

doi:10.19306/j.cnki.2095-8110.2022.03.009

# 动态环境下融合激光雷达和 IMU 的激光里程计设计

张涛<sup>1,2</sup>, 张晨<sup>1,2</sup>, 魏宏宇<sup>1,2</sup>

(1. 东南大学仪器科学与工程学院, 南京 210096;

2. 东南大学微惯性仪表与先进导航技术教育部重点实验室, 南京 210096)

**摘要:** 激光 SLAM 通常通过多帧点云配准, 完成位姿变换矩阵的估计, 而点云中的动态点会降低激光里程计的定位精度。为了减少动态点引入的误差, 提出了一种动态点云识别算法, 并结合该算法改进了传统特征匹配策略, 组成了动态环境下融合激光雷达和 IMU 的激光里程计。通过约束范围角与动态中心点, 将点云快速分割成多个簇类, 再借助 IMU 信息, 快速建立点云簇类配准关系, 从而去除动态点, 最后根据簇类的对应关系进行约束, 以提高匹配的精度与速度。使用 KITTI 数据集和 UGV 在多个情形下进行了实验。实验结果表明, 该算法可以成功识别点云中的多个动态对象, 并通过去除动态点, 有效地减少了位姿估计的累积误差。

**关键词:** 动态识别; 激光里程计; 点云分割

中图分类号: V556.6

文献标志码: A

文章编号: 2095-8110(2022)03-0070-09

## Design of Lidar Odometry Integrating Lidar and IMU in Dynamic Environment

ZHANG Tao<sup>1,2</sup>, ZHANG Chen<sup>1,2</sup>, WEI Hong-yu<sup>1,2</sup>

(1. School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China;

2. Key Lab of Micro Inertial Instruments and Advanced Navigation Technology of Ministry of Education, Southeast University, Nanjing 210096, China)

**Abstract:** Multi-frame point cloud registration is usually used to estimate the pose transformation matrix. The dynamic points in the point cloud reduce the positioning accuracy of the lidar odometry. In order to reduce the error introduced by dynamic points, a dynamic point cloud recognition algorithm is proposed, and combined with this algorithm, the traditional feature matching strategy is improved to form a laser odometry that integrates lidar and IMU in a dynamic environment. The point cloud is quickly divided into multiple clusters by restricting the range angle and the dynamic center point, and then using the IMU information, the registration relationship of the point cloud clusters is quickly established. Thereby the dynamic points are removed, and finally the precision and speed of matching are improved by constraining according to the corresponding relationship of the clusters. Experiments are carried out in multiple situations by using the KITTI data set and UGV. The results show that the design effectively reduce the cumulative error of pose estimation by removing dynamic points.

**Key words:** Dynamic recognition; Lidar odometry; Point cloud segmentation

收稿日期: 2021-11-02; 修订日期: 2021-12-22

基金项目: 中央高校基本科研业务费(2242021K1G008); 东南大学国家自然科学基金剩余资金培育项目(9S20172204)

作者简介: 张涛(1980-), 男, 博士, 教授, 主要从事激光 SLAM、组合导航等方面的研究。

## 0 引言

近年来,世界各地的各个行业都在推动机器人的发展。智能移动机器人逐渐走入大众视野,成为人类日常生产、生活中不可或缺的组成部分<sup>[1]</sup>。在疫情期间,服务机器人用于辅助诊断、快速检测、精准测温、目标识别<sup>[2-4]</sup>,极大地提高了诊疗的智能化程度,减少医务人员感染风险,提高管控效率。机器人需要克服许多技术难点才能成功完成上述复杂的操作。其中,同时定位和地图绘制(Simultaneous Localization and Mapping, SLAM)技术是机器人学中最活跃和最具挑战性的研究课题之一<sup>[5]</sup>。激光雷达 SLAM 前端根据前后两帧的点云进行配准,在估计变换矩阵时,动态物体会引入一定的误差,最终导致累积误差不断增大。尽管通过引入回环检测节点,可以消除部分估计的累积误差<sup>[6]</sup>。但是,当没有回环检测和高精度地图时,累积误差依旧无法消除。因此,需要通过去除动态物体对应的点云,使得激光里程计估计变换矩阵时引入的误差尽可能小,以实现更准确的定位和建图。

D. Giveki 等<sup>[7]</sup>提出了将颜色和纹理特征同 Choquet 模糊积分相结合的方法应用于背景建模,通过该建模背景对现有对象进行分割,以完成动态对象检测。Zou C. 等<sup>[8]</sup>提出了 3D 全连接条件随机场方法,该网络对激光雷达采集到的连续两个关键帧的点云进行处理,得到每个点作为动态点的概率,从而判断动态点。Yao W. 等<sup>[9]</sup>提出了一种基于物体特征的目标 3D 分割方法,完成了对移动车辆速度估计的改进,该方法主要应用于一些机载激光雷达在空中获取的少量点云。李波等<sup>[10]</sup>提出了一个全卷积神经网络,以激光雷达点云数据为输入样本,结合数据增强训练,得到一个可以检测车辆的神经网络,在神经网络学习的过程中需要手动标记学习样本。李响<sup>[11]</sup>采用傅里叶单像素欠采样成像对数据进行压缩,提取图像特征,并在未获取完整图像信息的情况下,采用深度学习卷积网络对动态物体进行检测和识别。文献<sup>[12]</sup>的解决方案是先使用前馈神经网络定位目标,计算其边界位置,通过融合方法提取特征,最后用 3D 卷积网络判断目标,并且能够得到其实际移动速度。

由于实际场景中存在各种形状和大小的物体,因此本文采用常规的聚类算法得到的效果不佳,且由于点云数量过于庞大,存在复杂度高、聚类速度慢等特点。为了识别激光雷达点云中的动静态目标并

提取动态点,本文提出了一种融合激光雷达和惯性测量单元(Inertial Measurement Unit, IMU)的动态点云识别算法。该策略通过约束激光雷达扫描范围角与动态中心,将点云快速分割成多个簇类,再借助 IMU 预估计的位姿变换信息代替特征点匹配,快速建立点云簇类配准关系,从而根据中心点偏移的程度判断出簇类是否为动态物体,完成动态点云的识别与分割任务。最后,再根据簇类的对应关系进行约束,以提高传统特征匹配策略的精度与速度。

## 1 整体框架

本文的整体方案如图 1 所示,系统通过搭载的激光雷达扫描得到点云与 IMU 测量得到的信息作为输入,最终输出系统的位姿信息。整个系统主要包含 3 个模块:点云预处理模块、特征点提取模块和激光里程计模块。点云预处理模块首先将地面点去除,再通过本文提出的动态点识别去除动态物体与离群散点。特征点提取模块先将由运动造成的畸变通过 IMU 信息进行畸变补偿,再结合簇类信息与点云的粗糙度,提取各自簇类的边缘特征和平面特征。激光里程计模块通过簇类信息与特征信息进行点云配准,从而完成位姿估计的过程,最终输出位姿。

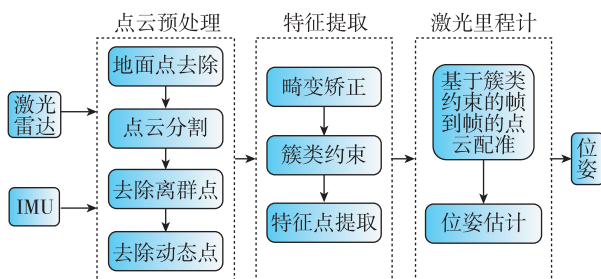


图 1 方案框架

Fig. 1 Scheme framework

规定本文中载体坐标系(图 2)的中心位于激光雷达的坐标系原点,在激光雷达的中心,  $XOZ$  平面平行于激光雷达的顶平面,  $X$  轴指向无人地面车(Unmanned Ground Vehicle, UGV)的右侧,  $Z$  轴指向 UGV 的前方,  $Y$  轴垂直于  $XOZ$  平面向下。

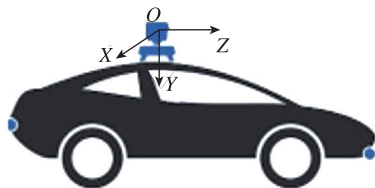


图 2 载体坐标系

Fig. 2 Carrier coordinate system

## 2 算法流程

### 2.1 点云预处理

设  $P_t = \{p_1 \ p_2 \ \dots \ p_n\}$  为  $t$  时刻激光雷达采集得到包含  $n$  个点的点云集合, 其中每个点在笛卡尔坐标系表示为  $p_k = \{x_k \ y_k \ z_k\}$ 。

由于地面点与所有语义对象相连接, 因此, 首先对点云进行预处理, 将地面点进行识别与去除, 本文选择采用文献[13]中的方法去除地面点。由于实际场景中存在各种形状和大小的物体, 且扫描得到的点云数量庞大, 本文根据这些特点提出了一种融合激光雷达和 IMU 的动态点云识别算法。本文提出的语义分割与点云算法的工作流程描述如下:

第一步: 在点云中任意取一点  $p_m$ , 在该点对应的激光雷达扫描角周围的一定范围内 (搜索约束角), 搜索与其欧氏距离相近的点。即满足

$$\|p_m - p_i\| \leq \epsilon \quad (1)$$

其中,  $\epsilon$  表示两点之间的欧拉距离阈值。通过不断循环第一步, 直到范围内所有的点都遍历完毕后, 将这些满足条件的点定为一个簇类。

第二步: 在上述簇类中, 选取上下左右前后 6 个方向上距离  $p_m$  最远的点, 不断进行第一步操作, 且已遍历过的点不再进行二次遍历, 将符合条件的也加入上述簇类,

第三步: 不断重复第一步, 直到没有点再满足式(1), 则停止迭代。如果最终聚类后得到的簇类中的数字大于给定的数量阈值, 则通过式(2)计算出中心点, 将该中心点标记为该簇类的核心点, 并为该簇分配一个新的标签; 如果小于数量阈值, 则将该簇类内的所有点统一标记为离群点。

$$p_{\text{mid}} = \frac{1}{n} \sum_{i=m}^{m+n} p_i \quad (2)$$

通过不断执行上述三步, 最终将点云  $P_t$  分割聚类成  $N$  个簇类  $P_t = \{P_{t,-1} \ P_{t,1} \ \dots \ P_{t,N}\}$ , 其中第二个下标为  $-1$  时代表为离群点。图 3 给出了聚类前后点云俯视图的对比图, 其中每种颜色代表一类聚类点。

对于每个  $P_{t,k}$  都有一个中心点坐标  $P_{\text{mid},t,k}$  与之对应, 因此可以得到每个有效簇类对应的中心点集合为  $P_{\text{mid},t} = \{P_{\text{mid},t,1} \ P_{\text{mid},t,2} \ \dots \ P_{\text{mid},t,N}\}$ 。同理, 可以将最终  $t+1$  时刻采集得到的点云  $P_{t+1}$  分割聚类成  $N$  个簇类  $P_{t+1} = \{P_{t+1,-1} \ P_{t+1,1} \ \dots \ P_{t+1,N}\}$ , 其有效簇类对应的中心点集合为  $P_{\text{mid},t+1} =$

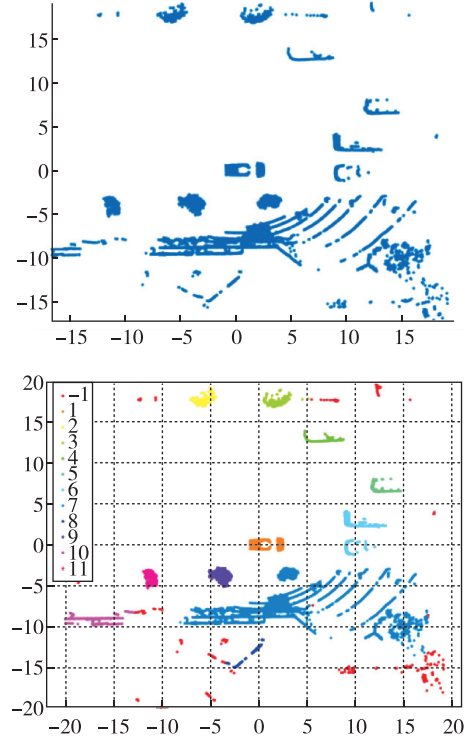


图 3 聚类结果

Fig. 3 Clustering results

$\{P_{\text{mid},t+1,1} \ P_{\text{mid},t+1,2} \ \dots \ P_{\text{mid},t+1,N}\}$ 。

通过 IMU 可以测量计算得到从  $t$  时刻到  $t+1$  时刻的横滚角  $\alpha$ 、俯仰角  $\beta$ 、航向角  $\theta$  与位移量  $st = [st_x \ st_y \ st_z]$ , 可以得到从  $t+1$  时刻载体坐标系转换到  $t$  时刻载体坐标系的姿态变换矩阵为

$$R'_{t+1} = \begin{bmatrix} ca \ c\beta & ca \ s\beta s\theta - sa \ c\theta & ca \ s\beta c\theta + sa \ s\theta \\ sa \ c\beta & sa \ s\beta s\theta + ca \ c\theta & sa \ s\beta c\theta - ca \ s\theta \\ -s\beta & c\beta s\theta & c\beta c\theta \end{bmatrix} \quad (3)$$

其中,  $c$  表示  $\cos$  函数;  $s$  表示  $\sin$  函数。

设  $\tilde{P}_{\text{mid},t}$  为点云  $P_{t+1}$  转换到  $t$  时刻的点云集合, 即

$$\tilde{P}_{\text{mid},t} = R'_{t+1} P_{\text{mid},t+1} + st'_{t+1} \quad (4)$$

借助最近邻迭代算法, 可以得到  $\tilde{P}_{\text{mid},t}$  和  $P_{\text{mid},t}$  的一一对应关系, 计算出对应中心点之间的欧拉距离, 当该距离大于给定阈值时, 则认为是动态物体。

### 2.2 特征提取

激光雷达扫描过程中, 由于载体的运动致使点云变形, 即运动畸变。本文通过线性插值法, 计算每个点对应的相对运动变换矩阵, 去除运动畸变。设  $t_{\text{start}}$  为当前雷达扫描的起始时刻,  $t_{\text{end}}$  为当前雷达扫描的结束时刻,  $t_{\text{now}}$  为当前点对应的扫描时刻。假设  $T_{\text{end}}^{\text{start}}$  表示从  $t_{\text{end}}$  到  $t_{\text{start}}$  的相对运动变换矩阵, 则将当前时刻转换到初始时刻的相对运动变换矩

阵可以表示为

$$\mathbf{T}_{\text{now}}^{\text{start}} = \frac{t_{\text{now}} - t_{\text{start}}}{t_{\text{scan}}} \mathbf{T}_{\text{end}}^{\text{start}} \quad (5)$$

其中,  $t_{\text{scan}}$  为激光雷达扫描一周的时长。

为了均匀且快速地提取点云中的特征点, 此处对每个簇类分别计算各自的粗糙度  $c$ , 从而提取出边缘特征和平面特征, 计算公式如下

$$c = \frac{1}{\|\mathbf{P}_{t,i}\| \cdot \|\mathbf{p}_{k,i}\|} \left\| \sum_{l \in \mathbf{P}_{t,i}, l \neq k} \mathbf{p}_{k,i} - \mathbf{p}_{l,i} \right\| \quad (6)$$

其中,  $\mathbf{P}_{t,i}$  表示  $t$  时刻, 标签为  $i$  的点云簇;  $\mathbf{p}_{k,i}$  为  $\mathbf{P}_{t,i}$  中的一个点  $k$ ;  $\mathbf{p}_{l,i}$  为在  $\mathbf{P}_{t,i}$  中与  $\mathbf{p}_{k,i}$  不同的其他点。

根据  $c$  值对每个簇类中的点进行排序, 选择较大的几个点为边缘特征点, 较小的几个点为平面点。最终得到  $t$  时刻的边缘特征集合为  $\boldsymbol{\varepsilon}_k$ , 平面特征集合为  $\mathbf{H}_k$ 。

### 2.3 激光里程计

激光里程计的目的在于通过最小化点到边缘特征与点到平面特征之间的距离, 估计 2 个连续点云之间的变换关系。为了找到各自的对应关系, 本文中使用的激光里程计的基本算法与文献[14]中的一致, 但在搜索过程中, 加入了簇类约束, 因此在进行特征搜索匹配时, 仅对同一簇类进行点云配准, 通过簇类约束条件排除错误的对应关系。根据 2.2 节提到的算法, 可以得到连续 2 个时刻的边缘特征与平面特征, 分别为  $\boldsymbol{\varepsilon}_k, \boldsymbol{\varepsilon}_{k+1}$  与  $\mathbf{H}_k, \mathbf{H}_{k+1}$ 。引入簇类约束后, 点到线的距离公式变为

$$d_\varepsilon = \frac{|(\mathbf{p}_{k+1,i,\text{label}} - \mathbf{p}_{k,j,\text{label}}) \times (\mathbf{p}_{k+1,i,\text{label}} - \mathbf{p}_{k,l,\text{label}})|}{\|\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,l,\text{label}}\|} \quad (7)$$

其中, label 为对应簇类的标签;  $i \in \boldsymbol{\varepsilon}_{k+1}; j, l \in \boldsymbol{\varepsilon}_k$ 。

同样, 点到面的距离公式变为

$$d_H = \frac{\left| (\mathbf{p}_{k+1,i,\text{label}} - \mathbf{p}_{k,j,\text{label}}) \cdot \frac{(\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,l,\text{label}}) \times (\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,m,\text{label}})}{\|(\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,l,\text{label}}) \times (\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,m,\text{label}})\|} \right|}{\|(\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,l,\text{label}}) \times (\mathbf{p}_{k,j,\text{label}} - \mathbf{p}_{k,m,\text{label}})\|} \quad (8)$$

其中,  $i \in \mathbf{H}_{k+1}; j, m \in \mathbf{H}_k$ 。

通过结合 IMU 的预估计姿态变换矩阵式(3)与式(7)~式(8), 可以得到一个非线性优化函数

$$f(\mathbf{T}_k^{k+1}) = \begin{pmatrix} d_\varepsilon \\ d_H \end{pmatrix} = d \quad (9)$$

激光里程计最后的目标是将式(9)通过迭代使得  $d$  尽可能地趋向于 0。

$$\mathbf{T}_k^{k+1} \leftarrow \mathbf{T}_k^{k+1} - (\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J}))^{-1} \mathbf{J}^T d \quad (10)$$

其中,  $\mathbf{J} = \partial f / \partial \mathbf{T}_k^{k+1}$ ;  $\lambda$  为非线性优化算法 LM 的一个因子。

### 3 实验验证

本实验使用的 UGV 如图 4 所示, 采用激光雷达采集点云, 同时为了便于验证, 使用相机采集对应时间的图像信息。激光雷达选择使用 Velodyne-VLP16, 该设备以 10 帧/s 的速度旋转, 每个周期捕获大约 30k 点。相机选用 ZED2, 以 10 帧/s 和 720p 的分辨率获取图像, 同时该相机内部自带 IMU, 可输出 400Hz 的 IMU 信息。处理器为英伟达的 Jetson AGX Xavier, 最高能提供 32TOPs 的运算能力。在本节中, 首先使用配备激光雷达的 UGV 在多种场景下采集数据, 并使用本文提出的算法验证其可行性, 最后给出一个直观的分割结果。然后, 将该算法应用于 UGV 和 KITTI 数据集<sup>[15]</sup>, 并与其他算法进行对比, 以验证去除点云中动态物体对应的相关点后对激光雷达 SLAM 定位精度和效率的影响。整个系统在 Ubuntu 18.04 (Linux) 中使用 ROS 执行, 并借助 ROS 中 Time\_Synchronizer 模块对 3 个传感器输入的信息进行时间戳对齐。实验中涉及的参数如表 1 所示。

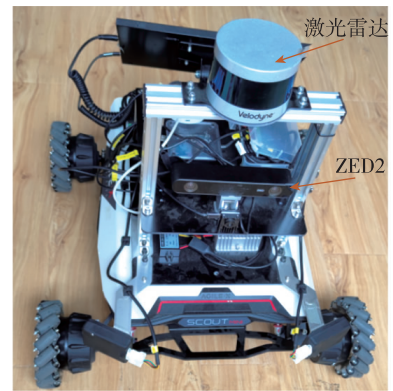


图 4 无人地面车

Fig. 4 Unmanned ground vehicle

表 1 实验参数

Tab. 1 Experiment parameters

参量	值
距离阈值/m	0.02
搜索约束角/(°)	5
种群最小点数	30

### 3.1 动态点检测试验

实验地点选在走廊。实验分为以下四种情况。

情况一：当激光雷达静止时，检测到单个动态物体并移除；

情况二：当激光雷达静止时，检测到多个动态物体并移除；

情况三：当激光雷达处于动态状态时，检测并去除走廊拐角处的动态物体；

情况四：当激光雷达处于动态状态时，检测直线走廊并去除动态物体。

如图 5 所示，从四种不同的情况中选择间隔为 1s 的两帧进行处理。在实验环境中，移动的物体主要由人来扮演。通过使用第 2 节提出的算法进行动态点去除，四种不同情况的结果如图 6 所示。为了观察方便，图 6 显示了处理后的点云的俯视图，其中蓝色六边形代表激光雷达所在位置，聚类得到的动态物体用青色虚线标记。



(b) 情况二



(c) 情况三



(d) 情况四

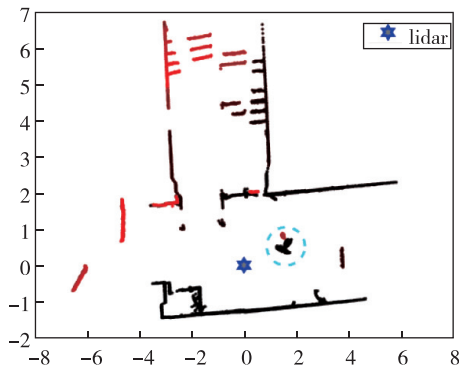
图 5 四种情况对应的视觉图像

Fig. 5 Visual images corresponding to the four situations

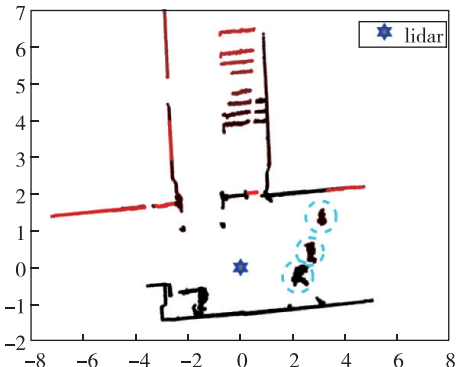
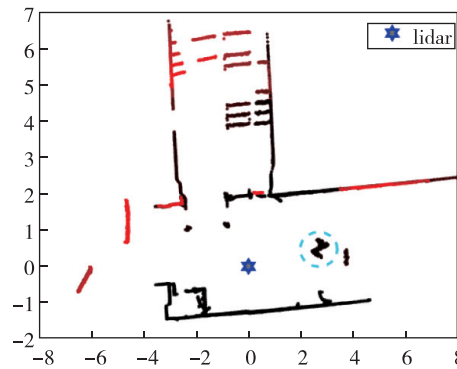


(a) 情况一

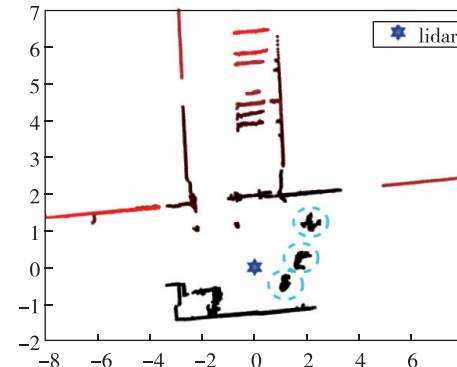
最后，可以在去除图 6 中的动态点后得到点云的顶视图。从图 7 中可以看出，在上述四种情况下，本文提出的算法成功去除了环境中的动态物体。在这四种情况中，还可以观察到部分情况下由于人体离激光雷达太近，以至于挡住了身后的整面墙。

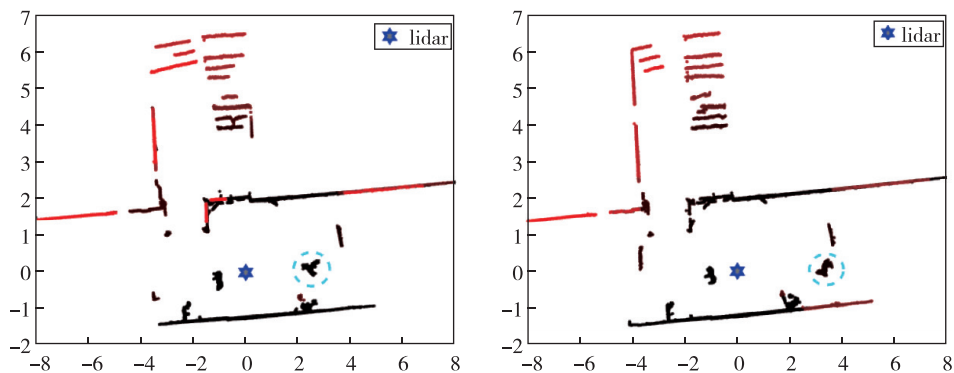


(a) 情况一

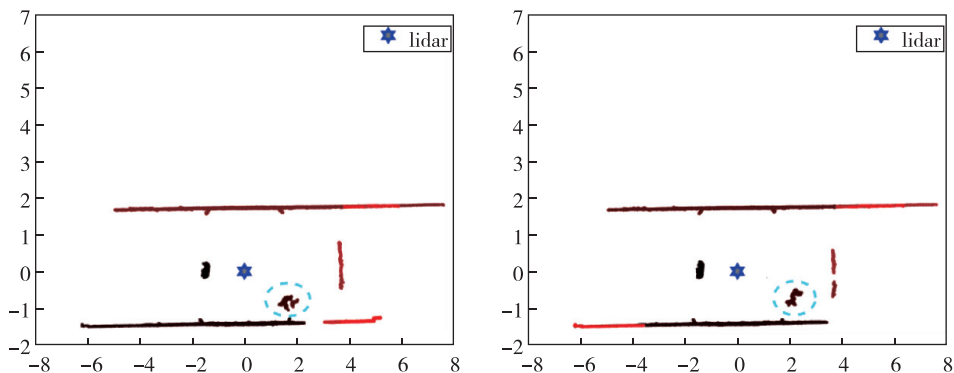


(b) 情况二





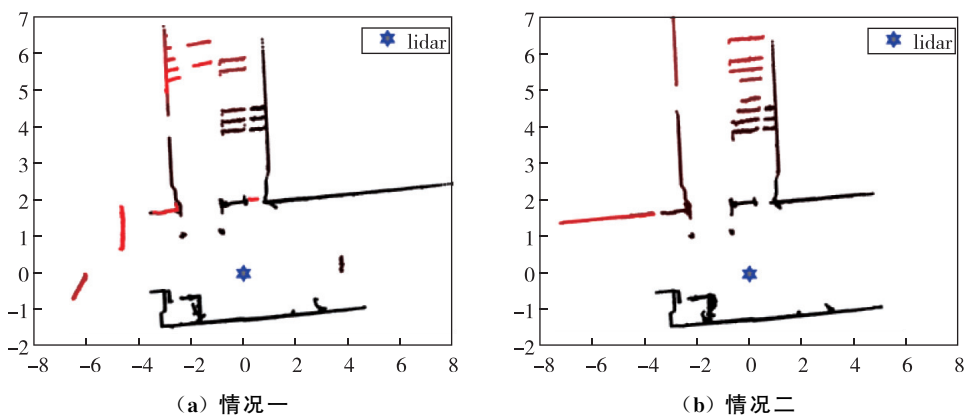
(c) 情况三



(d) 情况四

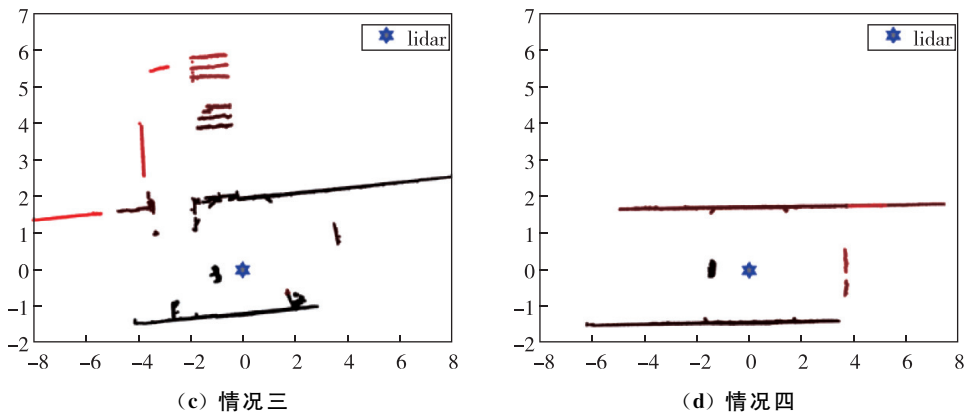
图 6 对应的俯视图

Fig. 6 The top views corresponding to the four situations in Fig. 5



(a) 情况一

(b) 情况二



(c) 情况三

(d) 情况四

图 7 去除动态物体后点云俯视图

Fig. 7 The top views of point cloud after removing dynamic objects

因此,部分墙体在俯视图中产生了断层。由于部分墙只在两个时刻中的一个时刻被检测到,在该算法的处理中,墙的这部分同样被当作动态点被去除。在后续的点云配准过程中,如果一帧中有一些点在另一帧中没有被检测到,那么在配准过程中强行寻找不对应的点进行配准会引入相当大的误差。因此,将这部分静态点识别为动态点并去除它们,也有助于提高姿态估计的准确性。表2显示了处理前后的点云数量,以及在CPU i7-4720上运行的平均处理时间。

表2 分割实验仿真结果

情况	初始点数	处理后点数	平均处理时间/s
1	20883	20053	0.0107
2	20775	19192	0.0105
3	21052	19738	0.0116
4	22214	21594	0.0119

### 3.2 定位实验

这部分主要使用装载了激光雷达的UGV和KITTI数据集开展实验。通过对比实验结果,可以验证本文提出的动态点去除策略对定位精度的影响。选择开源解决方案LEGO\_LOAM与本文的方案进行实验比较。

这部分实验分为以下四种情况:

情况一:UGV静止;

情况二:UGV运动,路径中不存在环路;

情况三:UGV运动,路径中存在环路;

情况四:将两种解决方案应用于KITTI数据集。

情况一:由于UGV处于静止状态,实际坐标始终为 $[0, 0, 0]$ 。但是,动态物体的存在也会影响UGV本身的位姿估计。实验结果如图8所示,其中蓝线表示使用LEGO\_LOAM估计得到的实时位置,绿线表示使用本文方案估计得到的实时位置。从图8可以看出,即使UGV处于静止状态,由于周围物体的移动,在估计位置时也会存在偏差。通过去除动态对象对应的点云,可以减少动态对象引入的误差。具体误差值如表3所示。由表3可知,由动态误差引入的误差的标准偏差降低了约57.7%,最大偏差降低了约49.2%。

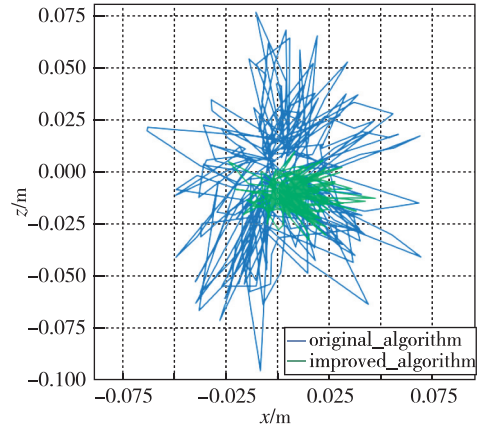


图8 情况一的定位结果

Fig. 8 The positioning result of situation 1

表3 情况一的定位结果偏差

	LEGO_LOAM	本文方案
标准差/m	0.0182	0.0077
最大偏差/m	0.0962	0.0489
最小偏差/m	0.0007	0.0004
平均偏差/m	0.0097	0.0059
平均处理时间/s	0.0257	0.0179

情况二:该情况下,由于实时真值难以获取,因此本文仅比较终点的估计位置和实际位置,以观察最后的定位误差。分别在LEGO\_LOAM和本文方案下运动UGV,获得估计位置的路径如图9所示。UGV实际行驶了18.035m,且路径呈直线。通过算法估计得到,LEGO\_LOAM的终点位置偏差为0.5494m,本文方案的终点位置偏差为0.4237m,误差降低了约23%。

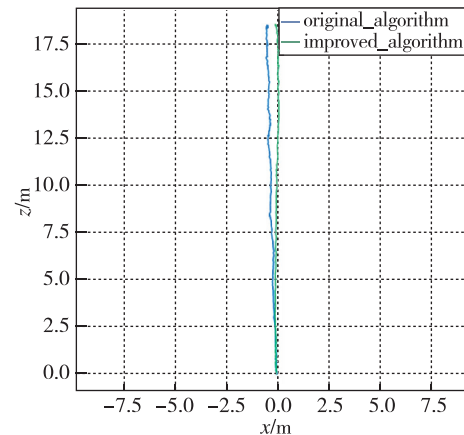


图9 情况二的路径

Fig. 9 The path of situation 2

情况三:这种情况下,UGV 从一个点出发,再次回到起点,从而形成一个闭环。LEGO\_LOAM 的最终偏差值为 0.1736m,本文方案的最终偏差值为 0.1040m。如图 10 所示,2 个最终位置偏差都不大,这是由于 LEGO-LOAM 中使用了检测节点,因此可以有效地消除环路中的累积误差,即使行驶的路径更远,也可以得到比情况二更准确的定位结果。

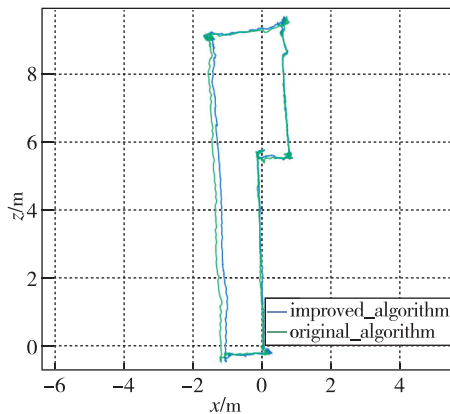


图 10 情况三的路径

Fig. 10 The path of situation 3

情况四:为了验证大范围复杂场景下全路径定位精度的提升,将算法应用到 KITTI 数据集下。最终两种解决方案的结果如图 11 和表 4 所示。在整

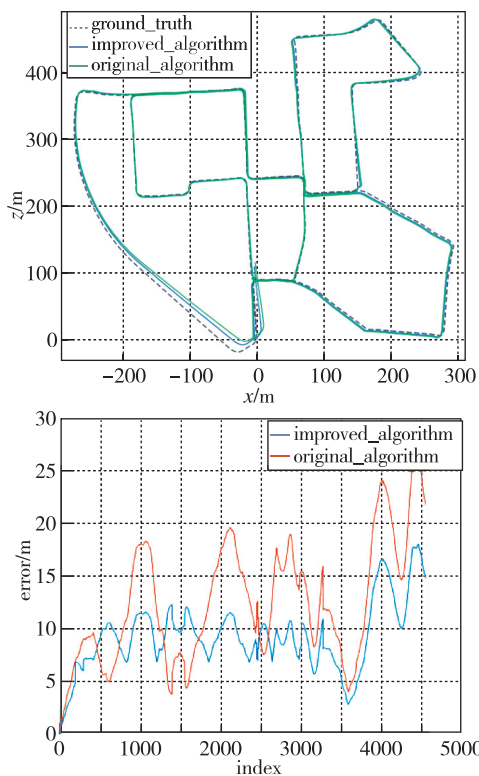


图 11 情况四的路径及位置误差

Fig. 11 The path and positioning error of situation 4

表 4 情况四的定位结果偏差

Tab. 4 Positioning results deviation of situation 4

	LEGO_LOAM	本文方案
标准差/m	5.3717	4.4878
最大偏差/m	26.4013	18.0254
最小偏差/m	0.1964	0.1537
平均偏差/m	12.2687	8.0059
平均处理时间/s	0.0256	0.0187

个路径中,最大偏差减少了约 31.7%,标准偏差减少了约 16.5%。

### 4 结论

本文针对动态物体造成激光里程计定位产生偏差的问题,提出了一种融合激光雷达和 IMU 的动态点云识别算法,并且结合该过程改进了传统特征匹配策略,组成了动态环境下融合激光雷达和 IMU 的激光里程计。算法分析与实验结果表明:

1)动态物体的存在对系统自身位姿估计确实产生了一定的影响,而本文提出的动态点云识别算法,通过约束范围角与动态中心点,将点云快速分割成多个簇类,再借助 IMU 预估计的变换矩阵,快速建立点云簇类配准关系,从而去除动态点,能够有效地去除环境中的单个和多个动态物体。

2)当动态物体和静态物体在三维空间中重合时,很容易将两者聚类为一个整体,在后续过程中则会将它们作为一个完整的动态物体进行点云去除。然而,在实际情况中,通常只有在动态对象与静态对象碰撞时才会发生,因此可以单独处理。

3)当存在回环的情况下,能够通过增加回环检测节点减小动态物体引入的累积误差,但无法彻底消除,而通过去除动态点可以较大地提高激光里程计定位的准确度。

4)当连续两帧中检测到的物体在另外一帧中没有出现时,本文提出的方案会将其认为是动态物体进行去除,能够提高点云配准的精度。

### 参考文献

[1] Biswal P, Mohanty P K. Development of quadruped walking robots: a review[J]. Ain Shams Engineering Journal, 2021, 12(1): 2017-2031.

[2] 李振威,程咏梅,刘刚. INS/ADS/GPS 组合导航高灵敏度故障检测和识别方法[J]. 中国惯性技术学报, 2020, 28(6): 694-700.



- Li Zhenwei, Cheng Yongmei, Liu Gang. High-sensitivity fault detection and identification method for INS/ADS/GPS integrated navigation[J]. *Journal of Chinese Inertial Technology*, 2020, 28(6): 694-700 (in Chinese).
- [3] Ha Q P, Yen L, Balaguer C. Robotic autonomous systems for earthmoving in military applications[J]. *Automation in Construction*, 2019, 107(13): 102934.
- [4] Xu Y, Wang Z. Visual sensing technologies in robotic welding: recent research developments and future interests[J]. *Sensors and Actuators A Physical: NATO*, 2021, 320(1): 112551.
- [5] Wu Z, Yue Y, Wen M, et al. Infrastructure-free hierarchical mobile robot global localization in repetitive environments[J]. *IEEE Transactions on Instrumentation and Measurement*, 2021(9): 1.
- [6] Deng C, Luo X, Zhong Y. Improved closed-loop detection and Octomap algorithm based on RGB-D SLAM [C]// *Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. Dalian, China, 2020: 73-76.
- [7] Giveki D. Robust moving object detection based on fusing Atanassov's intuitionistic 3D fuzzy histogram roughness index and texture features[J]. *International Journal of Approximate Reasoning*, 2021, 135(18): 1-20.
- [8] Zou C, He B, Zhu M, et al. Encode-decode network with fully connected CRF for dynamic objects detection and static maps reconstruction[J]. *Signal Processing Image Communication*, 2021, 95(10): 116237.
- [9] Yao W, Hinz S, Stilla U. Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018, 66(3): 260-271.
- [10] Li B, Zhang T, Xia T. Vehicle detection from 3D lidar using fully convolutional network[J/OL]. *arXiv preprint, arXiv: 1608.07916*, 2016.
- [11] 李响. 基于单像素探测和深度学习的运动物体识别技术[D]. 广州: 暨南大学, 2020.  
Li Xiang. Moving object recognition based on single-pixel detection and deep learning[D]. Guangzhou: Jinan University, 2020(in Chinese).
- [12] Xian X H. Moving object detection based on deep learning[J]. *Electronic Technology and Software Engineering*, 2020(11): 146-149.
- [13] Himmelsbach M, Hundelshausen F V, Wuensche H J. Fast segmentation of 3D point clouds for ground vehicles [C]// *Proceedings of the IEEE Intelligent Vehicles Symposium*. IEEE, 2010: 560-565.
- [14] Zhang J, Singh S. Low-drift and real-time lidar odometry and mapping [J]. *Autonomous Robots*, 2017, 41(2): 401-416.
- [15] 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.

(编辑:李瑾)