

动态场景中的多传感器融合 SLAM 算法研究 *

吴永豪 李胜 邹文成

(南京理工大学自动化学院 南京 210094)

摘要:针对机器人在动态复杂环境下轨迹漂移以及无法建立静态地图问题,设计了一种动态点云去除的多传感器融合同步定位与建图(SLAM)算法。算法前端利用惯性测量单元(IMU)预积分实现点云畸变补偿,并采用迭代误差状态卡尔曼滤波(IESKF)算法在前端获得初始位姿估计。针对动态物体干扰,提出了一种结合地面分割和时空法向量分析的动态点云去除策略,有效剔除了动态目标的影响,保证了静态地图的全局一致性。后端基于因子图优化,融合激光惯导里程计、IMU与编码器预积分,并引入地平面因子,通过多重约束有效抑制了累积误差和Z轴漂移问题。在校园实测的复杂动态环境中,该算法相较于LeGO-LOAM、FAST-LIO和LIO-SAM主流SLAM方案,定位均方根误差(RMSE)分别降低了46.2%、49.4%和35.9%,同时有效地去除了地图中的动态点云,验证了算法的优越性,为复杂动态环境下机器人的自主导航与精确建图提供了可靠的技术支撑。

关键词:多传感器融合;动态场景;激光SLAM;地面分割;因子图

中图分类号: TP242.6; TN958.98 文献标识码: A 国家标准学科分类代码: 510.70

Research on multi-sensor fusion SLAM algorithm in dynamic scenes

Wu Yonghao Li Sheng Zou Wencheng

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: To address the challenges of robot trajectory drift in dynamic and complex environments, and overcome the limitations of conventional static map construction, we propose a robust multi-sensor fusion SLAM algorithm integrated with dynamic point cloud removal. Our front-end processing employs IMU pre-integration to compensate for point cloud distortion and utilizes an iterative error state Kalman filter (IESKF) for refined initial pose estimation. Furthermore, we introduce a novel dynamic point cloud removal strategy that combines ground segmentation with spatio-temporal normal vector analysis. It effectively eliminates moving objects and preserves static structures to ensure global map consistency. On the back end, our method leverages factor graph optimization, fusing laser-inertial odometry, IMU pre-integration, and wheel encoder data to enhance trajectory estimation. In addition, we incorporate ground plane constraints to suppress cumulative errors and mitigate z-axis drift. Experimental validation in a complex campus environment demonstrates that our method significantly reduces positioning root mean square error (RMSE) by 46.2%, 49.4%, and 35.9% compared to LeGO-LOAM, FAST-LIO, and LIO-SAM, respectively. Moreover, our method successfully removes dynamic point clouds from the constructed maps, showcasing superior robustness in dynamic scenarios. These advancements provide reliable support for autonomous robot navigation and high-precision mapping in complex dynamic environments.

Keywords: multi-sensor fusion; dynamic scenes; ground segmentation; lidar SLAM; factor graph

0 引言

同步定位与建图 (simultaneous localization and mapping, SLAM) 是自主移动机器人实现自主导航和环境感知的核心技术, 它使机器人能够在未知环境中通过传感器数据同时进行自身定位和地图构建^[1]。SLAM 系统广泛应用于自动驾驶、室内服务机器人等领域, 精确的地图和实时的定位信息是保证系统正常运行和高效决策的基础。与视觉 SLAM 相比, 激光雷达在光照变化较大的环境中表现更为稳定, 且能够直接提供三维空间信息, 更适合用于室外复杂环境中的移动机器人定位与建图^[2]。

近年来, 研究人员提出了许多基于 3D 激光雷达的 SLAM 算法。LOAM (lidar odometry and mapping) 算法^[3]作为经典的激光 SLAM 方法, 其主要由 3 个模块组成, 分别为点云特征提取、激光里程计以及地图构建。基于曲率特征分析筛选边缘与平面几何特征点, 结合迭代最近点 (ICP) 配准算法完成特征匹配, 构建高精度定位建图系统, 然而系统在长期运行中仍面临位姿估计漂移问题^[4]。LeGO-LOAM^[5] 算法对特征提取进行了轻量化处理, 用地平面分割技术优化计算效率, 并引入回环检测模块来减少长期累计漂移。为提高机器人定位系统鲁棒性, 基于多传感器数据融合的 SLAM 技术逐渐成为发展趋势。LIO-SAM^[6] 算法前端延续 LOAM 提取特征点的思想, 通过引入惯性测量单元 (IMU) 数据实现点云畸变校正, 同时为激光里程计提供位姿初始化参数; 后端构建因子图优化框架, 融合激光-惯导里程计约束、闭环检测约束以及可选融合 GPS 观测数据, 通过全局位姿优化有效抑制里程计漂移误差。FAST-LIO^[7] 是针对固态激光雷达和 IMU 进行紧耦合的算法, 基于 IESKF 迭代误差状态卡尔曼滤波融合激光和 IMU 数据, 并提出了新的卡尔曼增益计算方法, 不再考虑观测维度而是使用状态维度, 大大降低了计算量。FAST-LIO^[8] 在 FAST-LIO 的基础上进行了改进, 引入了动态 KD-Tree 存储点云数据, 提高了算法的运行效率, 适用于更多应用场景。LIW-OAM^[9] 是一种紧耦合算法, 基于光束平差 (BA) 的优化框架, 融合 LiDAR、IMU 和轮速编码器数据, 从而产生准确的里程计信息。

激光雷达 SLAM 的研究大多基于静态场景假设^[10], 然而现实场景中普遍存在动态干扰因素, 尤其在城市道路等开放环境中, 移动车辆和行人等动态物体会在激光雷达建图中留下“鬼影”, 导致生成的地图无法准确反映静态环境^[11]。此外, 大量的动态物体的干扰可能导致点云匹配失败影响机器人定位的稳定性和可靠性。针对动态点云的识别与去除, 常用的方法有基于分割的方法、基于光线追踪的方法、基于可见性的方法和基于学习的方法

法^[12]。基于分割的方法受限于标注数据质量和训练结果, 容易受标注误差或未识别动态目标的干扰。OctoMap^[13] 和 Peopleremover^[14] 等基于体素的光线追踪方法能有效构建场景信息, 但基于体素的方法需要消耗大量的内存和计算资源, 导致难以实时在线运行。Remover^[15] 方法作为常见的可见性方法, 其首先将原始点云转换为深度图的投影, 再通过对比局部地图与当期扫描数据间的可见性来去除动态物体。基于可见性的方法存在近地面反射噪点的问题, 导致静态地面点误判为是动态点云。基于深度学习的方法需要高昂的物体标注与训练成本, 只对特定场景适用, 缺乏泛化能力^[16]。

针对室外大规模动态场景下 SLAM 定位精度与建图质量问题。开展了多传感器融合的激光雷达 SLAM 算法以及实时动态点云去除算法的研究。首先在前端利用 IMU 预积分进行点云畸变补偿, 利用 IESKF 算法融合激光雷达与 IMU 信息, 获得机器人位姿初值。然后在动态物体去除算法中分割地面点与非地面点, 使用改进的在线动态点云去除方法去除非地面点中的动态点。最后, 为实现 SLAM 系统定位精度与地图全局一致性的协同优化, 后端优化模块采用基于因子图的联合优化框架^[17]。将前端里程计因子、IMU 与编码器预积分因子、地平面因子作为约束因子, 有效地限制了高度方向的漂移^[18], 在线获得优化后的位姿信息与全局一致的静态地图。

1 前端里程计

动态环境多传感器融合 SLAM 算法流程如图 1 所示, 主要分为 3 个模块。1) 前端里程计模块, 首先对 IMU、LiDAR 及轮速计进行数据同步与校准, 通过 IMU 反向传播推算相邻雷达帧间相对位姿, 并完成点云去运动畸变处理, 确保数据时空对齐; 分别对 IMU 和轮速编码器进行预积分计算。利用迭代误差扩展卡尔曼滤波 (IESKF) 算法, 融合去畸变后的雷达点云数据与 IMU 预积分数据, 为后面动态点云去除模块输出里程计预测的位姿初值。2) 动态点云去除模块, 首先采用同心圆模型的方法对地面进行分割; 接着根据前端提供的里程计的初始位姿, 对时空法向量进行分析, 识别并去除动态点云, 进而消除其对建图的影响。3) 后端因子图优化模块, 利用因子图融合地平面因子、前端里程计因子、IMU 和轮速编码器因子, 最终获得高质量的静态地图。

多传感器融合算法使用多个坐标系, 包括相对传感器坐标和全局坐标。定义世界坐标系为 $\{W\}$, 将机器人机体坐标系表示为 $\{B\}$, 并假设 IMU 传感器坐标与机体坐标系一致。机器人系统状态 X 定义为:

$$\begin{aligned} X &= [x_0, x_1, \dots, x_n] \\ x_k &= [{}^W\mathbf{R}_k, {}^W\mathbf{p}_k, {}^W\mathbf{v}_k, \mathbf{b}_k]^T, \quad k \in [0, n] \end{aligned} \quad (1)$$

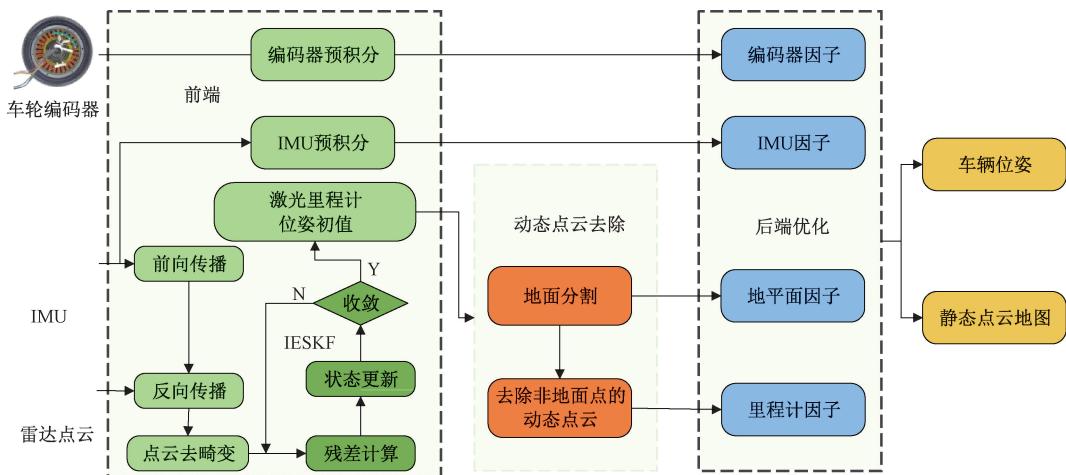


图 1 算法流程

Fig. 1 Diagram of the algorithm

式中: \mathbf{x}_k 表示第 k 个关键帧机器人的运动状态;

${}^w\mathbf{R}_k \in SO(3)$ 代表世界坐标系下机体旋转矩阵;

${}^w\mathbf{p}_k \in R^3$ 表示世界坐标系下机器人位置; ${}^w\mathbf{v}_k \in R^3$ 表示世界坐标系下机器人速度; \mathbf{b}_k 表示 IMU 六轴参数的零偏。

1.1 IMU 和编码器预积分

在时刻, IMU 的原始加速度计和陀螺仪测量值在 IMU 坐标系中测量值定义为:

$$\begin{aligned}\hat{\mathbf{a}}_t &= {}^B\mathbf{R}_t({}^w\mathbf{a}_t - \mathbf{g}) + \mathbf{b}_t^a + \mathbf{n}_t^a \\ \hat{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \mathbf{b}_t^\omega + \mathbf{n}_t^\omega\end{aligned}\quad (2)$$

式中: \mathbf{a}_t 和 $\boldsymbol{\omega}_t$ 分别表示加速度和角速度的真实值; \mathbf{b}_t^a 和 \mathbf{b}_t^ω 分别表示 IMU 加速度计和陀螺仪的零偏; \mathbf{n}_t^a 和 \mathbf{n}_t^ω 则则对应二者白噪声; ${}^B\mathbf{R}_t^w$ 为世界坐标系至 IMU 坐标系的旋转变换。

根据上述 IMU 测量数据可以推导出机器人的动态运动状态。机器人在时间 $t + \Delta t$ 时的速度、位置和旋转计算公式如下:

$${}^w\mathbf{R}_{t+\Delta t} = {}^w\mathbf{R}_t^B e^{(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_t^\omega - \mathbf{n}_t^\omega) \Delta t} \quad (3)$$

$${}^w\mathbf{v}_{t+\Delta t} = {}^w\mathbf{v}_t + ({}^w\mathbf{g} + {}^w\mathbf{R}_t^B(\hat{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{n}_t^a)) \Delta t^2 \quad (4)$$

$${}^w\mathbf{p}_{t+\Delta t} = {}^w\mathbf{p}_t + {}^w\mathbf{v}_t \Delta t + \frac{1}{2} ({}^w\mathbf{g} + {}^w\mathbf{R}_t^B(\hat{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{n}_t^a)) \Delta t^2 \quad (5)$$

假设 IMU 角速度和加速度积分过程中保持不变, 其中 ${}^w\mathbf{R}_t^B = ({}^B\mathbf{R}_t^w)^T$, 表示 IMU 坐标系到世界坐标系的旋转矩阵。

车辆控制单元 (VCU) 通过采集编码器输出的脉冲信号确定车轮的转速, 通过如下公式可计算出 t 时刻左后轮线速度 \hat{v}_t^l 和右后轮的线速度 \hat{v}_t^r 为:

$$\begin{bmatrix} \hat{v}_t^l \\ \hat{v}_t^r \end{bmatrix} = \begin{bmatrix} \hat{\tau}_t^l \cdot r_l & 0 & 0 \\ \hat{\tau}_t^r \cdot r_r & 0 & 0 \end{bmatrix} \quad (6)$$

$$\hat{\tau}_t^l = \tau_t^l + n_{\tau_t^l}, \hat{\tau}_t^r = \tau_t^r + n_{\tau_t^r} \quad (7)$$

式中: $\hat{\tau}_t^l$ 和 $\hat{\tau}_t^r$ 分别为测得的左右车轴转速; τ_t^l 和 τ_t^r 分别为左右车轴转速真实值; r_l 和 r_r 分别为左、右车轮半径。假设 $n_{\tau_t^l}$ 和 $n_{\tau_t^r}$ 都是零均值的高斯白噪声, 则车轮编码器坐标系 $\{E\}$ 下的测量模型可以表示为:

$$\hat{v}_t^E = \frac{\hat{v}_t^l + \hat{v}_t^r}{2} + n_{v_t} \quad (8)$$

$$\hat{\omega}_t^E = \frac{\hat{v}_t^l - \hat{v}_t^r}{b} + n_{\omega_t} \quad (9)$$

式中: \hat{v}_t^E 和 $\hat{\omega}_t^E$ 分别表示机器人在 t 时刻的线速度和角速度; b 表示车轮间距; n_{v_t} 和 n_{ω_t} 都是零均值的高斯白噪声。

采用上述编码器模型, 假设机器人当前位姿为 (x_t, y_t, θ_t) , v_t 和 $\boldsymbol{\omega}_t$ 分别表示机器人在 t 时刻的线速度和角速度真值, 对机器人在 $t + \Delta t$ 时间内的编码器数据进行预积分计算:

$$\begin{bmatrix} x_{t+\Delta t} \\ y_{t+\Delta t} \\ \theta_{t+\Delta t} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \cos\theta_t & 0 \\ \sin\theta_t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \Delta t \\ \omega_t \Delta t \end{bmatrix} \quad (10)$$

1.2 点云去畸变

当车载移动时, 机械式激光雷达测量结果会受到运动畸变的影响, 获得的点云信息与真实位置存在差异, 前端里程计中采用 IMU 提供的短时间积分获得位姿估计信息, 结合该帧点云数据中每个点的采集时间, 将当前帧所有点映射至同一坐标系下, 以消除雷达运动失真。

利用 IMU 的积分信息进行反向传播, 推算出每个点

云被采集时的位姿,并利用其将点云去畸变。若 IMU 在激光雷达末尾扫描时刻为末尾时刻为 t_k , 计算此时的 IMU 位姿变换矩阵为 \mathbf{T}_k , 在该段时间内点云为 p^k , 则时间戳为 t_i 的激光矫正后的点云为:

$$\tilde{p}_i^k = \mathbf{T}_k(\mathbf{T}_{t_i})^{-1} p_i^k \quad (11)$$

将每个点投影到扫描末尾时刻坐标系, 利用去畸变的点云进行后续的点云匹配、位姿预测与更新。

1.3 激光惯导里程计

该前端里程计融合 IMU 数据与去除畸变后的激光雷达数据, 前端里程计参考 FAST-LIO2 算法, 采用 IESKF 进行紧耦合优化, 主要步骤如下:

1) 前向传播

设系统上一次关键帧最优位姿为 \mathbf{x}_{k-1} 及其协方差矩阵为 \mathbf{P}_{k-1} , IMU 的测量值 $\mathbf{u}_k = (\hat{a}_k, \hat{\omega}_k)$ 包含加速度与角速度测量值, 两个 IMU 测量采样间隔时间为 Δt 。状态转移方程为 $f(\mathbf{x}_k, \mathbf{u}_k)$, 可以得到 t_k 时刻前向传播更新的预测位姿 $\hat{\mathbf{x}}_k$ 及其预测协方差矩阵 $\hat{\mathbf{P}}_k$:

$$\hat{\mathbf{x}}_k = \mathbf{x}_{k-1} \oplus (f(\mathbf{x}_k, \mathbf{u}_k) \cdot \Delta t) \quad (12)$$

$$\hat{\mathbf{P}}_k = \mathbf{F}_{\tilde{x}} \mathbf{P}_{k-1} \mathbf{F}_{\tilde{x}}^T + \mathbf{F}_w \mathbf{Q} \mathbf{F}_w^T \quad (13)$$

式中: \oplus 表示李群上的广义加法; $\mathbf{F}_{\tilde{x}}$ 为系统预测状态矩阵; \mathbf{F}_w 为噪声雅可比矩阵; \mathbf{Q} 为 IMU 噪声协方差矩阵。

2) 构建观测模型

点面量测模型如图 2 所示, 当前帧点云 p_j 去畸变后通过外参 ${}^w p_L$ 矩阵转换至世界坐标系下, 其中任意一点在地图中搜索最近邻平面, 构建点面残差:

$$0 = h_j(\mathbf{x}_k) = \mathbf{u}_j^T ({}^w T_L p_j - q_j) \quad (14)$$

式中: q_j 表示世界坐标系下该平面上的一个坐标点; \mathbf{u}_j 为法向量法向量。残差反映点到平面的距离, 理想情况下应趋近于 0。

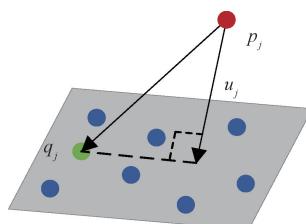


图 2 点面量测模型

Fig. 2 Point-to-plane measurement model

3) 迭代状态更新

迭代更新以不断更新状态误差, 对式(14)进行一阶泰勒展开可得:

$$0 = h_j(\mathbf{x}_k, \mathbf{n}_j) \approx h_j(\hat{\mathbf{x}}_k^\alpha, 0) + \mathbf{H}_j^\alpha \tilde{\mathbf{x}}_k^\alpha + v_j = z_j^\alpha + \mathbf{H}_j^\alpha \tilde{\mathbf{x}}_k^\alpha + v_j \quad (15)$$

式中: $\tilde{\mathbf{x}}_k^\alpha = \mathbf{x}_k \ominus \hat{\mathbf{x}}_k^\alpha$, α 为迭代次数, \ominus 表示李群上的广义减法; \mathbf{H}_j^α 为 $h_j(\mathbf{x}_k, \mathbf{n}_j)$ 的 Jacobin 矩阵; $v_j \in N(0, R_j)$ 为高斯白噪声。

迭代更新状态估计和卡尔曼增益分别为:

$$\tilde{\mathbf{x}}_k^{\alpha+1} = \tilde{\mathbf{x}}_k^\alpha + \mathbf{K}(0 - h(\tilde{\mathbf{x}}_k^\alpha)) - (\mathbf{I} - \mathbf{K}\mathbf{H})(\tilde{\mathbf{x}}_k^\alpha - \tilde{\mathbf{x}}_k^0) \quad (16)$$

$$\mathbf{K} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} + \mathbf{P}^{-1})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \quad (17)$$

4) 收敛输出最优估计

判断 $\alpha + 1$ 次的状态估计与 α 次迭代得到的状态估计的绝对差值是否小于阈值 ε :

$$\|\hat{\mathbf{x}}_k^{\alpha+1} \ominus \hat{\mathbf{x}}_k^\alpha\| \leq \varepsilon \quad (18)$$

若达到指定的阈值 ε , 则迭代完成, 最优位姿估计与更新后的协方差估计分别为:

$$\mathbf{x}_k = \hat{\mathbf{x}}_k^{\alpha+1} \quad (19)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\hat{\mathbf{P}}_k \quad (20)$$

2 动态物体去除算法

2.1 地面分割

地面分割对于减少动态点云识别计算速度和减少雷达垂直方向误差至关重要。传统的地面平面拟合计算要求较低, 但分割精度不足, 对于陡坡、颠簸的道路和路缘等是无效的。本文提出一种针对实时任务的分阶段地面点分割算法, 该算法基于地面平面拟合, 并对 Patchwork++^[19] 进行改进, 以解决细节处的粗糙分割问题, 并实现更详细的点云分割。

该算法基于同心区域模型 (CZM) 以实现点云的地面提取, 同心区域模型如图 3 所示。假设点云的集合 P , 其中每个点 P_i 都包含三维坐标 (x, y, z) 与反射强度 I 。将地面点的集合定义为 G , 将非地面点的集合定义为 N , 则集合 P 表示为:

$$P = \bigcup_i p_i = G \cup N \quad (21)$$

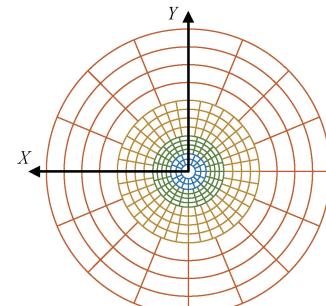


图 3 同心区域模型

Fig. 3 Concentric zone model

首先去除干扰点,提高拟合地平面的有效性。干扰点去除包括删除反射的噪点和垂直的非地面干扰点。如图 4 所示,位于地平面以下的噪点会因为镜面反射成为障碍点。每个扇形环状区域指定高度以下且反射强度低于阈值的点将会被识别和去除。

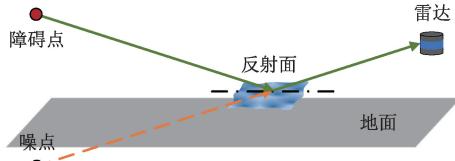


图 4 雷达反射噪点理论模型

Fig. 4 Lidar reflection noise theoretical model

此外,某些扇形环状区域内的垂直点数量会远超过地面点数量,拟合的地平面受垂直干扰点影响,导致拟合结果与实际地面存在显著偏差,如图 5 所示。

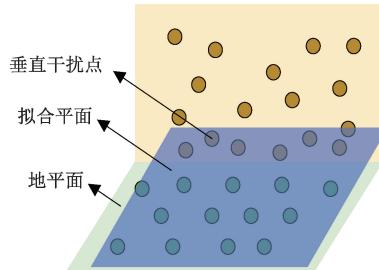


图 5 垂直点对地平面拟合的影响

Fig. 5 Effect of vertical points on ground plane fitting

为了解决上述问题,提出了一种地面分割算法,该算法旨在从每个扇区环区域集合中有效地删除大规模的垂直干扰点。扇区内 z 值大于阈值 z_{th} 的点提取为点集 P_z 。从 P_z 中选取 n_{th} 个最小 z 值的点作为候选点。使用主成分分析(PCA)方法拟合平面,获得单位矢量 (A, B, C) 和平面方程 $Ax + By + Cz + D = 0$ 。如果单位矢量和 z 轴之间的角度 θ_z 小于阈值 θ_{th} ,则拟合平面为垂直平面。计算 P_z 中的每个点到拟合平面的距离 d :

$$d = \frac{|Ax + By + Cz + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (22)$$

如果 d 小于阈值 d_{th} ,则该点被视为垂直干扰点,将其从 P_z 中去除。重新拟合平面,重复上述过程,直到 $\theta_z \leq \theta_{th}$ 或 P_z 中的点数小于 n_{th} ,得到移除垂直干扰点后的扇区点集 P'_z 。

去除干扰点后的点集 P' 重新进行地面平面拟合,计算每个点到平面的距离 d_i ,则:

$$p_i \in \begin{cases} G & i \leq d_{th} \\ N & i > d_{th} \end{cases} \quad (23)$$

最后,基于所有上述处理,将点集 P 分为地面点 G 和

非地点 N 。

2.2 动态物体去除

得到去除地面点后的点云集合 N_i ($i \in 1, \dots, n$) 后,在后端之前设计一种改进的非地面点动态点云去除算法,其不依赖于物体的具体类型,而是仅关注物体是否具有运动特征^[20]。

该方法的核心是通过计算时空法向量来估计每个点的动态得分。时空法线是一个包含空间和时间分量的向量,能够反映点的运动信息。在一维情况下,时空法线直接与物体的速度相关,而在更高维的场景中,时空法线的时间分量表示物体速度在空间法线上的投影。通过这种方式,时空法向量能够提供关于物体运动的有效信息。

图 6 所示为存在动态对象的几何数据时空视图,其中图 6(a)为一维空间图与二维时空图,图 6(b)为二维空间图,图 6(c)为三维时空图。红点是动态点,蓝点是静态点,箭头为法向量。说明了时空法向量与动态点的速度之间的关系。动态点时空法向量的时间分量不为零,这表明它们随着时间发生了运动;静态点的时间分量为 0,即法向量不会随时间变化。

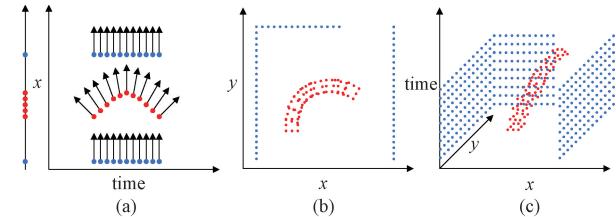


图 6 存在动态对象的几何数据时空视图

Fig. 6 Spatiotemporal view of geometric data with dynamic objects

N_i^j 表示每一帧非地面点集合中的第 j 个点; t_i^j 为该点的时间戳; T_i^j 表示该点的近邻点集合; d 为离差,表示观测与均值的差异。则 j 点局部协方差为:

$$Cov_i^j = \frac{1}{\|T_i^j\|} \sum_{p_u^v \in T_i^j} \mathbf{D} \mathbf{D}^T \quad (24)$$

$$\mathbf{D} = \begin{bmatrix} p_u^v \\ t_u^v \end{bmatrix} - \frac{1}{\|T_i^j\|} \sum_{p_u^v \in T_i^j} \begin{bmatrix} p_u^v \\ t_u^v \end{bmatrix} \quad (25)$$

若 $(f_i^j)_{\min}$ 和 $(\lambda_i^j)_{\min}$,分别代表 Cov_i^j 的最小特征向量和最小特征值,则:

$$Cov_i^j \cdot (f_i^j)_{\min} = (\lambda_i^j)_{\min} \cdot (f_i^j)_{\min} \quad (26)$$

动态得分 s_j 定义为与 Cov_j 的最小特征值相关的特征向量的时间分量的绝对值。

对于一个点,低动态分数表示它在三维空间中处于静止状态或具有非常低的速度,因此根据点的动态分数将其分类为静态点或动态点。动态点去除算法如算法伪代码如下:

动态点去除算法

1. 输入: 前端点云集合 P , 前端里程计获得的位姿变换 T_{odo}
2. 输出: 去除动态点后的静态点云集合 P_{rd}
3. 初始化: 点云数量 n , 体素大小 d_v , 近邻搜索半径 d_r , 动态点阈值 thr
4. 主循环:
5. 提取地面点 G , 提取非地面点 N
6. 对非地面点体素滤波 P_{rd}^v
7. 建立哈希地图 $M(2n + 1)$
8. if $\text{size}(M) > 2n + 1$ then
9. 去除旧的点云数据
10. end
11. 建立滑动窗口 $M(2n + 1)$
12. 建立 kd-tree (M)
13. 选择最新点云 $p = M[n + 1]$
14. for $i = 1: \|p\|$ do
15. 在 kd-tree 中搜索 p_i 近邻点(半径 d_r), 时间戳为 t_i
16. 计算 $score_i(p_i, t_i, n_i)$
17. if $score_i < thr$ then
18. $p_i \rightarrow P_{rd}$
19. end if
20. $P_{rd} = G \cup P_{rd}$
21. return P_{rd}
22. end for

动态点去除算法首先使用地面分割方法, 获得地面点云集合 G 与非地面点云集合 N 。然后利用哈希映射进行体素滤波采样, 体素大小为 d_v 。通过前端的里程计位姿变换 T_{odo} 将 G 与 N 聚合到地图 M 中。 M 为一个滑动窗口, 包含 $2n + 1$ 个点云。当一个新的点云添加到 M 时, 最旧的点云会被移除。使用 M 构建一个 kd-tree, 通过查询每个点在第 $n + 1$ 个最新点云中的近邻点(半径为 d_r)来计算时空法线。最后, 通过阈值 thr 对时空法线的时间分量进行处理, 从而检测并去除动态点, 获得不含有动态点的静态点云 P_{rd} 。

3 后端因子图优化

后端因子图优化的核心是通过最小化所有约束的误差来优化 SLAM 位姿和地图。因子图将系统中待优化变量节点和因子表示为节点, 将系统中的约束关系表示为边。求解因子图的过程旨在根据各因子的观测数据 Z 来确定系统状态量 X 的最优值, 这等价于求解一个最大后验概率估计问题:

$$X^{MAP} = \underset{X}{\text{argmax}} p(X \mid Z) = \underset{X}{\text{argmax}} \prod_i p_i(x_i, z_i) \quad (27)$$

将其转化为求解最小二乘问题:

$$X^{MAP} = \underset{X}{\text{argmin}} \sum_i \|e_i(x)\|_{\Sigma_i}^2 \quad (28)$$

$$e_i(x) = z_i - h_i(x) \quad (29)$$

式中: z_i 表示观测值; $h_i(x)$ 表示预测值; Σ_i 为各个因子

的协方差矩阵。

后端因子图优化融合了激光惯导里程计因子、地平面因子、IMU 以及编码器预积分因子, 如图 7 所示。

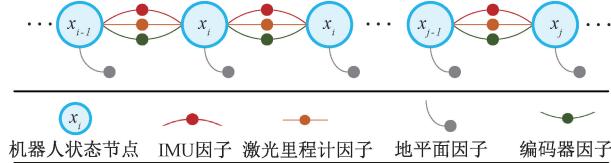


图 7 后端基于因子图优化的框架

Fig. 7 Backend framework based on factor graph optimization

3.1 前端里程计因子

前端里程计通过 IESKF 算法融合激光雷达与 IMU 数据, 能够生成高频低精的位姿信息。然而激光惯导里程计因子频繁加入因子图进行优化会消耗大量计算资源, 因此采用了关键帧选择策略, 从而大大改善了计算效率并确保算法可以实时运行在较大规模的地图上。当机器人当前帧相对于前一个关键帧的位姿变动超出预设的阈值时, 该当前帧即被判定为关键帧 F_k^L , 并被纳入滑动窗口进行处理。

构建的激光惯导里程计因子残差 e_L 可表示为:

$$e_L = z_k - h_L(x_k, x_{k+1}) \quad (30)$$

式中: $h_L(\cdot)$ 为机器人状态节点 x_k 和 x_{k+1} 之间的相对位姿变换预测值; z_k^L 为前端激光惯导里程计量位姿测值。

3.2 IMU 和编码器预积分因子

激光雷达关键帧 F_k^L 和 F_{k+1}^L 之间的 IMU 测量方法, 根据 IMU 预积分模型, 可得 IMU 里程计残差模型为:

$$e_I = \begin{bmatrix} R_W^{I_k}(p_{I_{k+1}}^W - p_{I_k}^W - p_{I_k}^W \Delta t_k + \frac{1}{2} g^W \Delta t_k^2) - {}^I p_{I_{k+1}}^{I_k} \\ R_W^{I_k}(v_{I_{k+1}}^W - v_{I_k}^W + g^W \Delta t_k) - {}^I v_{I_{k+1}}^{I_k} \\ 2[(q_{I_k}^W)^{-1} \otimes q_{I_{k+1}}^W \otimes ({}^I R_{I_{k+1}}^{I_k})^{-1}]_{xyz} \\ b_{a_{k+1}} - b_{a_k} \\ b_{\omega_{k+1}} - b_{\omega_k} \end{bmatrix} \quad (31)$$

其中 $[p_{I_{k+1}}^W, v_{I_{k+1}}^W, R_{I_{k+1}}^{I_k}]^T_{xyz}$ 分别为 IMU 预积分获得的位置、速度、姿态测量值, $[\cdot]_{xyz}$ 为姿态残差的四元数矢量运算, 在实验前离线获得了加速度计和陀螺仪的零偏以及机器人 IMU-LIDAR 的外部参数标定。

使用轮速里程计增量模型, 对运动过程中的关键帧之间的增量进行预积分, 以构成编码器约束因子。对于两个连续的雷达关键帧, 编码器的预积分残差定义为:

$$e_E = \begin{bmatrix} R_W^{I_k}(p_{I_{k+1}}^W - p_{I_k}^W) - {}^E p_{I_{k+1}}^{I_k} \\ R_W^{I_k}(v_{I_{k+1}}^W - v_{I_k}^W) - {}^E v_{I_{k+1}}^{I_k} \end{bmatrix} \quad (32)$$

其中 $[p_{I_{k+1}}^{I_k}, v_{I_{k+1}}^{I_k}]^T$ 为轮速编码器预积分的位置与速度测量值。这个残差反映了编码器测量得到的增量与 IMU 预积分增量之间的差异。

3.3 地平面因子

利用地面分割方法,对提取到的地平面点集构建地面约束,优化相邻帧之间的变换以提高位姿估计的准确性,有效减少 SLAM 高度上的累计误差。地平面因子残差定义为:

$$\mathbf{e}_G = \begin{bmatrix} R_W^{I_k}(p_{z, I_{k+1}}^W - p_{z, I_k}^W) - {}^G p_{z, I_{k+1}}^{I_k} \\ R_W^{I_k}(\theta_{roll, I_{k+1}}^W - \theta_{roll, I_k}^W) - {}^G \theta_{roll, I_{k+1}}^{I_k} \\ R_W^{I_k}(\theta_{pitch, I_{k+1}}^W - \theta_{pitch, I_k}^W) - {}^G \theta_{pitch, I_{k+1}}^{I_k} \end{bmatrix} \quad (33)$$

其中 $[p_{z, I_{k+1}}^{I_k}, \theta_{roll, I_{k+1}}^{I_k}, \theta_{pitch, I_{k+1}}^{I_k}]^T$ 分别表示根据地平面点云计算出的帧间 z 轴、横滚和俯仰角增量。

3.4 因子图优化

将式(30)~(33)代入式(28)中,得到后端定位问题因子图优化最小二乘目标函数为:

$$X^{MAP} = \operatorname{argmin}_X \sum_i (\|\mathbf{e}_L\|_{\Sigma_{L_k^{k+1}}}^2 + \|\mathbf{e}_I\|_{\Sigma_{I_k^{k+1}}}^2 + \|\mathbf{e}_E\|_{\Sigma_{E_k^{k+1}}}^2 + \|\mathbf{e}_G\|_{\Sigma_{G_k^{k+1}}}^2) \quad (34)$$

式中: $\Sigma_{(\cdot)}$ 表示每个残差模型的噪声协方差矩阵。因子图的优化涉及调整变量值以最大程度地减少因子的残差,采用 ISAM2 算法^[21]进行求解。ISAM2 是一种高效的增量式优化算法,适用于 SLAM 系统中大规模因子图优化问题,其能够实时更新优化结果而无需重新优化整个因子图,确保 SLAM 实时性。

4 实验

为验证算法有效性,搭载实验车辆在校园环境不同场景下进行实验分析,评估 SLAM 算法的定位效果以及动态点云的去除效果。

实验平台如图 8 所示,实验车辆为 Ackerman 底盘的 PIX Moving 无人机,机器人平台配备了两组电动机与轮速编码器(100 Hz),左右车轮间距为 0.735 m。车辆搭载 IMU(200 Hz)、镭神 C32 激光雷达,采用华测 RTK 传感器获取的数据作为参考轨迹。

4.1 地平面分割与动态物体去除实验

根据地面点云分割与动态物体去除算法,遥控无人机在室内和室外环境中分别采集了实验数据,获得了不同场景下的实验效果。

图 9(a)和(b)所示分别为 LEGO-LOAM 算法和本文方法在室内外环境下的单帧点云地面分割效果,其中红色为地面点,绿色为非地面点。可以看出根据地面分割



图 8 移动机器人实验平台
Fig. 8 Mobile robot experiment platform

算法所构建的单帧点云,能清晰的区分出地面点与非地面点,地面分割效果更好。

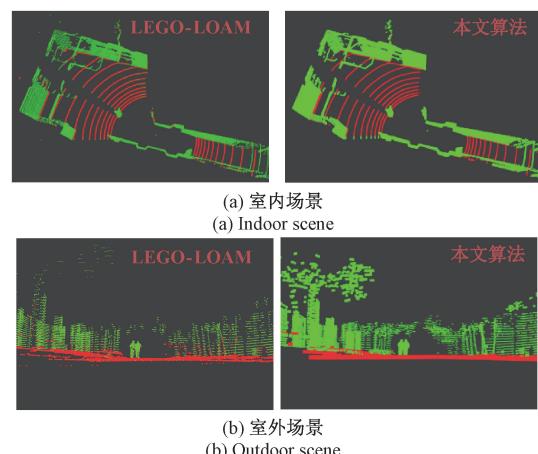


图 9 单帧场景地面分割效果对比图
Fig. 9 Comparison of single-frame ground segmentation results

图 10(a)和(b)所示分别为室内外环境中删除动态点的单帧点云效果。红色点云代表动态点,而绿色点云代表静态点。该方法可以实时检测各种动态物体,包括行人,运动的电动车等。针对室内和室外场景,实验结果表明,所提出的动态物体去除方法能够有效识别并剔除动态点云。

点云地图能够直观地反映算法建图的整体质量,同时展示动态物体去除的效果。图 11 所示分别为 LIO-SAM、FAST-LIO2、LEGO-LOAM 算法与本文算法在动态场景中的建图效果,红色区域内存在动态车辆与行人运动轨迹形成的残影。这些动态物体的存在会对地图的精度和一致性产生干扰。

实验结果表明,该动态物体去除方法能够有效识别

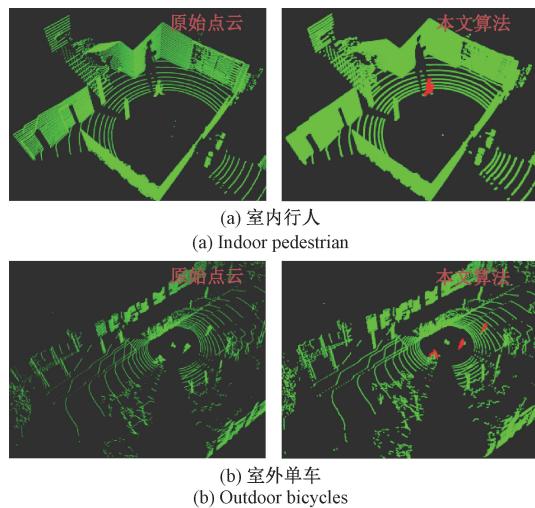


图 10 单帧动态点云去除效果对比

Fig. 10 Comparison of single-frame dynamic point cloud removal results

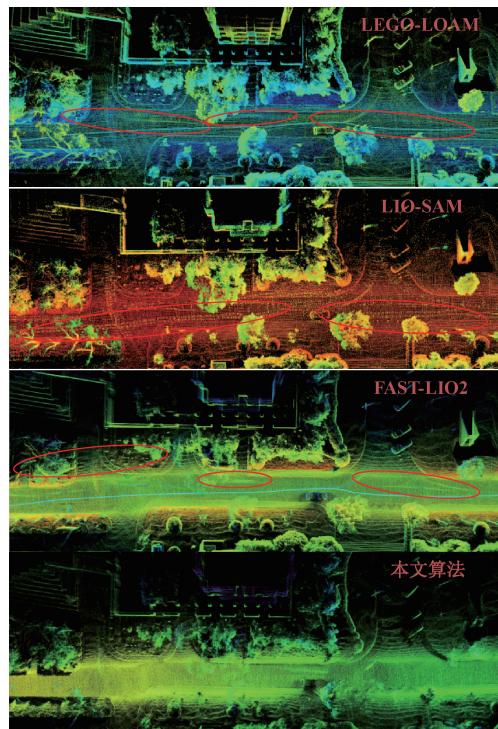


图 11 动态点云去除效果对比

Fig. 11 Comparison of dynamic point cloud removal results

并滤除运动目标,显著减少了点云地图中的行人、车辆等动态残影,提升了地图的静态性和可靠性,使其更加符合真实环境的静态结构。

4.2 定位结果实验

为验证本文算法后端因子图优化模块的有效性,在包含大量动态物体的校园环境中进行了定位与建图实

验,测试场景涵盖室外的上下坡和平地,以及室内的长直走廊等典型环境。

图 12 所示为 LIO-SAM、FAST-LIO2、LEGO-LOAM 与本文算法在校园动态场景中的轨迹对比结果。采用 RTK 数据作为轨迹的真实值,用于评估各 SLAM 算法的定位精度。从图 12 中可以观察到,本文算法与轨迹真值的拟合程度更高,尤其在动态场景和长直路径等退化场景下表现出更高的定位精度。

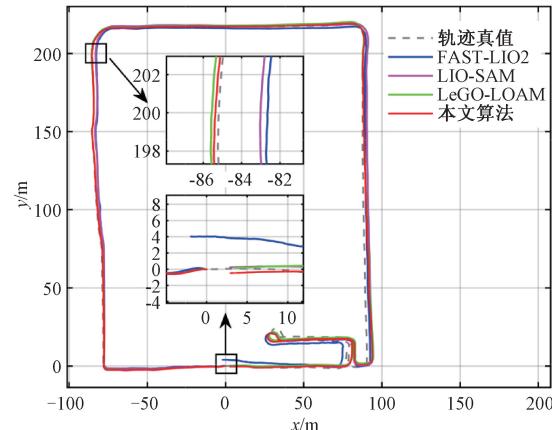


图 12 不同算法轨迹对比

Fig. 12 Comparison of trajectories from different algorithms

表 1 所示为本文算法的轨迹误差最大值、均值、标准差和均方根误差方面与 LIO-SAM、FAST-LIO2、LEGO-LOAM 算法的对比,可以看出本文算法定位最大偏差最小,定位精度更高,误差变动幅度较小。

表 1 定位误差对比

Table 1 Comparison of localization errors (m)

算法	最大值	平均值	标准差	均方根误差
LEGO-LOAM	8.168	4.088	2.491	4.451
FAST-LIO2	9.855	4.196	2.437	4.734
LIO-SAM	6.245	2.931	1.661	3.737
本文	5.687	1.774	1.205	2.395

图 13 所示为本文算法与其他算法在 x 、 y 、 z 方向上的轨迹对比。可以看出其他激光雷达 SLAM 算法在没有回环的情况下, z 轴漂移较大。在后端融入地平面因子约束优化后, z 轴误差明显小于其他算法,与参考轨迹更接近。因此算法中的地面因子可以显著减少环境中的垂直定位误差。

以上实验结果表明,算法由于进行了前端里程计、动态点云去除与后端因子图优化,降低了点云匹配产生的累计误差与 z 轴偏移,累积误差相对于其他算法更低,轨迹漂移减小,具有较好的轨迹闭合效果。在动态的室内外综合场景中,所提算法具有更高的定位精度,能够构建更精确的静态地图。图 14 所示为基于所提方法构建的

点云地图在卫星地图上的映射效果。

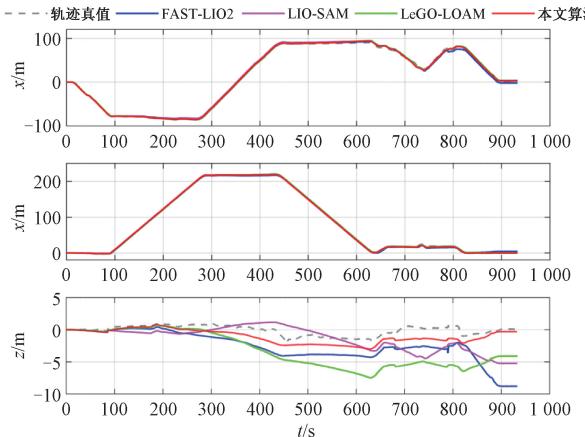


图 13 各轴位移变化曲线

Fig. 13 Displacement change curve of each axis

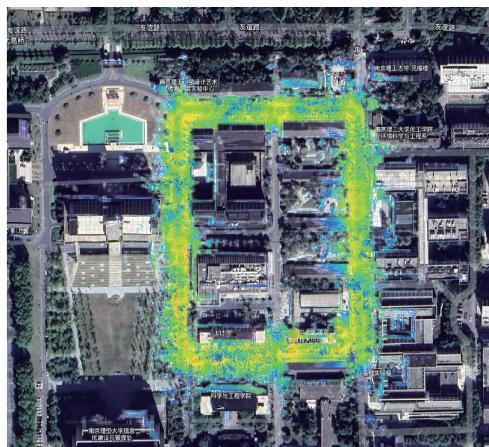


图 14 点云地图在卫星地图的映射

Fig. 14 Point cloud map mapping to satellite map

5 结 论

本文针对室外大规模动态场景下机器人定位与建图面临的问题,提出了一种基于多传感器融合的激光雷达 SLAM 算法,并结合实时地面分割与动态点云去除技术,构建了一个集前端里程计、动态物体识别与去除以及后端因子图优化的 SLAM 算法。实验结果表明,所提出的地平面因子有效地约束了激光雷达 SLAM 中 z 轴定位的漂移。动态物体去除方法通过精准识别并剔除运动目标,显著减少了点云地图中行人、车辆等动态残影,从而为后续的定位、导航和环境感知提供了更高质量的静态地图基础。未来将聚焦于提升动态物体识别的准确性与实时性,融合语义信息深化环境理解,推动算法在更复杂多变场景中的适配。

参考文献

- [1] 王伟. 动态环境下基于激光雷达的建图与定位算法研究 [D]. 北京: 北京化工大学, 2023.
- [2] 邓鹏, 罗静. 复杂环境下机器人多传感器融合定位方法 [J]. 电子测量与仪器学报, 2023, 37 (12): 48-57.
- [3] DENG P, LUO J. Robot multi-sensor fusion localization method in complex environment [J]. Journal of Electronic Measurement and Instrumentation, 2023, 37 (12): 48-57.
- [4] ZHANG J, SINGH S. LOAM: Lidar odometry and mapping in real-time [C]. Robotics: Science and Systems, 2014, 2(9): 1-9.
- [5] 路春晓, 钟焕, 刘威, 等. 复杂地形环境下的多传感器融合 SLAM 技术 [J]. 机器人, 2024, 46 (4): 425-435.
- [6] LU CH X, ZHONG H, LIU W, et al. Multi-sensor fusion SLAM in complex terrain environments [J]. Robot, 2024, 46(4): 425-435.
- [7] SHAN T, ENGLOT B. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain [C]. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018: 4758-4765.
- [8] SHAN T, ENGLOT 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), 2020: 5135-5142.
- [9] XU W, ZHANG F. FAST-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated Kalman filter [J]. IEEE Robotics and Automation Letters, 2021, 6(2): 3317-3324.
- [10] 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.
- [11] YUAN Z, LANG F, XU T, et al. LIW-OAM: LiDAR-inertial wheel odometry and mapping [C]. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2023: 1481-1488.
- [12] 王潇. 动态环境下无人车激光 SLAM 点云地图优化算法研究 [D]. 哈尔滨: 哈尔滨理工大学, 2024.
- [13] WANG X. Research on optimization algorithm of laser slam point cloud map for unmanned vehicles in dynamic environment [D]. Harbin: Harbin University of Science and Technology, 2024.

- and Technology, 2024.
- [11] 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.
- [12] PENG H, ZHAO Z, WANG L. A review of dynamic object filtering in slam based on 3D lidar [J]. Sensors, 2024, 24(2): 645.
- [13] 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.
- [14] SCHAUER J, NÜCHTER A. The people remover—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.
- [15] 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 (IRS), 2020: 10758-10765.
- [16] ZHANG C, CHEN M, WANG G, et al. LIWOM-GD: Enhanced lidar-inertial-wheel odometry and mapping by fusion with ground constraint and dynamic points elimination [J]. IEEE Sensors Journal, 2024, 24(19): 30287-30303.
- [17] 郑川川,柯福阳,汤琴琴. 基于图优化的激光惯导紧耦合 SLAM 研究 [J]. 电子测量技术, 2023, 46(1): 35-42.
- ZHENG CH CH, KE F Y, TANG Q Q. Tightly coupled SLAM for laser inertial navigation based on graph optimization [J]. Electronic Measurement Technology, 2023, 46(1): 35-42.
- [18] 张喜,张鹏,刘鹏. 基于垂直约束的紧耦合激光惯性 SLAM 方法 [J]. 国外电子测量技术, 2024, 43(9): 1-7.
- ZHANG X, ZHANG P, LIU P. Tightly coupled LiDAR-inertial SLAM based on vertical constraint [J]. Foreign Electronic Measurement Technology, 2024, 43(9): 1-7.
- [19] LEE S, LIM H, MYUNG H. Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud [C]. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022: 13276-13283.
- [20] FALQUE R, GENTIL C L, SUKKAR F. Dynamic object detection in range data using spatiotemporal normals [J]. ArXiv preprint arXiv: 2310. 13273, 2023.
- [21] KAESZ M, JOHANSSON H, ROBERTS R, et al. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering [C]. 2011 IEEE International Conference on Robotics and Automation, 2011: 3281-3288.

作者简介



吴永豪,2023 年于天津科技大学获得学士学位,现为南京理工大学硕士研究生,主要研究方向为多传感器融合与激光雷达 SLAM。

E-mail: 2452046757@qq.com

Wu Yonghao received his B. Sc. degree from Tianjin University of Science and Technology in 2023. Now he is a M. Sc. candidate in Nanjing University of Science and Technology. His main research interests include multi-sensor fusion and lidar SLAM.



李胜(通信作者),2006 年于南京理工大学获得博士学位,现为南京理工大学教授,主要研究方向为机器人控制、SLAM 和多智能体系统。

E-mail: lisheng@njust.edu.cn

Li Sheng (Corresponding author) received his Ph. D. degree from Nanjing University of Science and Technology in 2006. Now he is a professor in Nanjing University of Science and Technology. His main research interests include robot control, SLAM, and multi-agent system.



邹文成,2020 年毕业于南京理工大学,获博士学位。现为南京理工大学教授,主要研究方向为多智能体系统协同、非线性系统控制。

E-mail: gongchan00@163.com

Zou Wencheng received his Ph. D. degree from Nanjing University of Science and Technology in 2020. He is currently a professor at Nanjing University of Science and Technology, with research interests including multi-agent system cooperation and nonlinear system control.