

面向自动泊车的可变滑动窗口环视 SLAM 算法^{*}

张炳力^{1,2} 米向东^{1,2} 郑达^{1,2} 黄浩然^{1,2} 张梓迟¹

(1. 合肥工业大学汽车与交通工程学院 合肥 230009; 2. 安徽省智能汽车工程研究中心 合肥 230009)

摘要: 针对车辆在自动泊车场景中遇到减速带或坑洼地带导致的地图偏移问题,以及动态环境多变导致的系统实时性变差、精度降低的问题,提出了一种面向自动泊车的可变滑动窗口环视 SLAM 算法。首先,使用 IMU 对采集的环视图像进行实时姿态校准,提升建图的准确性;其次,结合多传感器融合优势,融合 IMU 和里程计数据对车辆位姿进行估计;最后,通过可变滑动窗口算法加速后端优化,提升系统的实时性和精度。仿真测试结果表明,该方法解决了减速带或坑洼地带的地图偏移问题,且在特征稀疏环境下,效率和实时性分别提升了 31.35% 和 25.06%。实车试验结果表明,该方法可以达到平均误差 0.039 m 的定位精度,为泊车提供了安全保障。

关键词: 自动泊车;同步定位与建图;可变滑动窗口;多传感器融合

中图分类号: U471.15; TN99 文献标识码: A 国家标准学科分类代码: 580.2010; 510.1050

Variable sliding window panoramic SLAM algorithm for automatic parking

Zhang Bingli^{1,2} Mi Xiangdong^{1,2} Zheng Da^{1,2} Huang Haoran^{1,2} Zhang Zichi¹

(1. School of Automobile and Traffic Engineering, Hefei University of Technology, Hefei 230009, China;

2. Anhui Intelligent Vehicle Engineering Research Center, Hefei 230009, China)

Abstract: Aiming at the problem of map offset caused by deceleration zone or pothole zone encountered by vehicles in automatic parking scenarios, and the problem of poor real-time performance and low accuracy of the system caused by changing dynamic environment, a variable sliding window look-around SLAM algorithm for automatic parking is proposed. Firstly, IMU is used to calibrate the real-time attitude of the collected panoramic image to improve the accuracy of mapping. Secondly, combined with the advantages of multi-sensor fusion, the IMU and odometer data are fused to estimate the vehicle pose. Finally, the variable sliding window algorithm is used to accelerate the back-end optimization and improve the real-time performance and accuracy of the system. The simulation test results show that the method solves the problem of map offset in the deceleration zone or the pit zone, and the efficiency and real-time performance are improved by 31.35% and 25.06% respectively in the sparse feature environment. The real vehicle test results show that the method can achieve the positioning accuracy with an average error of 0.039 m, which provides safety guarantee for parking.

Keywords: automatic parking; synchronous positioning and mapping; variable sliding window; multi-sensor fusion

0 引言

近年来,智能汽车成为全球汽车产业的主要发展方向,对于推动我国汽车产业高质量发展,提升综合国力具有重要战略意义^[1]。在智能汽车基础上,自动驾驶技术通过结合人工智能算法和传感器设备使汽车实现自主感知、决策和执行^[2]。自动泊车作为自动驾驶技术中的重要部分,有利于保障行车安全,受到越来越广泛的研究。自动泊车需

要控制车辆自动驶入停车场并停到目标车位。在这个过程中,车辆需要自行构建环境地图并进行定位,由于停车场通常比较狭窄,因此精准的同步定位与建图(simultaneous localization and mapping, SLAM)至关重要。

目前,SLAM 技术向多传感器融合的方向发展,传感器包括激光雷达、视觉、惯性测量单元(inertial measurement unit, IMU)、里程计和全球定位系统(global positioning system, GPS)等。文献[3]提出了一种融合激光雷达和惯性/

高度传感器数据的解耦算法,能够在高程退化情况下保持定位精度。文献[4]将激光里程计与 GPS 等传感器信息进行融合,在不增加资源负担的情况下进行室外大范围建图。但采用高成本的激光雷达和惯性导航等设备,不利于产业化量产。文献[5]通过融合基于图优化的稀疏直接法以及特征点进行位姿求解,解决低纹理环境下定位精度降低以及鲁棒性差的问题。文献[6]提出了一种单目视觉里程计的方法,通过多残差项对相机位姿进行优化,使系统具有跟踪稳定性。但仅依赖视觉信息容易受到特征稀疏环境的影响。

在自动泊车场景中,文献[7]采取视觉方式,通过对场景要素聚类维分析,得到提取的车位类型和特征。文献[8]通过利用超声波雷达、相机以及里程计,融合视觉以及距离信息,实现了对常见车位的识别。文献[9]提出了适合自动泊车环境的关键帧选择策略,通过计算图像相关性来进行匹配,解决了尺度漂移问题。但当车辆遇到减速带或坑洼地带时,以上文献方法无法根据车辆姿态对获取的环境信息进行修正,降低了系统准确性。文献[10]开发了一种视觉 SLAM 系统,通过提取语义特征信息来建立停车场地图,同时对车辆进行定位。文献[11]提出了一种基于粒子滤波和语义匹配描述子的 SLAM 方法,可在地下停车场进行实时定位。文献[12]在环视图像中提取空间轮廓进行位姿跟随,生成虚拟的点云信息。文献[13]基于语义信息,使用图匹配的方法进行帧间匹配,提高了回环检测的准确性。但在系统运行中,以上方法设定的特定参数不会随

动态环境进行实时变化,导致算力分配不均,影响系统的实时性和精度。

针对上述问题,本文的创新工作如下:1)采用成本较低的鱼眼相机、IMU 和里程计实现多传感器融合感知,有利于产业化量产和提升系统鲁棒性;2)设计了根据车辆 IMU 传感器数据进行实时姿态校准的算法,有利于提升车辆在经过减速带和坑洼地带时建图的准确性;3)设计了可根据车辆姿态和周围环境实时改变窗口大小的可变滑动窗口算法,与传统固定宽度的滑动窗口算法相比,有利于充分利用算力,提升系统的实时性。

1 环视 SLAM 系统设计

1.1 系统概述

本文提出的面向自动泊车的可变滑动窗口环视 SLAM 算法主要由以下 3 部分构成:1)感知信息采集部分:使用四颗鱼眼相机、IMU 和里程计进行多传感器融合感知;2)前端信息处理部分:通过 Deeplab v3+ 模型对环视图像进行语义分割,得到车道线信息,对车道线信息进行点云提取,并根据 IMU 数据对提取的点云进行实时姿态校准。通过扩展卡尔曼滤波对 IMU 和里程计信息进行融合,估计初始位姿。结合车辆的初始位姿进行扫描匹配,进而生成局部地图;3)后端优化部分:使用可变滑动窗口算法加速闭环检测和相对位姿的计算过程,实现对地图和位姿的优化。系统整体架构如图 1 所示。

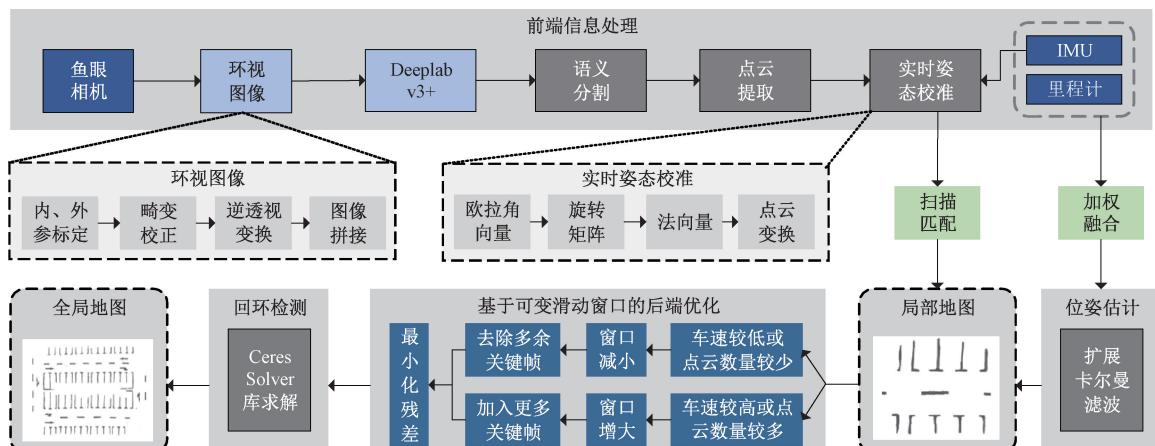


图 1 系统整体架构流程图

Fig. 1 The overall system architecture flow chart

1.2 感知信息采集

通过鱼眼相机获取当前车辆四周的环视图像,首先,对环视相机进行内、外参标定并对相机畸变进行校正;其次,采用逆透视变换将畸变校正后的图像转换为俯视图,并对四颗鱼眼相机的俯视图进行拼接,进而得到车辆周围的环视图像;最后,通过里程计获取当前车辆的行驶里程信息,通过 IMU 获取车辆的加速度和角度信息。

1.3 前端信息处理

1) 语义分割

利用全卷积神经网络^[14]进行语义分割,使用 Deeplab v3+ 模型对环视图像中的车道线进行像素级分割,得到分割后的车道线,网络架构如图 2 所示。

2) 点云提取

对语义分割后的图像进行点云提取,如图 3 所示。首先对图像像素点进行逐行逐列遍历,提取像素点的颜色信

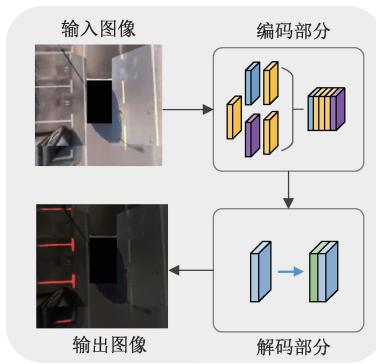


图 2 DeepLab v3+ 网络架构图

Fig. 2 DeepLab v3+ network architecture diagram

息,并根据设定的颜色阈值对其进行筛选,将无效区域删除。接着通过相机坐标系的逆变换,得到相机坐标系下的坐标值。

$$\mathbf{p}_1 = \mathbf{R}_w^{-1} \mathbf{R}_n^{-1} \mathbf{p} \quad (1)$$

式中: \mathbf{p}_1 是像素在相机坐标系下的三维坐标, \mathbf{R}_w 是相机的外参矩阵, \mathbf{R}_n 是相机的内参矩阵, \mathbf{p} 是像素的二维坐标。

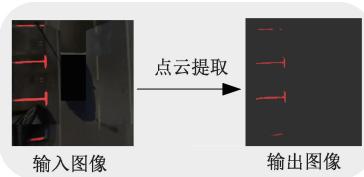


图 3 点云提取示意图

Fig. 3 Point cloud extraction schematic diagram

随后,对相机坐标系下的坐标进行归一化处理,计算点到相机光心的距离,对距离进行筛选,排除掉距离过远的点。

3) 实时姿态校准

本文提出了根据车辆 IMU 传感器对提取的点云进行实时姿态校准的方法,如图 4 所示。通过 IMU 获取车辆的横摆角 θ_{yaw} 、俯仰角 θ_{pitch} 和侧倾角 θ_{roll} ,进而可得欧拉角向量为 $\mathbf{M}_{euler} = (\theta_{yaw}, \theta_{pitch}, \theta_{roll})$,将欧拉角向量转换为旋转矩阵:

$$\mathbf{R}_{imu} = \mathbf{R}_{yaw} \cdot \mathbf{R}_{pitch} \cdot \mathbf{R}_{roll} \quad (2)$$

式中: \mathbf{R}_{imu} 是 IMU 的旋转矩阵, \mathbf{R}_{yaw} 、 \mathbf{R}_{pitch} 、 \mathbf{R}_{roll} 分别是欧拉角向量绕 Z 轴、Y 轴、X 轴的旋转矩阵。

取旋转矩阵 \mathbf{R}_{imu} 第 3 列作为 IMU 平面的法向量

\mathbf{M}_{normal} ,记为:

$$\mathbf{M}_{normal} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (3)$$

系统坐标系中的法向量定义为 $\mathbf{V}_s = (0, 0, 1)^T$,利用式(4)计算 IMU 平面的法向量 \mathbf{M}_{normal} 到 \mathbf{V}_s 的旋转矩阵:

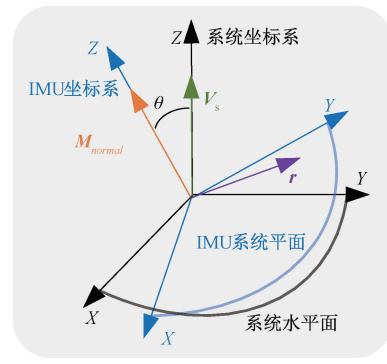


图 4 向量旋转示意图

Fig. 4 Vector rotation diagram

$$\left\{ \begin{array}{l} \theta = \cos^{-1}(\mathbf{V}_s \cdot \mathbf{M}_{normal}) \\ \mathbf{r} = \frac{\mathbf{V}_s \times \mathbf{M}_{normal}}{\|\mathbf{V}_s \times \mathbf{M}_{normal}\|} \\ \mathbf{C} = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \\ \mathbf{R} = \mathbf{I} + \mathbf{C} \sin \theta + \mathbf{C}^2 (1 - \cos \theta) \end{array} \right. \quad (4)$$

式中: θ 为旋转角度, \mathbf{r} 为旋转轴, \mathbf{C} 为旋转轴 \mathbf{r} 的叉乘矩阵, r_x, r_y, r_z 分别为旋转轴 \mathbf{r} 的 X 轴、Y 轴、Z 轴分量, \mathbf{R} 为 \mathbf{M}_{normal} 到 \mathbf{V}_s 的旋转矩阵, \mathbf{I} 为单位矩阵。

对提取的点云使用式(5)进行姿态校准,点云中的每个点 (x, y, z) 被旋转矩阵 \mathbf{R} 变换后的新坐标为 (x', y', z') 。

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (5)$$

4) 位姿估计

设车辆在系统坐标系下的位姿为:

$$\mathbf{x} = [x_v, y_v, \theta_{yaw}] \quad (6)$$

式中: \mathbf{x} 为车辆系统坐标系下的位姿, x_v 和 y_v 分别为车辆的 X 轴和 Y 轴坐标, θ_{yaw} 为车辆横摆角。

融合 IMU 和里程计数据,利用二者之间的互补信息进行初始位姿估计。为提高位姿估计的精度,根据车辆不同的行驶状况对 IMU 和里程计数据分配不同的权重,在车辆转弯时,分配 IMU 数据较大的权重。在车辆直行时,分配里程计数据较大的权重。采用扩展卡尔曼滤波方法对二者进行加权融合,分为预测步骤和更新步骤。

预测步骤:

$$\hat{\mathbf{x}}_t = f(\hat{\mathbf{x}}_{t-1}, \alpha, \omega) \quad (7)$$

式中: $\hat{\mathbf{x}}_t$ 为 t 时刻预测状态, $\hat{\mathbf{x}}_{t-1}$ 为 $t-1$ 时刻预测状态, α 和 ω 分别为 IMU 输出的加速度和角速度。

更新步骤:

$$\hat{\mathbf{x}}'_t = \hat{\mathbf{x}}_t + \mathbf{K}(\mathbf{n}_t - h(\hat{\mathbf{x}}_t)) \quad (8)$$

式中: $\hat{\mathbf{x}}'_t$ 为 t 时刻更新状态, \mathbf{K} 为卡尔曼增益矩阵, \mathbf{n}_t 为 t 时刻测量向量, $h(\cdot)$ 为观测模型。

5) 扫描匹配

当每帧点云插入地图前, 将点云和地图进行扫描匹配, 对此刻车辆位姿进行优化。扫描匹配的目的是搜寻到一个最佳的位姿, 使当前点云中的点最大概率地与地图匹配。通过非线性最小二乘法来解决此优化问题:

$$\arg \min_{\mathbf{x}} \sum_{n=1}^N (1 - M_{\text{smooth}}(\mathbf{T}_b \mathbf{p}_n))^2 \quad (9)$$

式中: N 为当前帧点云的数量, $M_{\text{smooth}}(\cdot)$ 为平滑性函数, \mathbf{T}_b 为相机坐标系相对于系统坐标系的变换矩阵, \mathbf{p}_n 为当前帧点云的位置坐标。 $M_{\text{smooth}}(\cdot)$ 函数使用双三次插值法, 将点以平滑的概率值形式映射到地图中, 并衡量平滑误差(1—平滑度), 计算每个点误差的平方和, 得到总体的平方误差。使用 Ceres Solver 库最小化总体平方误差, 得到最优的车辆位姿, 在此位姿下添加当前帧点云, 更新局部地图。

1.4 后端优化

在每帧点云与地图匹配的过程中, 随着时间变长或场景变大, 会产生累计误差, 过量的累计误差会导致建图失败。针对上述问题, 使用后端全局优化可以有效修正车辆位姿的偏差, 进而降低累计误差使地图具有较高的精度。当前常用方法是使用基于滑动窗口非线性优化方法对位姿进行优化, 滑动窗口中的状态变量如式(10)所示。

$$\begin{cases} \chi = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \lambda_1, \lambda_2, \dots, \lambda_M] \\ \mathbf{x}_t = [v_{v_t}^w, \mathbf{p}_{v_t}^w, \mathbf{q}_{v_t}^w, c_a, c_g], t \in [1, N] \end{cases} \quad (10)$$

式中: χ 为滑动窗口状态变量, \mathbf{x}_t 为 t 时刻相机在窗口中的状态, λ_i 为第 i 个视觉特征点的逆深度值, $v_{v_t}^w$ 为车辆的速度, $\mathbf{p}_{v_t}^w$ 为世界坐标系下车辆的位置, $\mathbf{q}_{v_t}^w$ 为车辆的方向信息, c_a 为 IMU 中加速度计的零位偏移, c_g 为 IMU 中陀螺仪的零位偏移。

通过相加不同类型的残差项, 最小化残差项的范数之和, 得到后端优化的代价函数为:

$$\min_{\chi} \left\{ \begin{array}{l} \|\mathbf{r}_p - \mathbf{H}_p \chi\|^2 + \sum_{i \in A} \|\mathbf{r}_A(\hat{\mathbf{z}}_{a_i}^{a_{i-1}}, \chi)\|_{P_{a_i}^{a_{i-1}}}^2 + \\ \sum_{(l, j) \in B} \|\mathbf{r}_B(\hat{\mathbf{z}}_l^{b_j}, \chi)\|_{P_l^{b_j}}^2 \end{array} \right\} \quad (11)$$

式中: \mathbf{r}_p 为边缘化产生的先验残差, \mathbf{H}_p 为边缘化产生的先验矩阵, A 为滑动窗口中 IMU 预积分的集合, $\mathbf{r}_A(\hat{\mathbf{z}}_{a_i}^{a_{i-1}}, \chi)$ 为 IMU 测量的残差项, $\hat{\mathbf{z}}_{a_i}^{a_{i-1}}$ 为相邻两关键帧测量信息, $P_{a_i}^{a_{i-1}}$ 为预积分的协方差矩阵, B 为视觉关键帧的集合, $\mathbf{r}_B(\hat{\mathbf{z}}_l^{b_j}, \chi)$ 为视觉测量的残差项, $\hat{\mathbf{z}}_l^{b_j}$ 为地图点 l 与第 j 帧图像之间的视觉测量信息, $P_l^{b_j}$ 为视觉测量的协方差矩阵。

然而, 固定大小的滑动窗口存在多方面的弊端, 包括计算效率低下、精度和鲁棒性降低、动态环境适应性差以及优化难度增加。固定窗口在简单环境中会进行大量冗

余计算, 浪费计算资源和存储空间。在复杂环境中则可能无法涵盖足够信息, 导致局部地图精度不足和回环检测困难, 影响全局一致性和鲁棒性。此外, 固定窗口无法适应车辆速度和点云密度的变化, 在速度较快或点云密度高时可能遗漏重要信息, 在速度较慢或点云密度低时处理过多无用信息, 降低效率。

本文针对上述问题进行改进, 提出了可变滑动窗口算法, 如图 5 所示, 用以提高回环检测和相对位姿的求解效率。通过式(12)实现基于车速和点云数量的动态调整: 窗口调整系数由车速和点云数量共同决定, 并引入权重系数用以平衡车速和点云数量的影响; 滑动窗口大小决定窗口内包含的关键帧数量和覆盖区域的大小, 将地图分辨率作为滑动窗口基础大小; 角度搜索步长决定搜索效率, 将点云中距离原点最远点的距离作为斜边和临边, 将地图分辨率作为对边, 根据余弦定理得到角度搜索步长基础大小。

$$\left\{ \begin{array}{l} \mu = \alpha \left(\beta \cdot \frac{v}{v_{\max}} + (1 - \beta) \cdot \frac{S}{S_{\max}} \right) \\ \delta = \mu \cos^{-1} \left(1 - \frac{m^2}{2 \left(\max_{n=1,2,\dots,N} \mathbf{p}_n \right)^2} \right) \\ w = \mu m \end{array} \right. \quad (12)$$

式中: μ 为窗口调整系数, α 为比例系数, β 为权重系数, v 为当前时刻车速, v_{\max} 为设定的车速最大值, S 为当前帧点云数量, S_{\max} 为所有帧点云数量最大值, δ 为滑动窗口角度搜索步长, m 为地图分辨率, w 为滑动窗口大小。

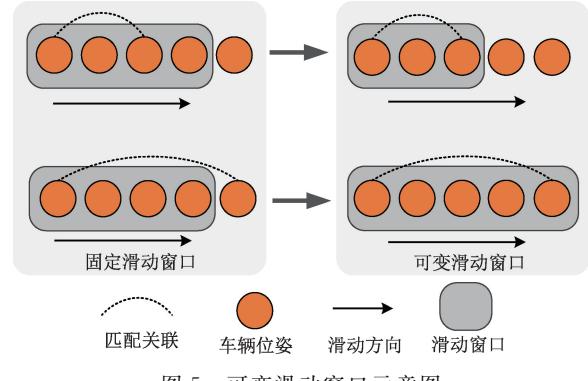


图 5 可变滑动窗口示意图

Fig. 5 Variable sliding window diagram

在车速较低或者点云数量较少时, 调整为较小的滑动窗口, 对原窗口内的关键帧进行分析, 将匹配程度小于阈值的判定为多余关键帧, 使用边缘化操作去除掉多余关键帧。重复上述操作, 直到关键帧数量符合窗口大小为止, 以减少计算量并提高匹配的精度和速度; 在车速较快或者点云数量较多时, 调整为较大的滑动窗口, 首先, 对原窗口内的关键帧进行分析, 将匹配程度大于阈值的关键帧标记为重要关键帧, 并在滑动窗口中保留重要关键帧。然后, 基于重要关键帧增大滑动窗口, 加入更多的关键帧以覆盖更大的区域。对新加入的关键帧进行匹配分析, 判断其与重要关键

帧的匹配程度,将匹配程度大于阈值的关键帧保留在滑动窗口中。重复上述操作,直至滑动窗口达到设定的大小,以保证能够充分覆盖环境,提高检测的精度和鲁棒性。

在使用滑动窗口搜索过程中,使用 Ceres Solver 库求解当前帧点云与局部地图的匹配程度。若匹配程度达到阈值,则检测到回环的存在,生成回环约束,优化车辆位姿和运动轨迹,对全局地图进行更新。

2 实验验证

在仿真软件以及自主搭建的实车试验平台上进行测试。基于公开数据集 PS2.0^[15]使用 Labelme 软件标注了 1 850 张包含停车位轮廓的图像,使用其对 Deeplab v3+语义分割网络模型进行训练,得到训练完的模型用于后续语

义分割处理。

2.1 仿真实验

基于 Ubuntu18.04 系统和机器人操作系统,试验条件为配备 Intel i5-9500T 处理器,8 GB 运行内存的工业控制计算机。在 Gazebo 软件进行仿真,分别搭建特征稠密仿真环境和特征稀疏仿真环境,如图 6(a)、(f)所示。在特征稠密仿真环境中,设置 120 个停车位。在特征稀疏仿真环境中,设置 36 个停车位。仿真车辆运动的线速度设为 3 m/s,角速度设为 0.5 rad/s,比例系数 α 设为 1,权重系数 β 设为 0.5。同时将改进前无实时姿态校准和使用固定滑动窗口的算法与改进后算法进行对比。之后在仿真环境中加入减速带,以验证改进后算法对车辆遇到减速带或坑洼地带时的优化效果。

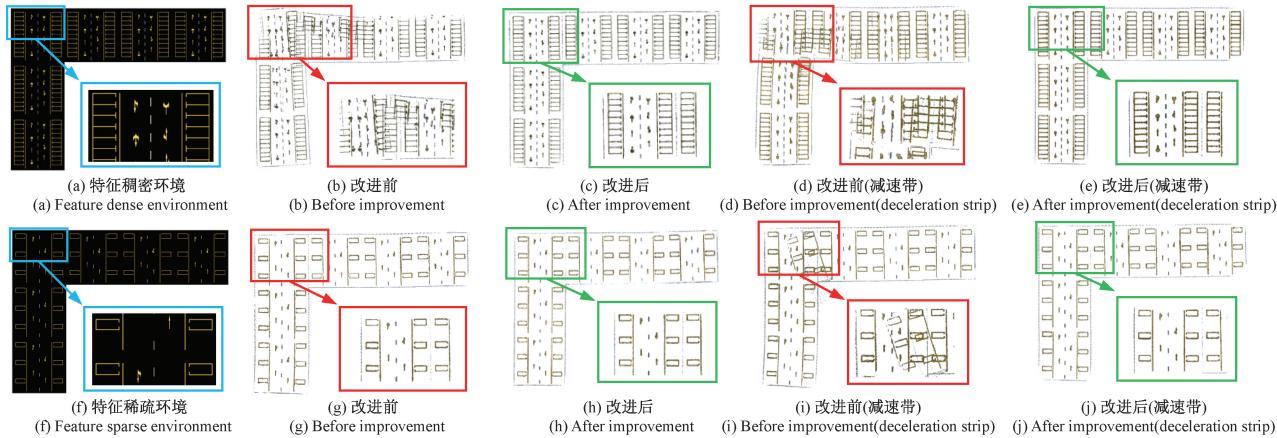


图 6 仿真环境及算法效果图

Fig. 6 Simulation environment and algorithm effect diagram

对算法的效率与实时性进行分析,统计结果如表 1 所示。在特征稠密仿真环境下,改进前算法占用内存 2.71 GB,改进后算法占用内存 1.90 GB,在效率上提升了 29.89%。改进前算法总计用时 61.95 ms,改进后算法总计用时 53.55 ms,在实时性上提高了 13.56%;在特征稀疏仿真环境下,改进前算法占用内存 1.85 GB,

改进后算法占用内存 1.27 GB,在效率上提升了 31.35%。改进前算法总计用时 53.47 ms,改进后算法总计用时 40.07 ms,在实时性上提高了 25.06%。由结果可知,在特征稀疏环境下,改进后算法节约了更多的计算资源,相较于特征稠密环境下提升的实时性而言,提升效果更为显著。

表 1 算法效率与实时性对比

Table 1 Comparison of algorithm efficiency and real-time performance

仿真环境	平均每帧 点云个数	实时姿态 校准	可变滑动 窗口	内存/ GB	前端用时/ ms	后端用时/ ms	总计用时/ ms
特征稠密环境	13 712	√	√	2.71	20.58	41.37	61.95
			√	2.79	23.74	40.25	63.97
			√	1.83	20.92	29.23	50.15
特征稀疏环境	4 319	√	√	1.90	22.79	30.76	53.55
			√	1.85	15.44	38.03	53.47
			√	1.89	17.81	37.16	54.97
		√	√	1.25	15.69	21.92	37.61
			√	1.27	17.09	23.07	40.16

对算法的建图效果进行分析,算法改进前、后建图效果如图 6 所示。在特征稠密环境下,如图 6(b)所示,改进前算法在处理点云数量较多的情况下,发生了地图偏移,不能构成回环,影响了全局的一致性和鲁棒性。改进后算法针对此情况进行了改善,如图 6(c)所示,形成了回环约束,保证了建图准确性。在特征稀疏环境下,如图 6(g)、(h)所示,算法建立的地图完整,具有较高的建图准确性。

对车辆遇到减速带时的情况进行分析,算法改进前、后处理效果如图 6 所示。在特征稠密、稀疏环境下,如图 6(d)、(i)中的红框所示,改进前算法在车辆经过减速带时,构建的停车场地图均产生了偏移,产生的错误信息无法用于后续泊车。改进后算法针对此问题进行了优化,如图 6(e)、(j)中的绿框所示,有利于根据车身姿态实时进行纠正,保持了建图的准确性。

2.2 实车试验

实车试验平台为改装的江淮 iev6s 纯电动汽车。如图 7 所示,包括 4 颗鱼眼相机、1 个 IMU 传感器、2 个轮式里程计和 1 套为实验提供真值轨迹的高精度惯性组合导航系统。



图 7 江淮 iev6s 纯电动汽车实验平台

Fig. 7 JAC iev6s pure electric vehicle experimental platform

在室外停车场进行实车试验,车速控制在 10 km/h 以内,在里程约 160 m 的区域进行了验证。构建的地图如图 8 所示,从图中可以看出,建立的停车场地图边界完整,无地图漂移问题。

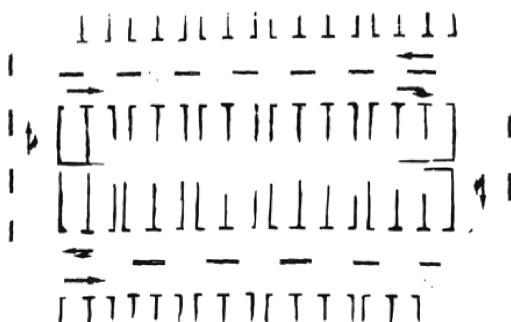


图 8 构建的停车场地图

Fig. 8 Constructed parking lot map

选取 ORB-SLAM3^[16] 算法进行对比,本文算法构建的行驶轨迹、ORB-SLAM3 算法构建的行驶轨迹以及高精度惯性组合导航系统采集的真值轨迹对比如图 9 所示。从图中可以看出,本文算法轨迹更加贴合真值,产生的误差较小。

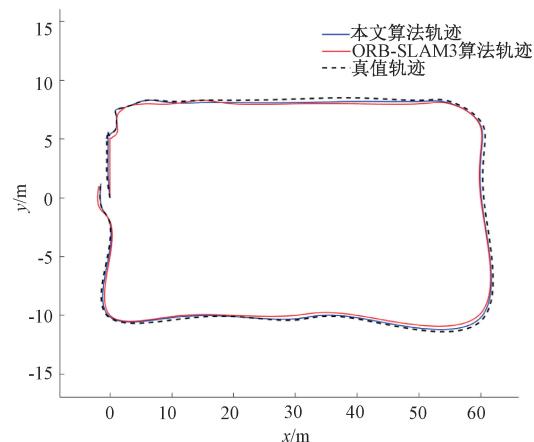


图 9 算法轨迹对比图

Fig. 9 Algorithm trajectory comparison diagram

对本文算法的定位误差进行统计,统计结果如表 2 所示。本文算法在均方根误差、最大误差和平均误差上均优于 ORB-SLAM3。本文算法的均方根误差为 0.073 m, 明显小于 ORB-SLAM3 的 0.094 m; 本文方法的最大误差为 0.153 m, 同样低于 ORB-SLAM3 的 0.190 m; 在平均误差方面,本文算法为 0.039 m, 相较于 ORB-SLAM3 的 0.053 m, 提升了 26.42%。由此可知,本文算法在定位精度上具有优势。

表 2 定位误差对比

Table 2 Positioning error comparison

算法	均方根	最大误差/	平均误差/
	误差/m	m	m
ORB-SLAM3	0.094	0.190	0.053
本文算法	0.073	0.153	0.039

对位姿误差进行统计,统计结果如表 3 所示。本文算法在平移误差和旋转误差上均表现出更高的精度。在平移误差方面,本文算法在 x、y、z 三个方向上的误差分别为 0.033 m、0.040 m 和 0.029 m, 均小于 ORB-SLAM3 的 0.047 m、0.043 m 和 0.031 m。在旋转误差方面,本文算法在 yaw、pitch 和 roll 三个方向上的误差分别为 2.63°、1.37° 和 2.75°, 均优于 ORB-SLAM3 的 2.98°、1.54° 和 3.19°。由此可见,本文算法在位姿估计的精度上也优于 ORB-SLAM3。

表 3 位姿误差对比

Table 3 Pose error comparison

算法	平移误差/m			旋转误差/(°)		
	x	y	z	yaw	pitch	roll
ORB-SLAM3	0.047	0.043	0.031	2.98	1.54	3.19
本文算法	0.033	0.040	0.029	2.63	1.37	2.75

3 结 论

本文提出了一种面向自动泊车的可变滑动窗口环视 SLAM 算法,通过融合了包括环视鱼眼相机、IMU 传感器和里程计在内的多源传感器数据,使用 IMU 对环视图像进行了实时姿态校准,通过设计的可变滑动窗口算法加速后端优化进程。仿真测试结果表明,该方法在效率和实时性上分别提升了 31.35% 和 25.06%。通过自主搭建实车试验,并与主流视觉算法 ORB-SLAM3 对比证明,本文算法在定位精度的平均误差方面提升了 26.42%,实验结果验证了本文算法在停车场自动泊车任务中的有效性。未来工作将更关注于多层室内停车场地图的构建,实现更加广泛的适用能力。

参考文献

- [1] 彭湃,耿可可,王子威,等.智能汽车环境感知方法综述[J].机械工程学报,2023,59(20):281-303.
PENG P, GENG K K, WANG Z W, et al. A review of intelligent vehicle environment perception methods [J]. Journal of Mechanical Engineering, 2023, 59 (20): 281-303.
- [2] 沈甜雨,李志伟,范丽丽,等.具身智能驾驶:概念,方法现状与展望[J].智能科学与技术学报,2024, 6(1): 17-32.
SHEN T Y, LI ZH W, FAN L L, et al. Embodied intelligent driving: Concepts, methods and prospects [J]. Chinese Journal of Intelligent Science and Technology, 2024, 6(1): 17-32.
- [3] 付林,郑佳楠,何洪磊,等.面向大型室内场景的无人机三维激光雷达解耦 SLAM 方法[J].电子测量技术,2022,45(13):96-103.
FU L, ZHENG J N, HE H L, et al. Decoupled SLAM method for UAV 3D laser radar in large indoor scenes [J]. Electronic Measurement Technology, 2022, 45(13):96-103.
- [4] 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. Multi-sensor fusion mobile robot outdoor laser SLAM algorithm optimization and system implementation [J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(2):48-55.
- [5] 叶俊强,尤睿,于明鑫,等.基于稀疏直接法的特征增强视觉 SLAM 算法[J].仪器仪表学报,2023,44(6):205-212.
YE J Q, YOU R, YU M X, et al. Feature-enhanced visual SLAM algorithm based on sparse direct method [J]. Chinese Journal of Scientific Instrument, 2023, 44(6): 205-212.
- [6] 谢波,张国良,李歆,等.一种单目 VIO 定位精度与跟踪稳定性优化方法[J].国外电子测量技术,2023, 42(4):23-30.
XIE B, ZHANG G L, LI X, et al. A monocular VIO positioning accuracy and tracking stability optimization method [J]. Foreign Electronic Measurement Technology, 2023, 42(4):23-30.
- [7] 张胜根,张强,杨良义,等.自动泊车系统测试场景的提取与转换方法研究[J].重庆理工大学学报(自然科学),2022,36(2):68-75.
ZHANG SH G, ZHANG Q, YANG L Y, et al. Research on the extraction and conversion method of automatic parking system test scenarios [J]. Journal of Chongqing University of Technology (Natural Science), 2022, 36(2):68-75.
- [8] 姜武华,辛鑫,陈无畏,等.基于信息融合的自动泊车系统多工况车位识别和决策规划[J].机械工程学报,2021,57(6):131-141.
JIANG W H, XIN X, CHEN W W, et al. Multi-condition parking space identification and decision-making planning of automatic parking system based on information fusion [J]. Journal of Mechanical Engineering, 2021, 57(6):131-141.
- [9] 林泽.面向自动泊车环境的视觉 SLAM 技术研究[D].长春:吉林大学,2020.
LIN Z. Research on visual SLAM technology for automatic parking environment [D]. Changchun: Jilin University, 2020.
- [10] 秦涛,陈涛,陈亚玲,等. AVP-SLAM: Semantic visual mapping and localization for autonomous vehicles in the parking lot [C]. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020:5939-5945.
- [11] 侯江霞,杨明,徐海泉,等.利用语义道路标记进行厘米级室内停车场定位与建图[C].2019 IEEE Intelligent Transportation Systems Conference, 2019:4068-4073.
- [12] 常志,余江,李江,等.基于多摄像头系统的 LiDAR-视觉里程计算法[J].2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019:2486-2492.
- [13] 周苏,卓家璇,尹丁,等.用于停车场视觉建图的语义回环检测算法[J].中国科技论文,2023, 18 (11): 1186-1191.

- ZHOU S, ZHUO J X, YIN D, et al. Semantic loop closure detection algorithm for parking lot visual mapping [J]. China Science and Technology Paper, 2023, 18(11): 1186-1191.
- [14] SHELHAMER E, LONG J, DARRELL T. Fully convolutional networks for semantic segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640-651.
- [15] ZHANG L, HUANG J H, LI X Y, et al. Vision-based park-ing-slot detection: A DCNN-based approach and a large-scale benchmark dataset [J]. IEEE Transactions on Image Processing, 2018, 27(11): 5350-5364.
- [16] CAMPOS C, ELVIRA R, RODRIGUEZ J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual inertial, and multimap SLAM [J]. IEEE

Transactions on Robotics, 2021, 37(6): 1874-1890.

作者简介

张炳力,博士,教授,主要研究方向为智能网联新能源汽车技术及产业发展。

E-mail: zhangbingli@hfut.edu.cn

米向东(通信作者),硕士研究生,主要研究方向为无人驾驶 SLAM 技术。

E-mail: mixiangdong66@163.com

郑达,博士研究生,主要研究方向为无人驾驶与自动泊车技术。

E-mail: zd656691@163.com

黄浩然,硕士研究生,主要研究方向为无人驾驶环境感知技术。

E-mail: 2607898856@qq.com

张梓迟,本科生,主要研究方向为智能汽车技术。

E-mail: 2543974175@qq.com