

DOI: 10.19650/j.cnki.cjsi.J2108287

面向变化场景的 LiDAR 鲁棒定位与地图维护方法^{*}

吕 品,季博文,赖际舟,方 玮,郑国庆
(南京航空航天大学自动化学院 南京 211106)

摘 要:无人车在工业场景自主运行时,通常基于先验地图匹配对自身进行定位,然而场景变化会影响其定位精度。针对于此,本文提出了一种面向变化场景的激光雷达鲁棒定位与地图维护架构,其包括地图匹配、定位优化、地图维护模块。在该架构中,提出了一种基于变化检测的匹配算法,降低了变化场景引起的匹配误差;设计了基于激光雷达里程计与先验地图匹配的因子图融合模式,提高了定位解算鲁棒性;提出了一种基于最近点搜索的误检测点滤波方法,提高了变化点检测准确性。最后,通过仿真与试验搭建了变化场景验证环境,对基于 Loam 的匹配算法与本文算法进行了对比验证。结果表明,本文算法可以有效抑制场景变化引起的匹配误差,在实际场景中定位均方根误差优于 3 cm,定位精度相较于传统算法提高 67.4%。

关键词:鲁棒定位;激光雷达;变化检测;地图维护

中图分类号: TH86 TP242.6 **文献标识码:** A **国家标准学科分类代码:** 510.80

LiDAR robust positioning and map maintenance method for changing scenes

Lyu Pin, Ji Bowen, Lai Jizhou, Fang Wei, Zheng Guoqing

(College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: Unmanned vehicles usually locate themselves based on prior map matching when operating autonomously in industrial scenarios. However, the scene change will affect the positioning accuracy of unmanned vehicles. In view of this, we propose a LiDAR robust positioning and map maintenance architecture for changing scenarios, which includes map matching, positioning optimization, and map maintenance modules. A matching algorithm based on change detection is proposed, which reduces the matching error caused by changing scenarios. A factor graph fusion mode based on LiDAR odometer and prior map matching is designed to improve the robustness of the positioning solution. A filtering method of false detection points based on nearest point search is proposed, which improves the accuracy of change point detection. Finally, we establish a changing scene verification environment through simulation and experimentation and compare the performance of matching based on Loam and the proposed algorithm. The results show that the algorithm can effectively suppress the matching error caused by the scene change, the root mean square error of positioning is better than 3 cm in the actual scenario, and the positioning accuracy is improved by 67.4% compared with the traditional algorithm.

Keywords: robust positioning; LiDAR; change detection; map maintenance

0 引 言

近年来,随着自动驾驶相关技术的发展,无人车在巡检、安防、运输等多个工业场景得到了广泛应用^[1]。在无人车自动运行过程中,自身位姿的精准估计是保障其顺利完成的关键技术之一。

目前,惯性/卫星/里程计融合是无人车常用的导航

方案^[2-3],然而卫星信号容易受到干扰或遮挡影响,在室内无法使用,在密集建筑物周边定位精度也会下降。在一些固定场景内(如工厂、码头),通过预先铺设二维码^[4]、磁条^[5]等方式,可以为无人车提供高精度定位参考。然而,无人车仅可沿标志物行驶,限制了其作业灵活性。视觉传感器凭借体积小、成本低等优点,在无人车领域中得到了广泛应用,如通过视觉同时定位与构图^[6-8](simultaneous localization and mapping, SLAM)进行车辆

收稿日期:2021-07-23 Received Date: 2021-07-23

^{*} 基金项目:国家自然科学基金(61973160)、航空科学基金(2018ZC52037)、工信部民机专项(2018-S-36)资助

位姿估计,然而其定位性能会受到光线干扰、环境特征的影响。

激光雷达具有全天候、高精度的优点,结合 SLAM 技术可以实时提供位置、姿态信息,成为自动驾驶的常用传感器之一。激光雷达分为二维激光雷达和三维激光雷达。二维激光雷达采用单线扫描形式,成本相对较低,但仅能感知平面点云信息,通常用于结构化室内环境。常用的二维激光雷达 SLAM 算法包括 Hector SLAM^[9]、Gmapping^[10]、Cartography^[11]等。三维激光雷达采用多线扫描形式,可测量三维环境点云,在室内、室外场景均得到了广泛应用。Zhang 等^[12]将环境中的角点特征和平面点特征用于匹配,并基于此设计了经典的 Loam SLAM 方法。在此基础上,Shan 等^[13]添加了地面约束以抑制高度的发散,增加了回环检测功能,设计了 Lego-Loam SLAM 方法。

为获得高精度的定位信息,通常对无人车运行场景进行预先建图,继而通过实时匹配的方式进行位置求解。激光雷达地图可以分为路标地图与点云地图两种。Kim 等^[14]从激光雷达扫描的反射率值中提取车道标志特征与路标地图进行匹配。Wang 等^[15]从多帧点云中提取路缘特征与高精地图进行匹配,实现车辆的高精定位。但是基于路标地图的定位方法在路标稀疏或受遮挡的情况下易受干扰。相比之下,基于点云地图的定位方法具有更强的鲁棒性。Levinson 等^[16]利用点云地图实现了城市环境下的厘米级定位,Ding 等^[17]通过视觉传感器实现了激光雷达点云地图下的高精度匹配定位。

然而,当无人车在长期作业过程中,受季节、气候等自然因素,以及建筑拆迁、物品位置变化等人为因素影响,其地图精度会有所下降,从而影响定位性能。因此,需要通过变化检测来对地图进行维护。Pagad 等^[18]将神经网络训练与点云地图变化检测相结合,对构图时的动态物体进行剔除,从而保障了长期构图的稳定性。Kim 等^[19]基于距离图像的差异性对构图过程中的动态物体进行剔除,并基于该方法对点云地图进行更新。Jinno 等^[20]提出了一种点云地图更新架构,通过栅格模型对变化点云进行检测与处理。Egger 等^[21]通过关键帧点云的改变比例进行场景变化检测,继而通过替换关键帧实现地图更新。然而,上述方法仅将变化检测用于地图维护,未将其与激光雷达地图匹配过程相结合,无法提高当前时刻定位精度,从而也影响了地图更新精度。

针对于此,本文提出了一种基于先验地图的激光雷达变化场景鲁棒定位方法,其主要创新点如下:

1) 建立了一种基于先验地图的激光雷达鲁棒定位与地图维护架构,相对传统激光雷达地图匹配算法,增加了

定位优化、变化误检测滤波模块,以保障无人车在变化场景中的长期可靠运行;

2) 提出了一种基于变化检测的激光雷达点云地图匹配算法,通过检测与剔除相较先验地图的变化点云信息,实现了激光雷达与先验地图的鲁棒匹配,提高了变化场景中的定位精度;

3) 设计了基于激光雷达里程计与先验地图匹配的因子图融合模式,实现了对当前位置的最优融合估计,从而为地图维护提供高精度位置基准;

4) 提出了一种基于最近点搜索的误检测点滤波方法,在检测得到变化点云后进行误检测判断,从而提高变化点云检测准确性。

1 激光雷达鲁棒定位与地图维护算法构架

在先验地图激光雷达匹配算法中,由于预先已对运行场景构图,因此区别于传统激光雷达 SLAM 构架,其核心步骤在于点云匹配以及地图维护^[21-22]。传统算法构架如图 1 所示。首先将激光雷达扫描的点云信息与地图进行帧图匹配,得到位姿信息;继而对当前帧激光雷达扫描的点云与先验地图进行对比,得到变化点云信息;然后将变化点云在地图中进行更新,对场景点云地图进行维护。然而,上述算法在帧图匹配过程中,将激光雷达扫描的点云信息直接与先验地图进行匹配,而场景变化会影响匹配精度,并进一步影响了地图维护效果。

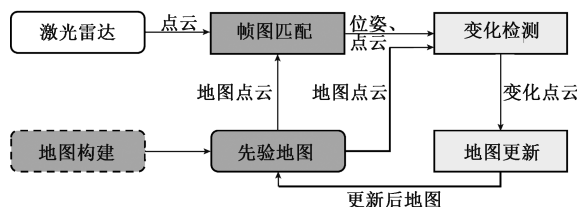


图1 传统激光雷达先验地图匹配算法框架

Fig. 1 Framework of the traditional LiDAR priori map matching algorithm

针对上述问题,本文提出了一种鲁棒激光雷达先验地图匹配构架,如图 2 所示。首先,改进帧图匹配过程,在匹配前对激光雷达扫描的当前帧进行新增点检测与剔除,降低了场景变化对匹配精度的影响;其次,新增定位优化模块,通过帧间匹配对载体位姿进行估计,并基于因子图对帧间匹配与帧图匹配结果进行融合,进一步提升了位姿估计精度;另外,在地图变化检测后,新增误检测滤波模块,通过临近点搜索对检测质量进行判断,提高了变化检测准确度。

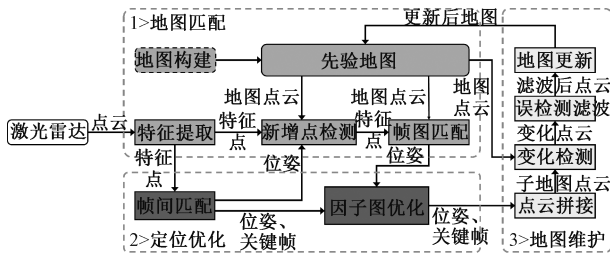


图2 改进的激光雷达先验地图匹配算法框架

Fig.2 Framework of the modified LiDAR priori map matching algorithm

matching algorithm

2 激光雷达鲁棒定位与地图维护方法

2.1 地图匹配

基于预先构建的场景地图,通过地图匹配可以获取载体在地图坐标系(记为 n 系)下的绝对位姿。传统算法将点云与地图直接匹配,在场景相对地图发生变化时匹配精度会受到影响。在本文提出的鲁棒地图匹配算法中,首先将每一帧点云的特征点集与先验地图在八叉树结构下进行对比,进行新增点检测与剔除,继而利用处理后的点云与先验地图进行匹配,从而提高点云的匹配精度,其整体流程如图3所示。

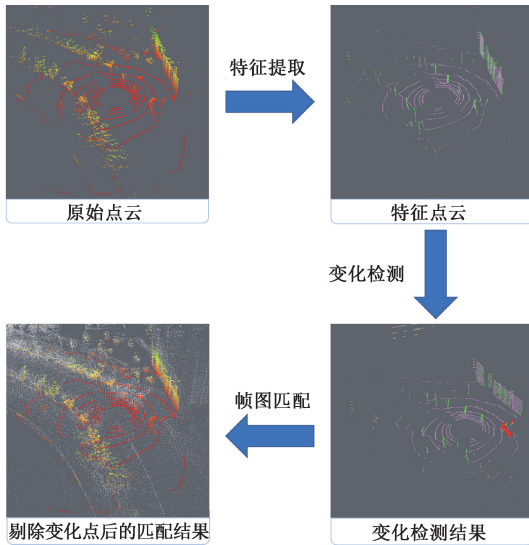


图3 基于变化检测的先验地图匹配流程

Fig.3 Priors map matching diagram base on change detection

1) 特征提取

在进行新增点检测前,需对每一帧点云进行特征提取,本文选择角点、平面点特征^[12]。首先,将当前帧点云 P_{l_t} 投影至距离图像 $D_{b \times c}$ 中, b 代表激光雷达的线数, c 代表激光雷达的水平角分辨率, $D_{b \times c}$ 中每个像素对应 P_{l_t} 中的每个点 p_i ,像素值记为 r_i ,其为 p_i 到激光雷达中心的欧

氏距离。对于图像中的每一列进行评估^[22],从而区分出地面点云 P_g 和非地面点 P_g 。随后,对距离图像中的非地面点进行聚类分割^[23],剔除点云数量较少的类别后,得到分割后点云 L_t 。针对 P_g 和 L_t 进行特征提取,根据该点距离图像中同一行的连续点集合 Q ,计算其粗糙度 c :

$$c = \frac{1}{|Q| \times \sum_{i \in Q} \|r_i\|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \quad (1)$$

其中, r_i 代表该点的深度, r_j 代表集合 Q 中点的深度。将 c 值较大的非地面点作为角点 $P_{l_t}^e$, c 值较小的非地面点作为平面点 $P_{l_t}^p$;从 $P_{l_t}^e$ 选取 n_e 个 c 值最大的点作为角点特征 $F_{l_t}^e$, n_p 个 c 值最小的点和地面点 P_g 作为平面点特征 $F_{l_t}^p$,从而得到特征点集 $F_{l_t}^l = \{F_{l_t}^e, F_{l_t}^p\}$ 。

2) 新增点检测

将先验地图点云表示为 P_M , t 时刻激光雷达点云表示为 P_{l_t} 。对 P_{l_t} 中新增点的检测,实质是求解存在于 P_{l_t} 但不存在于 P_M 中的点云集合 P_{l_t} ,可表示为:

$$P_{l_t} = \{p_i | p_i \in P_{l_t}, p_i \notin P_M\} \quad (2)$$

为求解该问题,本文将当前帧点云 P_{l_t} 与地图点云 P_M 在八叉树结构下进行对比。八叉树是一种有效的数据存储结构^[24],其内存占用小,处理效率高。八叉树的每个节点代表一个体素,每个节点中存储着其被占用的概率。通过对八叉树的节点进行遍历,可以快速地检测出八叉树的变化,从而实现地图变化检测。

在得到当前帧特征点集 $F_{l_t}^l$ 后,将其由激光坐标系(记为 l 系)变换至 n 系:

$$\begin{bmatrix} F_{l_t}^n \\ 1 \end{bmatrix} = \hat{T}_{l_t}^n \begin{bmatrix} F_{l_t}^l \\ 1 \end{bmatrix} \quad (3)$$

其中, $\hat{T}_{l_t}^n$ 表示 t 时刻激光雷达的位姿矩阵,其通过激光雷达帧间匹配推算得到:

$$\hat{T}_{l_t}^n = T_{l_{t-1}}^n \Delta T_{l_t}^{l_{t-1}} \quad (4)$$

其中, $T_{l_{t-1}}^n$ 为上一时刻激光雷达位姿, $\Delta T_{l_t}^{l_{t-1}}$ 为 $t-1$ 时刻到 t 时刻激光雷达帧间匹配得到的相对位姿转换矩阵。

在得到 $F_{l_t}^n$ 后,将地图点云 P_M 投影至八叉树下得到 O_M ,并将 $F_{l_t}^n$ 投影至 O_M 中得到 O_{l_t} 。通过对比 O_M 与 O_{l_t} ,可以得到存在于 O_{l_t} 但不存在于 O_M 的新增的体素集合 $\{I_i\}_{i=1,2,\dots,n}$,而位于体素集合中的点即为新增点 P_{l_t} ,从 $F_{l_t}^n$ 中剔除新增点 P_{l_t} 可得到用于匹配的点云 $\tilde{F}_{l_t}^n$ 。

3) 帧图匹配

通常先验地图以关键帧的形式进行存储,其包含了各关键帧的角点特征 F_i^e 和平面点特征 F_i^p 。为了提高地图匹配效率,以激光雷达帧间匹配得到的位姿 $\hat{T}_{l_t}^n$ 为初值,提取 $\hat{T}_{l_t}^n$ 附近的 n 帧关键帧拼接为一个子地图 S 。包

含了角点子地图和平面点子地图:

$$S = \{S_i^e, S_i^p\} \quad (5)$$

其中, $S_i^e = \bar{F}_i^e \cup \bar{F}_{i+1}^e \cup \dots \cup \bar{F}_{i+n-1}^e$

$$S_i^p = \bar{F}_i^p \cup \bar{F}_{i+1}^p \cup \dots \cup \bar{F}_{i+n-1}^p$$

继而,将剔除新增点后的点云 $\bar{F}_{L_i}^n$ 与子地图 S 进行匹配,可以得到载体位姿。对于 $\bar{F}_{L_i}^n$ 中角点 $\bar{F}_i^e$ 和平面点 $\bar{F}_i^p$, 分别在 S_i^e 和 S_i^p 寻找其对应点,搜寻方法可参考 Loam^[12]。

点到线的距离和点到面的距离的计算方法如式(6)、(7)所示。

$$d_e = \frac{|(p_{f,i}^e - p_{s,j}^e) \times (p_{f,i}^e - p_{s,l}^e)|}{|p_{s,j}^e - p_{s,l}^e|} \quad (6)$$

$$d_p = \frac{\left| \begin{array}{c} (p_{f,i}^p - p_{s,j}^p) \\ (p_{s,j}^p - p_{s,l}^p) \times (p_{s,j}^p - p_{s,m}^p) \end{array} \right|}{|(p_{s,j}^p - p_{s,l}^p) \times (p_{s,j}^p - p_{s,m}^p)|} \quad (7)$$

其中, $p_{f,i}^e$ 代表 $\bar{F}_i^e$ 中的角点, $p_{f,i}^p$ 代表 $\bar{F}_i^p$ 中的平面点; $p_{s,j}^e, p_{s,l}^e$ 代表 $p_{f,i}^e$ 对应的 S_i^e 中的角点, $p_{s,j}^p, p_{s,l}^p, p_{s,m}^p$ 代表 $p_{f,i}^p$ 对应的 S_i^p 中的平面点。因此,可以构建最小二乘问题式(8)对 t 时刻地图匹配位姿 $T_{L_i}^n$ 进行求解:

$$\min_{T_{L_i}^n} \left\{ \sum_{p_{f,i}^e \in \bar{F}_i^e} d_{e_i} + \sum_{p_{f,i}^p \in \bar{F}_i^p} d_{p_i} \right\} \quad (8)$$

其可通过高斯-牛顿法进行求解。

2.2 定位优化

通过将激光雷达当前帧点云与上一帧点云进行匹配,可以得到载体两帧之间的相对位姿变换关系。对于面向先验地图的激光雷达匹配算法,由于场景点云地图事先已知,因此在传统算法中通常直接通过帧图匹配得到载体位姿,而不进行帧间匹配。在本文算法框架中引入帧间匹配,以实现变化场景下定位精度的进一步提高。

虽然激光雷达帧间匹配得到的位姿精度不受场景变化影响,然而其定位误差会随时间累积。因此,本文将帧图匹配与帧间匹配相融合,对帧间匹配误差进行抑制。目前融合方法可以分为基于滤波的融合方法^[25-28]和基于因子图的融合方法^[29-32]。由于因子图对历史时刻优化效果更好,本文采用因子图对帧图匹配、帧间匹配解算的位姿进行融合,将帧间匹配因子作为相对约束,帧图匹配因子作为绝对约束,其融合构架如图4所示。

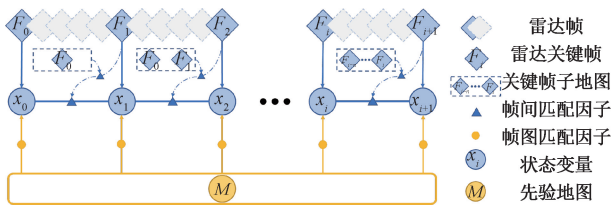


图4 基于帧间匹配与帧图匹配融合的因子图框架

Fig. 4 Factor graph fusion framework base on scan-to-scan and scan-to-map

将待优化变量定义为:

$$x = [R, a] \quad (9)$$

其中, R 代表 l 系到 n 系的旋转矩阵, a 代表 l 系到 n 系的平移矩阵。

为了提高优化效率,只有当载体位姿变换超过一定程度时,才向因子图中添加节点。在添加新的节点后,采用增量平滑和贝叶斯树映射进行优化^[33]。本文中选取的关键帧距离为 0.5 m,角度为 5°。

2.3 地图维护

在系统长期运行过程中,地图的即时维护是保障其长期运行精度的关键。在本文设计的地图维护算法,首先对周围场景进行实时变化检测,然后通过临近点搜索对误检测点进行判断与剔除,继而更新地图,实现对地图的维护。变化检测主要包含新增点检测和消失点检测,新增点检测在前文进行了描述,本节将着重介绍消失点检测和误检测点滤波方法。

1) 消失点检测

在新增点检测中,直接将激光雷达扫描点云与先验地图投影到八叉树地图中进行对比。然而对于消失点检测,当由于遮挡等因素导致激光雷达扫描区域发生变化时,上述策略容易发生误检测。因此,首先要对激光雷达扫描区域进行提取,并在该区域内进行消失点检测。光线追踪法是一种求解光线经过路径的方法,其通过计算点云到激光雷达的路线,可以对激光雷达的扫描区域进行判断。假设 $P_{L_i}^n$ 中某一个测量点为 $p_{L_i}^n$,同时当前时刻激光雷达在 n 系下的位置为 $a_{L_i}^n$,根据光线的起点 $a_{L_i}^n$ 和终点 $p_{L_i}^n$ 可以确定该光线穿过的体素集合 v_i ,如图5所示。通过对当前帧中所有点云进行光线追踪,可以得到先验地图 O_M 中所有被点云穿过的体素集合 $\{V_i\}_{i=1,2,\dots,n}$ 。

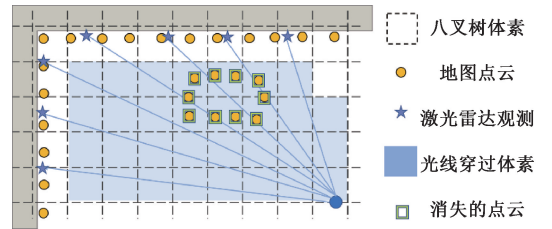


图5 消失点检测示意图

Fig. 5 Schematic diagram of disappeared points detection

消失点可定义为同时存在于 $\{V_i\}_{i=1,2,\dots,n}$ 与先验地图点云 P_M 中的点云集合:

$$P_{D_i} = \{p_d \mid p_d \in P_M, p_d \in \{V_i\}_{i=1,2,\dots,n}\} \quad (10)$$

在消失点检测过程中,地面点由于激光雷达测量噪声或传感器定位误差等原因,会存在部分地面体素被光线穿过而造成误检测的情况。因此在获得激光雷达点云后,需要先进行地面分割并剔除地面点,不进行地面点的消失检测。

2) 误检测点滤波

受传感器定位误差影响,部分位于同一体素的点云会被划分到不同体素。如图6所示,右斜线标注的体素会被判断为新增体素,则体素内的点会被判断为新增点,这会导致误检测点的产生。因此,在变化检测后增加误检测点滤波模块,对变化点云进行临近点搜索,并通过与临近点的距离进行误检测判断。

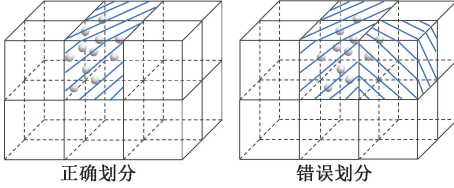


图6 点云正确/错误体素划分示意图

Fig. 6 Schematic diagram of right/wrong voxel division of point cloud

误检测点滤波主要分为两个步骤:

(1) 邻近点过滤

新增点与消失点的邻近点过滤有所不同:对于新增点,对新增点集合 P_{I_i} 中每一个点 p_i 在地图点云 P_M 搜索最近点 p_m ;对于消失点,对消失点集合 P_{D_i} 中每一个点 p_d 在当前帧点云 P_{I_i} 搜索最近点 p_n 。若 p_i 或 p_d 与最近点距离小于一定阈值,则该点为误检测点,并将其在新增点集合 P_{I_i} 或消失点集合 P_{D_i} 中删除,得到过滤后的新增点集合 P'_{I_i} 或消失点集合 P'_{D_i} :

$$\begin{cases} E_{I_i} = \{p_a | p_i \in P_{I_i}, \|p_i - p_m\| < d_1\} \\ E_{D_i} = \{p_b | p_d \in P_{D_i}, \|p_d - p_n\| < d_2\} \\ P'_{I_i} = \{p'_i | p'_i \in P_{I_i}, p'_i \notin E_{I_i}\} \\ P'_{D_i} = \{p'_d | p'_d \in P_{D_i}, p'_d \notin E_{D_i}\} \\ d_1 = \|p_i\| \times r/\lambda, d_2 = \|p_d\| \times r/\lambda \end{cases} \quad (11)$$

其中, d_1, d_2 代表设定的阈值, λ 代表激光雷达测距范围, r 代表八叉树的分辨率, E_{I_i} 为误检测的新增点集合, E_{D_i} 为误检测的消失点集合。

(2) 离群点过滤

在邻近点过滤基础上,进一步采用半径滤波对检测结果进行过滤。对于 P'_{I_i} 中的每一个点,构建一个以 r 为中心半径的球体。如果球体中其它点的数量低于给定的阈值,则该点将被半径过滤器删除。通过该方法,可以对检测结果中的离群点进行剔除。

基于本文提出的误检测滤波算法,可以有效地提高变化检测的精度,误检测滤波效果如图7所示。

3) 地图更新为了提高运行效率,本文变化检测结果以关键帧形式进行存储。关键帧根据距离进行选取,变化关键帧集合定义为 $F_i = \{T_{I_i}, I_i, D_i\}_{i=1,2,\dots,n}$, T_{I_i} 表示该

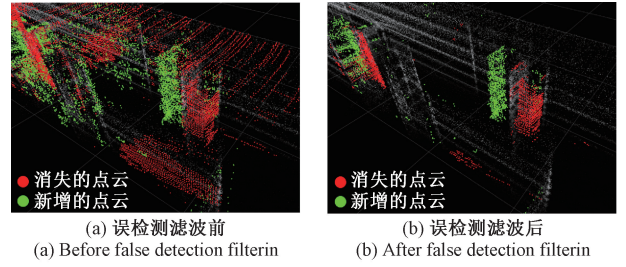


图7 误检测滤波效果图

Fig. 7 Effect of false detection filtering

关键帧点云位姿, I_i 表示该关键帧点云对应的新增点集合, D_i 该关键帧点云对应的消失点集合。因此,在进行地图更新时,需要搜索变化关键帧附近的地图关键帧对地图进行更新,其方法如下。

(1) 新增点云更新:根据变化点关键帧 F_i 中每一帧点云位姿 T_{I_i} ,从地图关键帧集合 $\{F_M\}$ 中搜寻距离最近的两帧关键帧点云 F_j, F_k ,然后将新增点集合 I_i 添加到其中,得到更新后的关键帧点云 F'_j, F'_k :

$$\begin{aligned} F'_j &= F_j \cup I_i \\ F'_k &= F_k \cup I_i \end{aligned} \quad (12)$$

(2) 消失点云更新:根据变化关键帧 F_i 中每一帧点云位姿 T_{I_i} ,从地图关键帧集合 $\{F_M\}$ 中搜索激光雷达扫描半径内的关键帧集合 $\{F_m\}$,对 $\{F_m\}$ 内所有关键帧删除消失点 D_i ,得到更新后的关键帧集合 $\{F'_m\}$:

$$\{F'_m\} = \{F'_k | F'_k = \{p'_k | p_k \in F_k, p_k \notin D_i\}, F_k \in \{F_m\}\} \quad (13)$$

3 验证与分析

为了对本文算法进行验证,分别通过仿真与试验对变化场景进行模拟,并构建相应验证平台。在仿真验证中,基于虚幻4(unreal engine 4, UE4)搭建码头场景,并通过移动集装箱位置模拟场景变化。试验验证在走廊环境开展,通过移动纸箱模拟场景变化。Loam算法是目前激光雷达地图匹配的常用方法,因此将本文提出的方法与其进行对比,以体现在变化场景中定位精度的提升。本文相关算法在机器人操作系统(robot operating system, ROS)中基于C++实现,在i7-9750 CPU处理器中运行。

3.1 仿真验证与分析

本文通过UE4与Airsim^[34],搭建了基于真实物理仿真引擎的算法验证环境。UE4是一款用于构建三维场景的物理引擎,可实现对三维世界的真实模拟。Airsim是一款自动驾驶仿真软件,可实现对无人车、各类传感器的仿真模拟。仿真平台构建如图8所示,通过UE4对码头

变化场景进行仿真,并通过 Airsim 模拟无人车运行、激光雷达测量。

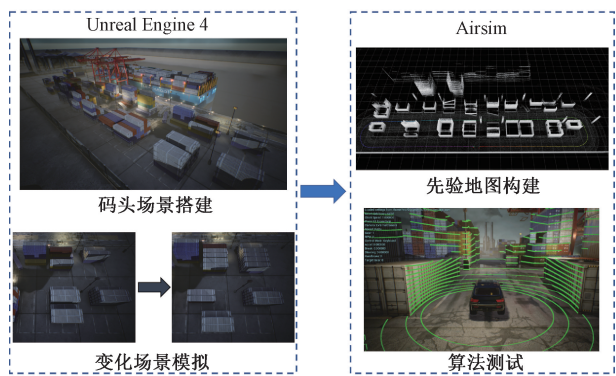


图 8 仿真平台框架

Fig. 8 Simulation platform framework

在仿真中,激光雷达线数为 16,垂直扫描角度为 $-22.5^{\circ}\sim 22.5^{\circ}$,水平扫描角度为 $0^{\circ}\sim 360^{\circ}$,输出频率为 10 Hz。首先通过激光雷达构建码头先验点云地图,然后对集装箱进行移动和删减以模拟场景改变,并再次采集激光雷达数据,进而对比本文提出算法与 Loam 在变化场景中的定位精度。场景变化检测结果如图 9 所示,定位结果对比如图 10 所示,定位误差对比如图 11 和表 1 所示。

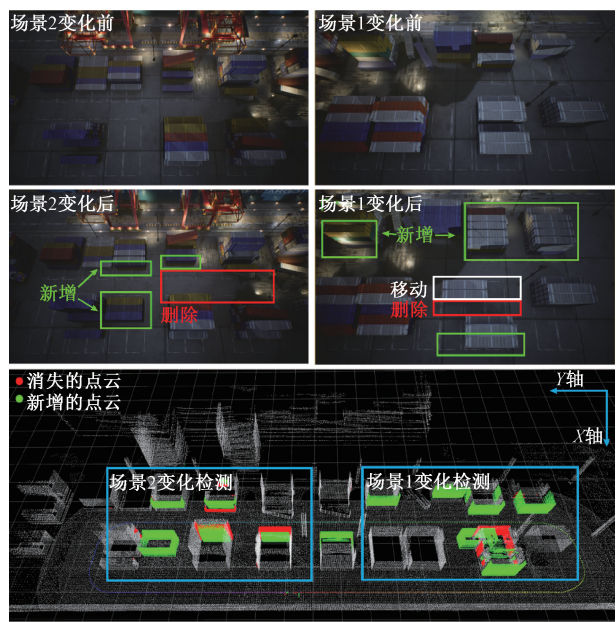
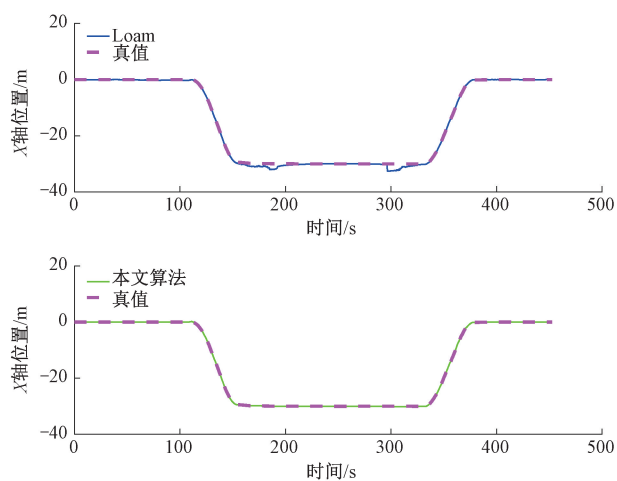


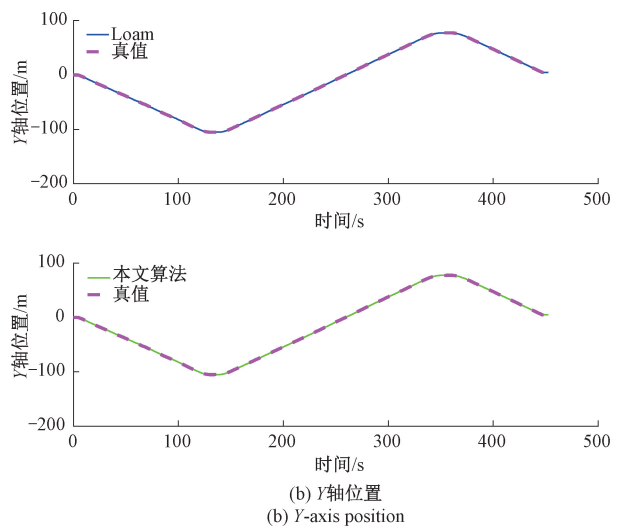
图 9 仿真环境下整体变化检测结果

Fig. 9 Total change detection result in simulation

在仿真中设置了两处变化场景,对集装箱进行了搬动、移除以及增加,每个集装箱体积为 $2\text{ m}\times 2\text{ m}\times 10\text{ m}$ 。由图 9 可以看出,本文算法可对场景变化进行有效检测。

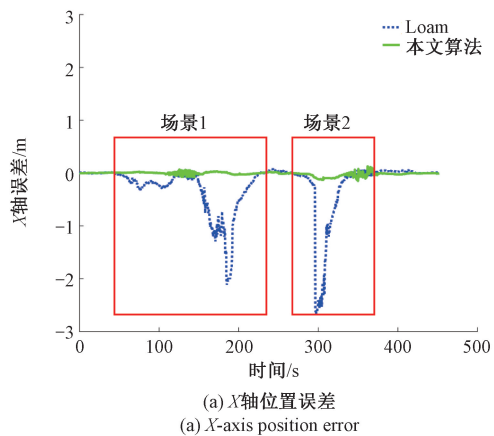


(a) X轴位置
(a) X-axis position



(b) Y轴位置
(b) Y-axis position

图 10 仿真环境下定位结果
Fig. 10 Positioning result in simulation



(a) X轴位置误差
(a) X-axis position error

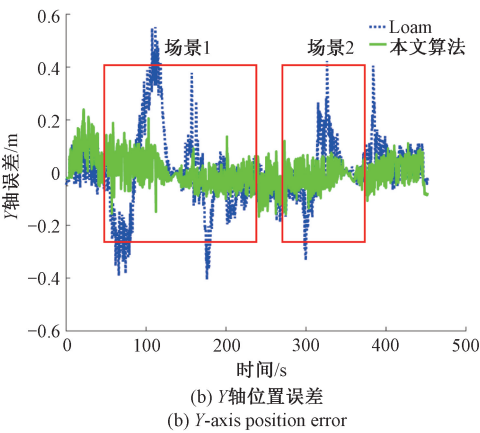


图 11 仿真环境下定位误差对比

Fig. 11 Comparison of positioning error in simulation

表 1 仿真环境下 Loam 和本文算法定位误差对比

Table 1 The comparison of Loam and the proposed algorithm about position error in simulation

方法	总位置 RMSE/m	场景 1 最大误差/m	场景 2 最大误差/m
Loam	0.141	2.114	2.665
本文算法	0.062	0.094	0.163

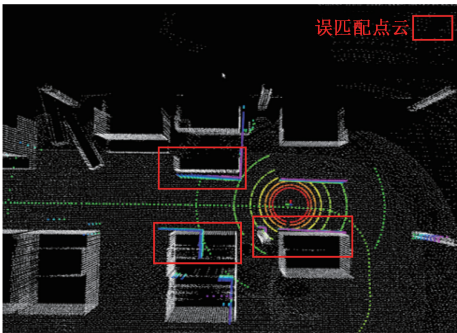
由图 11、表 1 可以看出:1) 基于 Loam 的地图匹配算法全程定位均方根误差 (root mean square error, RMSE) 为 0.141 m。该方法在场景无变化时保持了较高的定位精度,但当无人车经过场景变化区域时其误差迅速增大,最大误差达到 2.67 m。该精度已无法满足无人车自主运行的需求,其主要原因为场景变化引起了点云误匹配,如图 12 所示。2) 本文算法在运行全程均保持了较高的定位精度,全程定位 RMSE 为 0.062 m,最大误差为 0.24 m。当无人车运行至变化场景时,定位精度并未受到影响。

为对变化检测精度进行验证,本文选取场景 1 中两个点进行定位精度对比,如图 13 所示。对变化点的定位结果如表 2 所示。可以看出,本文算法可对场景中变化点进行有效检测,误差优于 0.15 m。

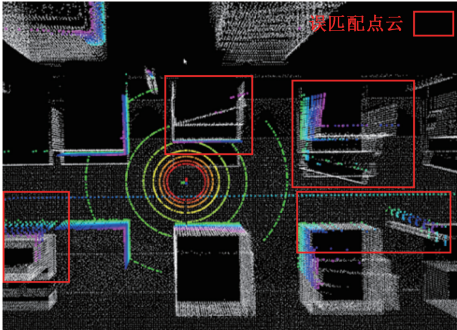
表 2 仿真环境下 Loam 和本文算法定位误差对比

Table 2 The comparison of Loam and the proposed algorithm about position error in simulation

序号	真值	测量值	误差
A 点	(-34.70, -87.70)	(-34.57, -87.76)	0.14
B 点	(-34.70, -99.70)	(-34.73, -99.84)	0.15



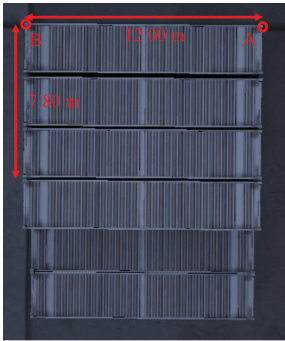
(a) 场景1误匹配情况
(a) Fault matching in the first scenario



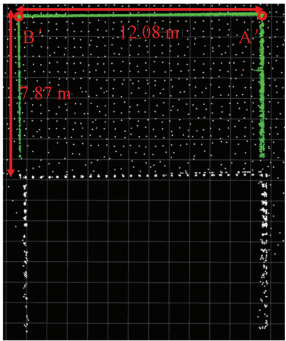
(b) 场景2误匹配情况
(b) Fault matching in the second scenario

图 12 仿真环境下误匹配结果

Fig. 12 Fault matching result in simulation



(a) 仿真场景
(a) Simulation scenes



(b) 变化检测结果
(b) The result of change detection

图 13 检测精度示意图

Fig. 13 The diagram of detection accuracy

3.2 试验验证与分析

本文基于走廊环境搭建了试验场景,通过纸箱、沙发的移动模拟了场景变化。在试验中,无人车平台搭载了 Velodyne 16 线激光雷达,并采用 Leica MS60 全站仪作为位置基准,其定位精度优于 0.01 m,如图 14 所示。本文所采用的计算机 CPU 型号为 Intel i7-8850 H,操作系统为 Ubuntu 18.04。场景变化检测结果如图 15 所示,定位误差对比如图 16 和表 3 所示。

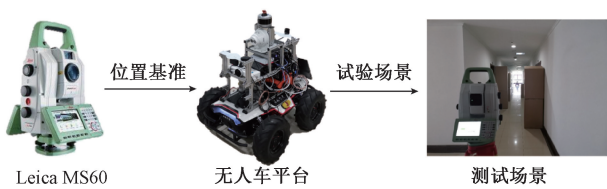


图 14 实验设计框图
Fig. 14 Experiment design framework

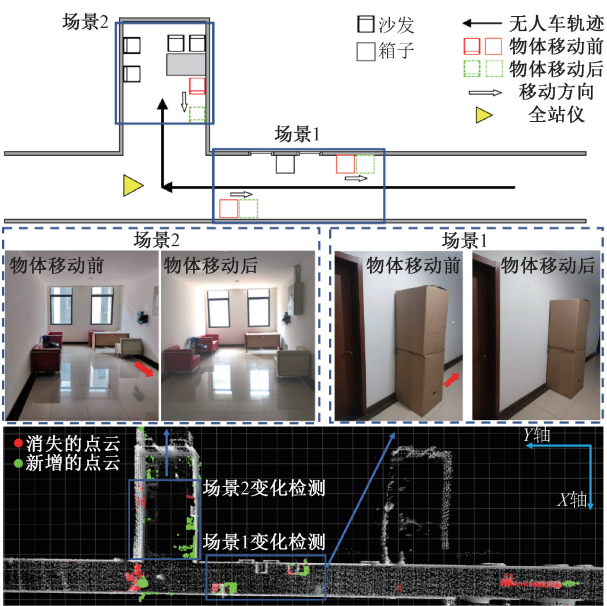


图 15 实验场景整体检测效果
Fig. 15 Total change detection result in experiment

在试验过程中设置了两处变化场景,在场景 1 对体积为 $0.6\text{ m}\times0.6\text{ m}\times2\text{ m}$ 的纸箱移动了 0.6 m ,在场景 2 对沙发移动了 1 m 。由图 15 可以看出,本文算法可对场景 1、2 的变化情况进行有效检测。由图 16 和表 3 可以看出:1) 基于 Loam 的地图匹配算法全程定位 RMSE 为 0.051 m 。该方法在场景无变化情况下保持了较高的定位精度,在场景变化 1 中最大误差为 0.175 m ,场景变化 2 中最大误差为 0.051 m 。可以看出,不同变化场景引起的匹配误差有所区别。由于 Loam 算法具有一定鲁棒性,当场景变化小时其仍能保持较高匹配精度,而在场景变化较大时其会引起误匹配(场景变化 1 的匹配情况如图 17 所示)。2) 本文提出的算法在运行全程均保持了较高的定位精度,全程定位 RMSE 为 0.029 m ,最大误差为 0.104 m 。当无人车运行至变化场景时,定位精度并未受到影响。此外,对比试验与仿真结果,可以看出试验定位精度相较仿真有所提高,主要由于走廊比码头场景小,使得激光雷达匹配精度更高。

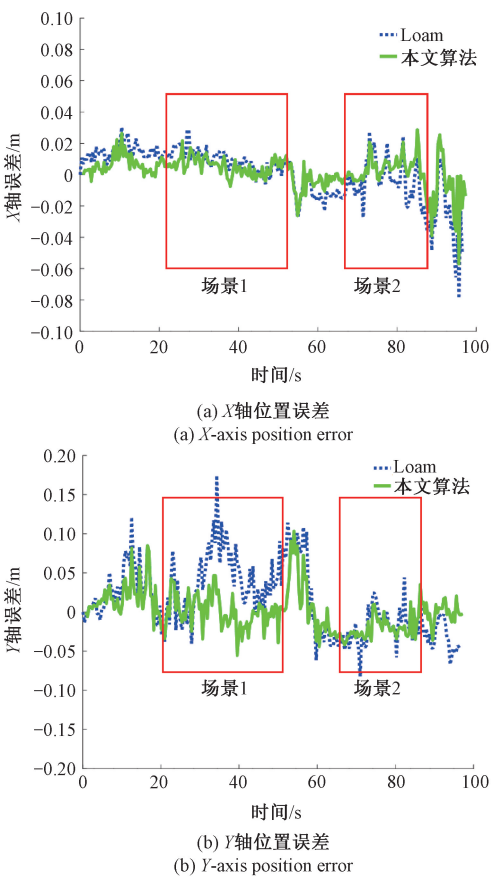


图 16 实验场景下定位误差对比
Fig. 16 Comparison of positioning error in experiment

表 3 实验场景下 Loam 和本文算法的 RMSE 和最大误差对比
Table 3 The compared of Loam and ours algorithm about RMSE and max-error in experiment

方法	总位置 RMSE/m	场景 1 最大误差/m	场景 2 最大误差/m
Loam	0.051	0.175	0.051
本文算法	0.029	0.057	0.039

本文算法与 Loam 算法的耗时以及 CPU 占用对比如表 4 所示。本文算法相较于 Loam 算法 CPU 占用率增加了 5%,算法耗时增加了 148 ms (其中包括变化检测耗时 16 ms ,误检测滤波耗时 130 ms ,其他耗时 2 ms),额外耗时主要用于误检测滤波中的最近点搜索。虽然本文算法耗时有所增加,然而通过融合激光里程计,可将导航解算输出频率提高至 10 Hz ,能够满足工业无人车实时性需求。

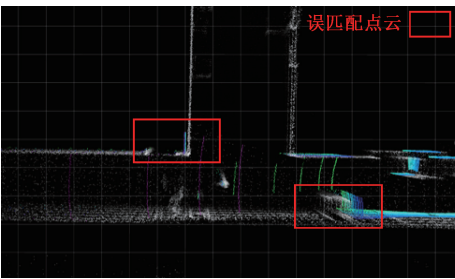


图 17 场景 1 的误匹配情况

Fig. 17 Fault matching result in the first scenario

表 4 Loam 和本文算法的 CPU 占用和耗时对比
Table 4 The compared of Loam and ours algorithm
about CPU usage and time consumption

方法	CPU 占用率/%	算法耗时/ms
Loam	9.0	150
本文算法	14.0	298

4 结 论

本文面向场景变化引起的激光雷达地图匹配精度下降问题,提出了一种激光雷达鲁棒定位与地图维护方法。该方法通过检测与剔除变化点,实现了激光雷达鲁棒地图匹配;基于因子图架构,实现了激光雷达里程计与地图匹配定位结果的有效融合;通过变化检测、误检测滤波、地图更新,实现了对变化场景下地图的长期维护。通过仿真与试验进行了验证,结果表明本文算法提高了变化场景下的激光雷达定位精度,对于无人车的长期自主运行具有较好参考意义。

参考文献

[1] GARZÓN M, VALENTE J, ZAPATA D, et al. An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas [J]. Sensors, 2013, 13(1): 1247-1267.

[2] GEORGY J, KARAMAT T, IQBAL U, et al. Enhanced MEMS-IMU/odometer/GPS integration using mixture particle filter [J]. GPS solutions, 2011, 15 (3): 239-252.

[3] 沈凯, 刘庭欣, 左思琪, 等. 复杂城市环境下 GNSS/INS 组合导航可观测度分析及鲁棒滤波方法 [J]. 仪器仪表学报, 2020, 41(9): 252-261.

SHEN K, LIU T X, ZUO S Q, et al. Observability

analysis and robust fusion algorithms of GNSS / INS integrated navigation in complex urban environment [J]. Chinese Journal of Scientific Instrument, 2020, 41(9): 252-261.

[4] 王家恩, 肖献强. 基于 QR 码视觉定位的移动机器人复合导航方法研究 [J]. 仪器仪表学报, 2018, 39(8): 230-238.

WANG J EN, XIAO X Q. Mobile robot integrated navigation method based on QR code vision positioning [J]. Chinese Journal of Scientific Instrument, 2018, 39(8): 230-238.

[5] XU H G, WANG C X, YANG R Q, et al. Extended Kalman filter based magnetic guidance for intelligent vehicles [C]. 2006 IEEE Intelligent Vehicles Symposium, IEEE, 2006: 169-175.

[6] 李帅鑫, 李广云, 周阳林, 等. 改进的单目视觉实时定位与测图方法 [J]. 仪器仪表学报, 2017, 38(11): 2849-2857.

LI SH X, LI G Y, ZHOU Y L, et al. Improved monocular simultaneous localization and mapping solution [J]. Chinese Journal of Scientific Instrument, 2017, 38(11): 2849-2857.

[7] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: A versatile and accurate monocular SLAM system [J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.

[8] QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator [J]. IEEE Transactions on Robotics, 2018, 34 (4): 1004-1020.

[9] KOHLBRECHER S, VON STRYK O, MEYER J, et al. A flexible and scalable SLAM system with full 3D motion estimation [C]. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, IEEE, 2011: 155-160.

[10] GRISETTI G, STACHNISS C, BURGARD W. Improved techniques for grid mapping with rao-blackwellized particle filters [J]. IEEE Transactions on Robotics, 2007, 23(1): 34-46.

[11] HESS W, KOHLER D, RAPP H, et al. Real-time loop

- closure in 2D LIDAR SLAM [C]. 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016: 1271-1278.
- [12] ZHANG J, SINGH S. LOAM: Lidar odometry and mapping in real-time [C]. Robotics: Science and Systems, 2014.
- [13] SHAN T, ENGLLOT 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), IEEE, 2018: 4758-4765.
- [14] KIM D, CHUNG T, YI K. Lane map building and localization for automated driving using 2D laser rangefinder [C]. 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2015: 680-685.
- [15] WANG L, ZHANG Y, WANG J. Map-based localization method for autonomous vehicles using 3D-LIDAR[J]. IFAC-PapersOnLine, 2017, 50(1): 276-281.
- [16] LEVINSON J, THRUN S. Robust vehicle localization in urban environments using probabilistic maps[C]. 2010 IEEE International Conference on Robotics and Automation, IEEE, 2010: 4372-4378.
- [17] DING X, WANG Y, LI D, et al. Laser map aided visual inertial localization in changing environment[C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018: 4794-4801.
- [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), IEEE, 2020: 10765-10771.
- [19] 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), IEEE, 2020: 10758-10765.
- [20] JINNO I, SASAKI Y, MIZOGUCHI H. 3D map update in human environment using change detection from LIDAR equipped mobile robot[C]. 2019 IEEE/SICE International Symposium on System Integration (SII), IEEE, 2019: 330-335.
- [21] EGGER P, BORGES P V K, CATT G, et al. Posemap: Lifelong, multi-environment 3d lidar localization [C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018: 3430-3437.
- [22] HIMMELSBACH M, HUNDELSHAUSEN F V, WUENSCH H J. Fast segmentation of 3D point clouds for ground vehicles[C]. 2010 IEEE Intelligent Vehicles Symposium, IEEE, 2010: 560-565.
- [23] BOGOSLAVSKIY I, STACHNISS C. Fast range image-based segmentation of sparse 3D laser scans for online operation[C]. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2016: 163-169.
- [24] 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.
- [25] 王开瑞, 张永德, 刘云辉. 基于惯性导航与电磁导航融合的靶器官运动跟踪方法研究[J]. 仪器仪表学报, 2020, 41(11): 177-187.
- WANG K R, ZHANG Y D, LIU Y H. Research on target organ motion tracking method based on the fusion of inertial navigation and electromagnetic navigation[J]. Chinese Journal of Scientific Instrument, 2020, 41(11): 177-187.
- [26] LI R, LIU J, ZHANG L, et al. LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments[C]. 2014 DGON Inertial Sensors and Systems (ISS), IEEE, 2014: 1-15.
- [27] DEILAMSALEHY H, HAVENS T C. Sensor fused three-dimensional localization using IMU, camera and LiDAR[C]. 2016 IEEE Sensors, IEEE, 2016: 1-3.
- [28] QIN C, YE H, PRANATA C E, et al. Lins: A lidar-inertial state estimator for robust and efficient navigation[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020: 8899-8906.
- [29] LE GENTIL C, VIDAL-CALLEJA T, HUANG S. 3d lidar-imu calibration based on upsampled preintegrated

- measurements for motion distortion correction[C]. 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018: 2149-2155.
- [30] GENEVA P, ECKENHOFF K, YANG Y, et al. LIPS: Lidar-inertial 3d plane slam [C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018: 123-130.
- [31] SHAN T, ENLOT B, MEYERS D, et al. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping[C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020: 5135-5142.
- [32] 姚二亮, 张合新, 张国良, 等. 基于 Vision-IMU 的机器人同时定位与地图创建算法[J]. 仪器仪表学报, 2018, 39(4): 230-238.
- YAO ER L, ZHANG H X, ZHANG G L, et al. Robot simultaneous localization and mapping algorithm based on vision and IMU [J]. Chinese Journal of Scientific Instrument, 2018, 39(4): 230-238.
- [33] KAESS M, JOHANSSON H, ROBERTS R, et al. iSAM2: Incremental smoothing and mapping using the Bayes tree [J]. The International Journal of Robotics Research, 2012, 31(2): 216-235.
- [34] SHAH S, DEY D, LOVETT C, et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles [C]. Field and Service Robotics, Springer, Cham, 2018: 621-635.

作者简介



吕品(通信作者), 分别于 2008 年和 2015 年于南京航空航天大学获得学士和博士学位, 现为南京航空航天大学副教授, 主要研究方向为惯性导航、多信息可靠融合导航

及感知技术。

E-mail: lvpin@nuaa.edu.cn

Lyu Pin (Corresponding author) received his B. Sc. degree and Ph. D. degree both from Nanjing University of Aeronautics and Astronautics (NUAA) in 2008 and 2015, respectively. He is currently an associate professor at NUAA. His main research interests include inertial navigation, multi-information reliable fusion navigation and perception technology.