustering for accurate segmentation of 3D LiDAR point clouds with real-time performance [C]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019: 6459-6464.
- [8] YOON D, TANG T, BARFOOT T. Mapless online detection of dynamic objects in 3D LiDAR [C]. 2019 16th Conference on Computer and Robot Vision, 2019: 113-120.
- [9] 陈耀华, 何丽, 王宏伟, 等. 基于视点法剔除动态目标的激光-惯导 SLAM [J]. 仪器仪表学报, 2023, 44(9): 248-256.
CHEN Y H, HE L, WANG H W, et al. LiDAR-inertial SLAM based on visible point method to remove dynamic objects [J]. Chinese Journal of Scientific Instrument, 2023, 44(9): 248-256.
- [10] SCHAUER J, NUCHTER A. The peopleremover-removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid [J]. IEEE Robotics and Automation Letters, 2018, 3(3): 1679-1686.
- [11] 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, 2020: 10758-10765.
- [12] CHEN ZH M, ZHANG K, CHEN H, et al. DORF: A dynamic object removal framework for robust static lidar mapping in urban environments [J]. IEEE Robotics and Automation Letters, 2023, 8(12): 7922-7929.
- [13] 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, 2020: 10765-10771.
- [14] MILIOTO A, VIZZO I, BEHLEY J, et al. Rangenet++: Fast and accurate lidar semantic segmentation [C]. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019: 4213-4220.
- [15] 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.
- [16] ESTER M, KRIEGEL H P, SANDER J, et al. A

- density-based algorithm for discovering clusters in large spatial databases with noise[C]. KDD-96, 1996, 96(34): 226-231.
- [17] 周世波,徐维祥. 密度峰值快速搜索与聚类算法及其在船舶位置数据分析中的应用[J]. 仪器仪表学报, 2018, 39(7):152-163.
ZHOU SH B, XU W X. Clustering by fast search and find of density peaks and its application in ship location data analysis[J]. Chinese Journal of Scientific Instrument, 2018, 39(7): 152-163.
- [18] KIM G, KIM A. Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018: 4802-4809.
- [19] GUO M C, ZHANG L, LIU X, et al. 3D LiDAR SLAM based on ground segmentation and scan context loop detection[C]. 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, 2021: 692-697.
- [20] HIMMELSBACH M, HUNDELSHAUSEN F, WUENSCHEN H J. Fast segmentation of 3D point clouds for ground vehicles[C]. 2010 IEEE Intelligent Vehicles Symposium, 2010: 560-565.
- [21] BEHLEY J, GARBADE M, MILIOTO A, et al. Semantickitti: A dataset for semantic scene understanding of LiDAR sequences[C]. 2019 IEEE/CVF International Conference on Computer Vision, 2019: 9297-9307.

- [22] XU W, CAI Y, HE D, et al. Fast-lio2: Fast direct LiDAR-inertial odometry [J]. IEEE Transactions on Robotics, 2022, 38(4): 2053-2073.

作者简介



范明泽, 2022年于武汉工程大学获得学士学位,现为东南大学硕士研究生,主要研究方向为激光雷达定位与建图。

E-mail: f15830052644@163.com

Fan Mingze received her B. Sc. degree from Wuhan Institute of Technology in 2022.

Now she is a master's candidate at Southeast University. Her main research interests include LiDAR-based localization and mapping.



徐晓苏(通信作者), 1982年于东南大学获得学士学位, 1985年于东南大学获得硕士学位, 1991年于东南大学获得博士学位。现为东南大学仪器科学与工程学院教授, 主要研究方向为惯性导航、组合导航、卫星导航、视觉导航、智能滤波技术。

E-mail: xxs@seu.edu.cn

Xu Xiaosu(Corresponding author) received his B. Sc. degree from Southeast University in 1982, received his M. Sc. degree from Southeast University in 1985, received his Ph. D. degree from Southeast University in 1991. Now he is a professor at the School of Instrument Science and Engineering, Southeast University. His main research areas include inertial navigation, integrated navigation, satellite navigation, visual navigation, and intelligent filtering technologies.