

基于 ICP 算法的移动机器人激光定位研究

赵龙云^{1,2}, 伞红军^{1,2*}, 陈久朋^{1,2}, 彭真^{1,2}¹昆明理工大学机电工程学院, 云南 昆明 650500;²云南省先进装备智能制造技术重点实验室, 云南 昆明 650500

摘要 针对移动机器人在进行传统 2D 环境的定位时所存在的定位精度低且定位实时性差等问题, 提出一种改进的迭代近邻点(ICP)算法的定位方法。首先, 建立位姿搜索空间, 采用由低到高的分辨率对搜索空间进行逐层搜索, 并结合多点云密度进行部分点云扫描匹配, 排除非最优位姿, 加速搜索过程; 在进行点云匹配中, 采用帧对图的方式, 有效地利用了历史帧信息; 对得到的最优位姿进行稀疏矩阵位姿优化, 进一步提高定位精度。在 SLAM Benchmark 数据集上进行测试, 结果表明所提方法的算法效率是现流行的 Cartographer 算法的 1.8 倍到 4.9 倍之间, 同时平移误差较小。并利用 Turtlebot2 机器人进行实际测试, 结果表明所提方法的定位误差相比 Cartographer 和 Gmapping 均有明显的降低, 且实时性较好; 与传统的自适应蒙特卡罗重定位(AMCL)相比, 平移误差均值降低了 0.035 m, 旋转误差均值降低了 0.001 rad, 具有较高的重定位精度。

关键词 激光雷达; 移动机器人定位; 多分辨率; 迭代近邻点; 多点云密度
中图分类号 TP242 文献标志码 A

DOI: 10.3788/LOP231307

Laser Location of Mobile Robot Based on ICP Algorithm

Zhao Longyun^{1,2}, San Hongjun^{1,2*}, Chen Jiupeng^{1,2}, Peng Zhen^{1,2}¹Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, Yunnan, China;²Key Laboratory of Intelligent Manufacturing Technology for Advanced Equipment in Yunnan Province, Kunming 650500, Yunnan, China

Abstract This study presents a localization method based on the improved iterative closest point (ICP) algorithm to solve the localization problems of mobile robots, such as low positioning accuracy and poor real-time positioning, in traditional 2D environments. The algorithm begins by establishing a pose search space, which systematically explored layer-by-layer, transitioning from lower to higher resolutions. To accelerate the search process and eliminate nonoptimal poses, partial point cloud scanning matching was executed synergistically with multipoint cloud density. Adoption of the frame-to-image method by the point cloud matching enabled the effective utilization of historical frame information. Further enhancements in positioning accuracy were achieved through the sparse matrix pose optimization for obtained optimal pose. Tests conducted on the SLAM Benchmark dataset show that the proposed algorithm is considerably more efficient, boasting a 1.8–4.9 times efficiency gain over the popular Cartographer algorithm, and has less translation error. Real-world tests conducted on Turtlebot2 reveal that the proposed method exhibits substantially fewer positioning errors than Cartographer and Gmapping, showing superior real-time performance. Compared with the traditional adaptive Monte Carlo relocation (AMCL), the proposed method reduces mean translation errors by 0.035 m and mean rotation errors by 0.001 rad, resulting in higher relocation accuracy.

Key words lidar; mobile robot localization; multiresolution; iterative closest point; multipoint cloud density

1 引言

移动机器人定位^[1-2]是工业 AGV 和服务型机器人研究中的重点, 同时也是自主导航和运动控制中不可或缺的一步。目前的移动机器人定位方式主要基于视觉^[3]、里程计、蓝牙^[4]、WiFi^[5]、人工标注^[6]、惯性测量仪

收稿日期: 2023-05-15; 修回日期: 2023-06-24; 录用日期: 2023-07-24; 网络首发日期: 2023-08-15

基金项目: 云南省科技厅重大专项(202002AC080001)、云南省基础研究计划项目(202301AU070059)

通信作者: *sanhongjun@163.com

(IMU)、激光^[7-9]等来确定机器人的实时位置。基于蓝牙和 WiFi 进行定位容易受到硬件的影响,同时不适用于远距离定位;基于里程计进行定位容易出现漂移,随着时间的增加会出现累计误差;基于 IMU 进行定位在存在磁场的环境中会受到限制,影响定位效果。为了解决上述问题,采用视觉或激光结合同步定位与建图 (SLAM) 技术来获取机器人的实时位置,如通过外部相机获得自身位置信息的视觉 SLAM^[10]和通过激光雷达获得外部数据的激光 SLAM^[11]。基于视觉的定位会受到传感器的限制,在相应的目标算法匹配时,机器人进行姿态变化时容易出现误匹配的情况,同时容易受到光照的影响,导致定位结果失真。因激光雷达在测量精度和抗干扰方面相对相机传感器性能较好,因此成为了 2D 环境全局定位中重要的部分。

激光 SLAM 定位经过 20 多年的发展,现在已趋于成熟。激光 SLAM 技术已成为了移动机器人定位过程中的重要部分,例如 Gmapping、Hector SLAM^[12]、Cartographer^[13]算法等。但这些算法进行定位时存在着许多不足,例如基于 Gmapping 算法的定位需要严重依赖里程计,且只能在小规模环境中进行定位,定位精度不高;利用 Cartographer 进行定位时,算法计算量较大,严重影响机器人的实时定位效果。

目前,对于移动机器人定位的方法和改进的研究,采取的主要措施和方法如下。冯佳萌等^[14]采取了一种基于自适应蒙特卡罗的定位方式,但引入了扫描匹配 (SM) 算法和离散的傅里叶变化 (DFT),虽然得到了准确的定位效果,但导致算法的实时性降低。白崇岳等^[15]利用激光 SLAM 与多传感器融合的定位方式,实现了更高精度的无人驾驶轮椅的定位,但过多传感器间的数据不能精确融合,造成了定位方法的复杂化。王子润等^[16]将自适应蒙特卡罗与传统遗传算法相结合,解决了机器人打滑后造成定位丢失和定位后恢复速度慢的问题,并将迭代近邻点 (ICP) 算法所得出的位姿作为先验信息。黄永新^[17]采用了多帧到多帧的激光 SLAM 方式,通过该方式可以精确地获得外部环境信息,但无法得出实时的机器人位置信息。上述方法大部分是对多种数据的融合,会存在数据量较大、实时性不高等问题。本文着重对激光 SLAM 中的前端扫描匹配进行优化,根据 Besl 等^[18]提出的 ICP 算法(一种较为简单易行的前端匹配算法),通过相邻两帧之间的点云匹配,来估计机器人的运动位置。但 ICP 的本质还是通过迭代获取位置信息的最优近似解,但计算量大,获得机器人的位置时机器人的状态已经发生了改变,会产生很大的累计误差。

因此,研究人员着重于对 ICP 定位方法进行改进,目前采用的主要措施如下。Akca 等^[19]采用空间划分的方法来确定最邻近搜索点的范围,但该类方法是对两相邻帧之间进行匹配的,难以用到历史帧信息。

Olson^[20]采用了不同分辨率进行搜索匹配,加速了搜索过程,但需要对所有点云进行匹配。Jost 等^[21]提出多点云密度 ICP 扫描匹配方式,对低分辨率的点云采取部分点云扫描匹配,对高分辨率点云采取全部点云扫描匹配,该方法会存在局部收敛。本文方法的创新点主要在于:在 ICP 算法处理传统帧间匹配的基础上,利用了历史帧信息,采取了当前扫描帧与子图匹配的方式;采取了基于多分辨率搜索的方式提高定位精度,分辨率由粗到精,每一层分辨率的搜索空间均在上一层搜索位姿最优值附近建立;采用不同密度的点云剔除非最优位姿,加速搜索过程;最后通过实验验证所提方法在提高机器人定位实时性和定位精度上的可行性。

2 定位方法框架

改进的 ICP 定位方法如图 1 所示。利用点云库 (PCL) 中的体素滤波器 (voxel filter) 对激光雷达获得的点云数据进行处理,进行滤波以减少离散点云;建立多分辨率位姿搜索区间,确定不同分辨率下的位姿节点,通过 ICP 算法扫描子图进行点云匹配,剔除不同分辨率下目标函数较大值所对应的候选位姿,该过程中采用多点云密度的方式,对部分点云进行扫描匹配,重复上述过程,达到所设定的最高分辨率,确定较优位姿;对得到的最优位姿进行稀疏矩阵位姿优化,最终确定机器人的位置信息。

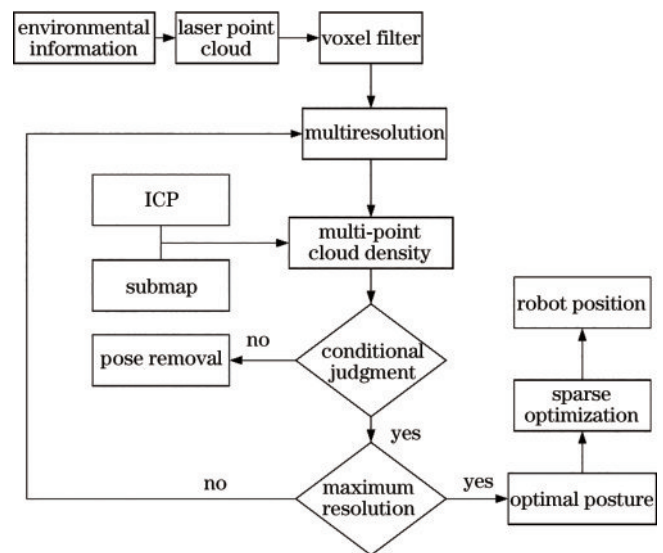


图 1 定位方法框图

Fig. 1 Block diagram of location method

3 前端匹配

在激光 SLAM 定位的过程中,常基于 ICP 算法的前端匹配来获得机器人的位置信息。通过构建欧氏距离子图,有效地利用历史帧信息;同时采用多分辨率与多点云密度结合的方式来减少计算量,确定较优位姿,获得机器人的位置信息。

3.1 多分辨率搜索

对全局位姿进行暴力匹配的方法能够得到搜索空间范围内的全局最优解,但计算量过大,效率低下。因此提出基于多分辨率搜索的方法来加速搜索过程。采用多分辨率搜索的方式,不断提高分辨率进行逐层搜索,且每次对起点位姿的搜索均在上一次搜索结果的邻近范围内进行。

图 2 为一个多分辨率搜索示意图,在进行搜索之前,首先要在位姿邻近范围内定义一个搜索空间,即为低分辨率进行搜索的最大范围,表达式为

$$\mathbf{W}_{\max} = (W_x, W_y, W_z), \quad (1)$$

式中: \mathbf{W}_{\max} 的值为在每一帧激光扫描的过程中,机器人在 x 和 y 轴运动的最大平移量和航向角度 θ ;然后定义搜索的最小步长 $\mathbf{r}_{\min} = (r_x, r_y, r_\theta)$, \mathbf{r}_{\min} 小于激光雷达传感器的最小测量误差。最后可计算出每层搜索区间下的搜索规模 $\mathbf{S} = (S_x, S_y, S_\theta)$:

$$S_x = \sqrt[N]{\frac{W_x}{r_x}}, S_y = \sqrt[N]{\frac{W_y}{r_y}}, S_\theta = \sqrt[N]{\frac{W_\theta}{r_\theta}}, \quad (2)$$

第 n 层的搜索区间表示为

$$\omega_{nx} = \frac{W_x}{S_x^{n-1}}, \omega_{ny} = \frac{W_y}{S_y^{n-1}}, \omega_{n\theta} = \frac{W_\theta}{S_\theta^{n-1}}, \quad (3)$$

第 n 层的搜索步长表示为

$$r_{nx} = \frac{\omega_{nx}}{S_x}, r_{ny} = \frac{\omega_{ny}}{S_y}, r_{n\theta} = \frac{\omega_{n\theta}}{S_\theta}, \quad (4)$$

第 n 层的位姿候选增量集合 $\bar{\mathbf{W}}_n$ 表示为

$$\begin{aligned} \bar{\mathbf{W}}_n = & \{ -\omega_{nx}, -\omega_{nx} + r_{nx}, \dots, \omega_{nx} - r_{nx}, \omega_{nx} \} \times \\ & \{ -\omega_{ny}, -\omega_{ny} + r_{ny}, \dots, \omega_{ny} - r_{ny}, \omega_{ny} \} \times \\ & \{ -\omega_{n\theta}, -\omega_{n\theta} + r_{n\theta}, \dots, \omega_{n\theta} - r_{n\theta}, \omega_{n\theta} \}, \end{aligned} \quad (5)$$

第 n 层的位姿搜索集合 $\bar{\mathbf{W}}_n$ 为

$$\bar{\mathbf{W}}_n = \{ \xi_{n-1} + \Delta\xi_n \mid \Delta\xi_n \in \bar{\mathbf{W}}_n \}, \quad (6)$$

式中: ξ_{n-1} 为第 $n-1$ 层分辨率中的最优匹配位姿; $\Delta\xi_n$ 为 $\bar{\mathbf{W}}_n$ 中的位姿候选增量。

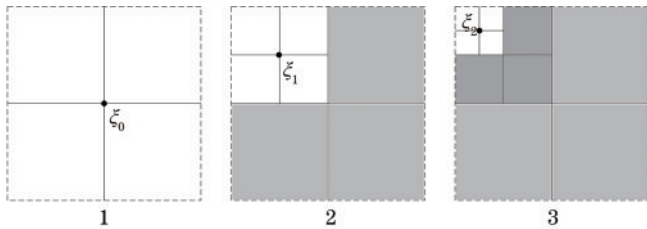


图 2 多分辨率搜索示意图

Fig. 2 Schematic of multi-resolution search

如图 2 所示,首先在位姿 ξ_0 附近建立搜索空间,进行全局搜索,搜索得到最优匹配位姿 ξ_1 (图 2 中的白色区域为搜索空间,非白色部分为候选位姿裁剪部分),将 ξ_1 作为下一个分辨率的初始位姿,并在 ξ_1 附近建立位姿搜索空间,提高分辨率,在 ξ_1 附近进行搜索,得到该层最优匹配位姿 ξ_2 ,重复上述步骤直到最后一层。

高分辨率位姿只在低分辨率区域划分,以此可以提高最优位姿的搜索效率。整个目标匹配过程中,通过匹配模型,计算目标函数值,作为位姿剔除的条件。

3.2 点云匹配模型

激光传感器旋转一圈,对所扫描的激光点云集合 $\mathbf{P} = \{ \mathbf{p}_k \}$ 与先验地图信息进行匹配时,需要将当前帧的点云坐标系扫描匹配到先验地图坐标系上,若当前机器人 2D 平面坐标系为 $\xi = (x, y, \theta)^T$,激光点云在障碍物上时所对应的点云集合为 \mathbf{p}_k ,点云 \mathbf{p}_k 转换到先验地图上的表达式为

$$\xi \oplus \mathbf{p}_k = \begin{pmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{pmatrix} \mathbf{p}_k, \quad (7)$$

式中: $\{ \xi \oplus \mathbf{p}_k \}$ 为当前帧转换到子图时所对应的点云集合。其中 \oplus 表示对当前帧点云坐标系进行的旋转平移操作,具体为

$$\xi \oplus \mathbf{p}_k = (x, y, \theta) \oplus \mathbf{p}_k = \mathbf{R}(\theta) \mathbf{p}_k + \mathbf{t}(x, y), \quad (8)$$

式中: $\mathbf{R}(\theta)$ 为旋转矩阵; $\mathbf{t}(x, y)$ 为平移向量。

3.3 ICP 匹配模型

针对传统的激光 SLAM 在进行移动机器人定位时需要依靠轮式编码器获得里程计信息或通过 IMU 来获取位置信息所存在的问题,采用基于 ICP 的方法来代替轮式里程计或者 IMU 作为激光里程计,利用机器人移动过程中相邻两帧激光点云来还原运动状态。ICP 利用点云匹配模型对当前帧与参考帧进行帧间匹配,通过近邻原则寻找待匹配点,并根据这些匹配点构建代价函数,表达式为

$$e = \operatorname{argmin}_{\xi} \sum_{i=1}^N \| \xi \oplus \mathbf{p}_k - \mathbf{q}_j \|^2, \quad (9)$$

式中: N 为当前帧与参考帧点云匹配的数量; \mathbf{q}_j 为参考帧点云集合。从而来寻找最佳的匹配参数 $\mathbf{R}(\theta)$ 和 $\mathbf{t}(x, y)$,进而获得机器人的位置信息,原理如图 3 所示。图 3 中的波浪线条表示障碍物的表面, \mathbf{p}_i 为待匹配点, \mathbf{x}_i 为上一帧点云中的点。 \mathbf{x}_i 与最近邻点 \mathbf{p}_i 进行匹配。但是该方法比较耗时且迭代后当前机器人的位

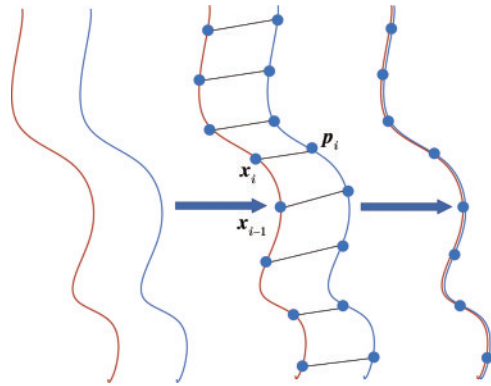


图 3 ICP 算法原理

Fig. 3 ICP algorithm principle

置已经发生改变,同时该方法的计算量庞大,匹配就会产生较大累计误差,影响机器人的定位精度,很难达到实时定位效果。

针对上述问题,本文采用一种基于文献[22]的当前帧与子图进行扫描匹配的方法,子图由连续几帧激光扫描组成。得到当前帧激光的扫描数据后,与当前建立的子图进行点云匹配,通过建立最小二乘模型,将当前帧的激光扫描数据插入到子图最优的位置,在不断插入新激光帧的同时该子图也得到了更新。当新的激光扫描数据插入到子图时,若子图不发生改变,就认为这个子图已经创建完成。最后利用之前构建的子图的最优位姿进行回环检测。具体过程如图4所示。通过连续的激光扫描(scan),创建出子图(submap)。通过匹配激光扫描当前帧与现有的子图信息,来获得机器人的位置信息。

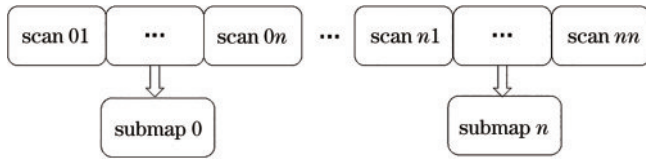


图4 子图构建

Fig. 4 Submap construction

通过将ICP算法与占据概率方法^[23]相结合,建立欧氏距离子图,建立点云匹配模型。将当前帧点云位姿转换到欧氏距离子图,使其到障碍物的欧氏距离和最小,其点云匹配的目标函数可描述为

$$e = \sum_{i=1}^N S^{\text{ref}}(\xi \oplus p_k), \quad (10)$$

式中: S^{ref} 为滤波后的欧氏距离值。点云匹配的问题可描述为求解式(10)的最小值,即

$$\xi_{w_n}^* = \operatorname{argmin} \sum_{i=1}^N S^{\text{ref}}(\xi \oplus p_k), \quad (11)$$

式中: \bar{W}_n 为初始位姿附近的搜索空间,为式(5)所示。

3.4 多点云密度匹配

对于多分辨率搜索得到的位姿集合,提出了一种

基于不同密度扫描点匹配以求得目标函数的方法,用来排除距离最优位姿较远的候选位姿,减少计算量,进一步加速搜索。点云的数量由少到多进行匹配,分步计算。该方法的核心是根据当前帧激光点云数量 N_p 来设定不同密度点云进行加速搜索。将当前帧的激光点云数划分为多层,进而逐层剔除非最优候选位姿。

图5为分层式搜索点云的示意图。图5中实心点代表该层需要进行扫描匹配的激光点,虚线连接的黑色点代表该层新增的待匹配点,灰色空心圆代表移动机器人当前所在位置。对于基于点云分层搜索的过程, M 为当前帧须要扫描的层数(M 的值过大时不利于收敛,过小时加速效果不明显,本文中 M 取3), n 为当前的层数, b_n 为当前层需要的激光点云数。每层扫描点云数可表示为

$$b_n = \frac{n}{M} \times N_p (n \leq M, n \in \mathbf{N}^+), \quad (12)$$

根据均匀采样的规则,其点云密度 d_n 为

$$d_n = \frac{n}{M}. \quad (13)$$

在进行多点云密度加速搜索时,点云密度的划分不会影响最优结果,但会对搜索的效率产生影响,因此在进行不同层扫描匹配时,每个待扫描点间隔 20° 选取,可以更好地均匀地进行激光点扫描。判断当前点云是否为新增待匹配点的函数 $C(k, n, M)$ 表示为

$$C(k, n, M) = \begin{cases} k, & k \bmod M = n \\ -1, & \text{else} \end{cases}, \quad (14)$$

式中: \bmod 表示求余数。若当前点为待匹配点,函数 $C(k, n, M)$ 将该点的序号返回,否则返回 -1 ,计算代价函数时就进行忽略。已知当前帧点云集合 \mathbf{p} 为 $\{\mathbf{p}_k\}$,则当前新增层待匹配点集合为 $\{\mathbf{p}_{C(k, n, M)}\}$ 。

通过式(10),可以得出在目标位姿下的 n 层的目标函数 e_n^d :

$$e_n^d = \sum_{j=1}^n \sum_{i=1}^{N_p} S^{\text{ref}}(\xi \oplus \mathbf{p}_{C(k, n, M)}). \quad (15)$$

定义搜索过程中的启发函数为

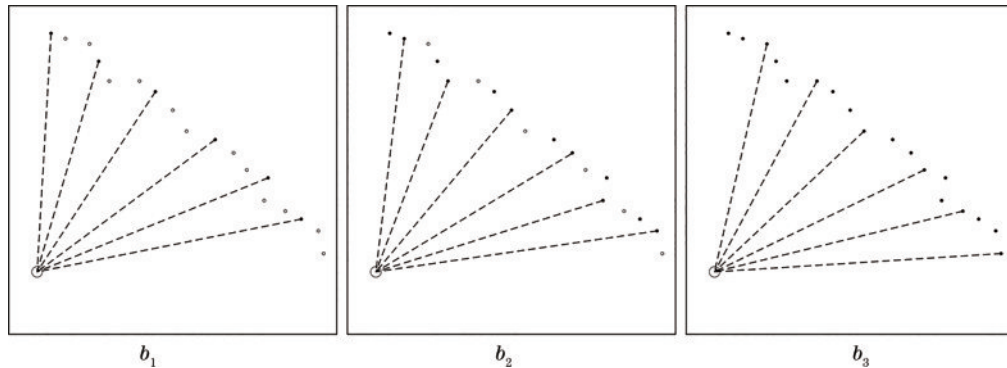


图5 多点云密度匹配示意图

Fig. 5 Schematic of multi-point cloud density matching

$$e_n^h = \frac{n}{M} \times e_n^d. \quad (16)$$

对于多分辨率加速搜索过程中的任意一层,计算其所对应的所有位姿节点在首层的启发函数值,同时建立位姿节点优先队列 Q_p ,将所有位姿节点添加至 Q_p ,并按启发函数 e_n^h 值的大小进行排列,如图 6 所示。从 Q_p 中选取最优位姿节点 q_{top} ,此时假设该节点的点云密度层数为 n ,计算其 $n+1$ 层的目标函数值 e_{n+1}^d ,记目前最优的目标函数值为 e_{best}^d ,若此时满足

$$\begin{cases} n+1 = M \\ e_{n+1}^d < e_{best}^d \end{cases}, \quad (17)$$

则更新 e_{n+1}^d 为 e_{best}^d ,此时的 e_{best}^d 所对应的位姿为最优位姿,结束搜索。若此时满足

$$\begin{cases} n+1 < M \\ e_{n+1}^d \geq e_{best}^d \end{cases}, \quad (18)$$

说明此时 e_{n+1}^d 所对应的候选位姿非全局最优,需要进行裁剪,提前结束该节点位姿匹配过程,如图 6 所示,虚线圆为裁剪的位姿节点。重复上述步骤,直至所有位姿节点搜索匹配完毕。搜索得到的最优位姿记为

$$\xi_n = \xi + \Delta\xi_1 + \Delta\xi_2 + \dots + \Delta\xi_n, \quad (19)$$

将 ξ_n 作为多分辨率加速搜索中下一个分辨率搜索的起点,直到将所有分辨率下的位姿搜索完成。

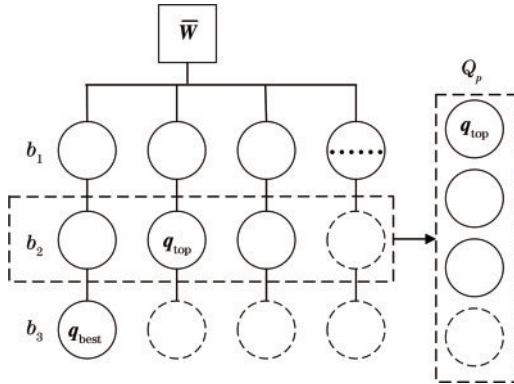


图 6 位姿裁剪示意图

Fig. 6 Diagram of pose cutting

4 稀疏矩阵位姿优化

为了得到全局最优且精度较高的解,对搜索得到的 ξ_n 进行非线性优化。采用图优化的方式,其中的位姿图由机器人在移动过程中非线性约束的位姿点组成,而这些约束也是位姿点周围的共同观测量。为了可以在实际过程中对移动机器人实现较高精度的定位,对于不同位姿之间的约束,采用 Levenberg-Marquardt(LM)作为框架,通过稀疏矩阵处理机器人位姿,该方法与视觉 SLAM 中通过 LM 处理相机传感器与外界环境的稀疏集束优化问题类似,最后采用直接稀疏 Cholesky decomposition 求解线性系统的方式来处理 2D 位姿图的优化问题^[24],进行位姿优化,确定

位置信息。

机器人在移动过程中的全局位姿表示为

$$\xi_i = [t_i, \theta_i] = [x_i, y_i, \theta_i], \quad (20)$$

式中: x_i 和 y_i 表示机器人的位置信息; θ_i 表示机器人的航向角。设 ξ_i 和 ξ_j 是式(20)中相邻的位姿节点,约束是位姿 ξ_i 和 ξ_j 之间的测量值,对于实际 ξ_i 和 ξ_j 之间的偏差为

$$h(\xi_i, \xi_j) = \begin{cases} R_i^T (t_j - t_i) \\ \theta_j - \theta_i \end{cases}, \quad (21)$$

式中: R_i^T 是 2×2 的旋转矩阵; $h(\xi_i, \xi_j)$ 为测量方程,即为估计变量和测量值间关系的方程。误差函数对应相邻两位姿节点之间的约束,形成的总误差为

$$e_{ij} = \bar{z}_{ij} - h(\xi_i, \xi_j), \quad (22)$$

式中: \bar{z}_{ij} 表示激光雷达获得的两位姿 ξ_i 和 ξ_j 的观测值。假设移动机器人的位姿误差服从高斯分布,该观测误差的平方定义为

$$\chi^2(\xi, p) = \sum_j e_{ij}^T \Lambda_{ij} e_{ij}, \quad (23)$$

式中: Λ_{ij} 为两位姿节点之间的观测,为信息矩阵。通过最小化式(22)和式(23)的误差,得到机器人位姿节点 ξ ,采用的方法为 LM。在位姿节点处进行迭代,求解的过程是首先将机器人的位姿节点 ξ 构建为向量 x ,构建误差函数 e ,其中 J 为雅各比矩阵。

$$\begin{cases} \Lambda = \begin{bmatrix} \Lambda_{ab} & & \\ & \ddots & \\ & & \Lambda_{mm} \end{bmatrix} \\ J = \frac{\partial e}{\partial x} \\ H = J^T \Lambda J \end{cases}. \quad (24)$$

构建的 LM 系统为

$$(H + \lambda \text{diag} H) \Delta x = J^T \Lambda e, \quad (25)$$

式中: λ 是拉格朗日正系数因子。对每一次测量的相邻位姿 $h(\xi_i, \xi_j)$ 之间的矩阵 H 增加下面 4 个分量,

$$\begin{matrix} \vdots & & & & \\ J_i^T \Lambda_{ij} J_i & \cdots & J_i^T \Lambda_{ij} J_j & & \\ \vdots & & \vdots & & \\ J_j^T \Lambda_{ij} J_i & \cdots & J_j^T \Lambda_{ij} J_j & & \\ \vdots & & & & \end{matrix}, \quad (26)$$

式中: J_i 相当于 e_{ij} 相对 c_i 的雅各比,每个分量都是 3×3 的矩阵块。在式(26)中的右边根据每个约束增加 3×1 的矩阵块 $J_i^T \Lambda_{ij} e_{ij}$ 和 $J_j^T \Lambda_{ij} e_{ij}$,求解完上述系统可以得到一个增量 Δx ,将该增量更新到当前位姿的 x :

$$\begin{cases} t_i = t_i + \Delta t \\ \theta_i = \theta_i + \Delta\theta_i \end{cases}. \quad (27)$$

通过 CSparse 功能包压缩列存储(CCS)格式处理稀疏矩阵,来有效地求解稀疏矩阵,进而求解式(25)所构建的线性系统。该功能包含求解稀疏线性系统的 Cholesky decomposition,可以高效进行机器人位姿优化,进而获得精确位置信息。

5 实验过程与结果分析

5.1 公开数据集测试

为了验证所提方法的有效性,实验在 CPU AMD Ryzen 7 5800H CPU 主频 3.2 GHz 的计算机平台上进行。采用公开数据集,衡量所提方法主要性能的指标,包括算法的实时性以及定位的平移误差和平移方差。

为了测试所提方法的实时性效果,采用 SLAM

Benchmark 中的 Intel、ACES、MIT Killian Court 数据集与典型的激光 SLAM 进行算法效率对比,各数据集的参数如表 1 所示,包括各数据集时长、算法的运行时间、算法的效率。Cartographer 算法的运行时间由文献[13]给出。

由表 1 可得,所提算法具有较高的实时性,相对于现在较为流行的 Cartographer 算法,所提算法具有明显的优势,算法效率是 Cartographer 算法的 1.8~4.9 倍,可以高效地对移动机器人进行实时位置估计。

表 1 定位估计效率对比

Table 1 Comparison of location estimation efficiency

Dataset	Total duration of the dataset /s	Total time of Cartographer /s	Total time of proposed method /s	Efficiency ratio
ACES	1366	41	22.3	1.8
Intel	2691	179	37.6	4.7
MIT Killian Court	7678	190	39.0	4.9

为了验证所提方法对移动机器人位置信息的定位精度,同样使用 Intel、ACES、MIT Killian Court 数据集进行测试,平移误差和方差评定采用文献[25]所述的

方法。将所提算法与 Gmapping 算法和 Cartographer 算法进行对比,3 种方法均采用里程计提供初始位姿,结果如表 2 所示。

表 2 定位估计误差对比

Table 2 Comparison of positioning estimation error

unit: m

Method	Intel		ACES		MIT Killian Court	
	Translation error	Translation variance	Translation error	Translation variance	Translation error	Translation variance
Proposed method	0.0181	0.0008	0.0365	0.0012	0.0583	0.0049
Cartographer	0.0299	0.0011	0.0375	0.0032	0.0395	0.0039
Gmapping	0.031	0.002	0.044	0.004	0.050	0.006

由表 2 可以看出:所提方法平移误差和方差在 Intel 数据集上相比 Cartographer 算法分别降低了 0.0118 m 和 0.0003 m,相比 Gmapping 算法分别降低了 0.0129 m 和 0.0012 m;在 ACES 数据集上,相比 Cartographer 算法分别降低了 0.0010 m 和 0.0020 m,相比 Gmapping 算法分别降低了 0.0075 m 和 0.0028 m;而在 MIT Killian Court 数据集上误差略有升高,但总体定位精度有所提高。

5.2 在 Turtlebot2 机器人上的实际测试

为了实际验证所提方法的绝对定位与重定位精度,对照组选取当前常用的 Gmapping 和 Cartographer 定位方法,绝对定位精度的评价指标为最大误差、平均误差、均方根误差(RMSE)和算法运行时间。对重定位精度的性能指标与当前常用的 AMCL 重定位进行比较,评价指标包括最大误差、最小误差、平均误差。

所有对比实验在 CPU AMD Ryzen 7 5800H CPU 主频 3.2 GHz 的计算机平台上进行,均运行于 Ubuntu20.04、ROS Noetic 中,以 Turtlebot2 机器人为实验平台,如图 7 所示,其搭载 RPLIDAR A1 激光雷达,各性能指标如表 3 所示。实验场地如图 8(a)所示,应用场景大小为 7.5 m×9.2 m。运动过程中以无



图 7 实验设备

Fig. 7 Experimental facility

线键盘操作节点来控制移动机器人运动,机器人的平均速度为 0.15 m/s,平均角速度为 57.3 (°)/s,最大瞬时速度为 0.65 m/s,角速度的最大值为 180 (°)/s。对于扫描频率为 5.5 Hz 的激光雷达 RPLIDAR A1,

表 3 RPLIDAR A1 激光雷达参数
Table 3 RPLIDAR A1 lidar parameter

Parameter	Value
Measuring radius / m	0.15–12
Sampling frequency / 10 ³	8
Scanning frequency / Hz	5.5
Angular resolution / (°)	≤1
Scanned area / (°)	360
Accuracy of ranging	actual distance 1% (≤3 m)
	actual distance 2% (3–5 m)
	actual distance 2.5% (5–12 m)

其单帧搜索的最大空间为 $W_{\max} = (0.13 \text{ m}, 0.13 \text{ m},$

$36^\circ)$, 定义的最小搜索步长为 $r_{\min} = (0.015 \text{ m}, 0.015 \text{ m}, 0.5^\circ)$ 。

针对实际生产环境中需要到达指定位置任务点的要求,需要测试移动机器人的重复定位精度。通过在先验地图上标定不同位置,选取 A、B、C、D 四组点,分别为 $(2.47, 0, 0)$ 、 $(4.95, 0.65, 0.524)$ 、 $(5.71, 0.35, 0.785)$ 、 $(3.73, 4.25, 1.047)$,记录位置信息。实验开始时从地图坐标系原点出发,在事先建好的先验地图上通过键盘操作节点控制移动机器人沿不同的路径到达事先标定的位置。先验地图采用 ICP 距离像^[26],如图 8(b)所示。每个位置记录 10 次数据,计算各标定点位姿输出与目标点位姿间的误差,结果如图 9 所示。

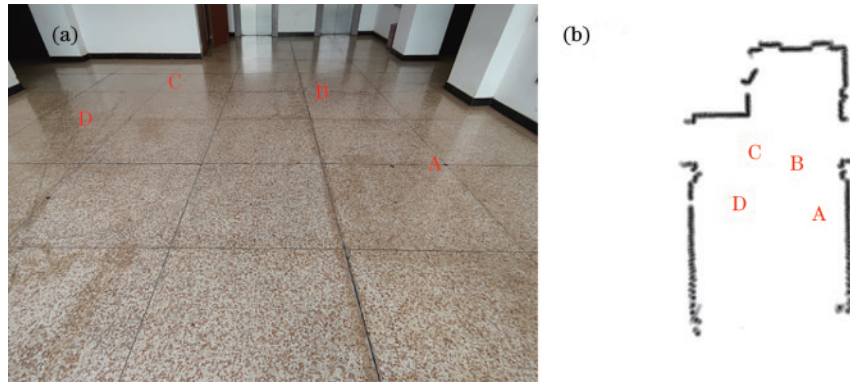


图 8 重定位实验。(a) 实验环境;(b) 距离像

Fig. 8 Relocation experiment. (a) Experimental environment; (b) distance image

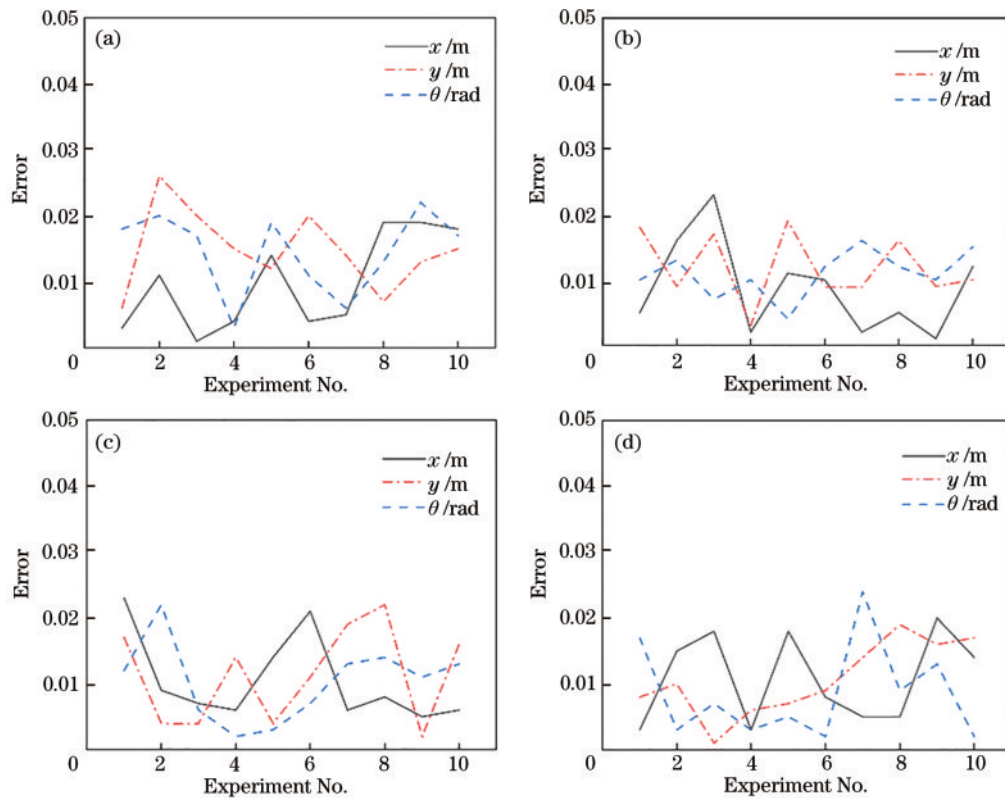


图 9 重定位误差。(a) A 点;(b) B 点;(c) C 点;(d) D 点

Fig. 9 Relocation error. (a) Point A; (b) point B; (c) point C; (d) point D

选取 4 个目标点来验证所提方法重复定位的精度,并与文献[14]中 AMCL 重定位的数据进行比较,结果如表 4 所示。结果表明:所提方法各个维度最大定位误差和平均定位误差均有明显降低;相比传统的 AMCL,所提方法的重定位精度有显著的提高,可以较好地应用于引导服务机器人和工业 AGV 到达指定位置的领域。

为分析所提方法定位效果相对目前现有方案的差别,实验选取如图 10(a)所示的大厅, Turtlebot2 移动

机器人沿如图 10(a)所示长为 3.8 m、宽为 2.5 m 的 S 形路径进行运动,整个 S 形路径分为 5 部分。分别采用不同的算法使 Turtlebot2 机器人沿线条轨迹运动,采用 Gmapping 和 Cartographer 作对比,进行 15 次实验,记录每种算法的定位信息,取定位结果的平均值进行比较。在实验开始时,为了可以较为准确地比较三种算法的区别,需要同时运行三种算法,三种算法采用相同的频率记录位置信息,以实现数据的精准匹配。

表 4 重定位误差对比

Table 4 Relocation error comparison

Parameter	AMCL			Proposed method		
	x / m	y / m	θ / rad	x / m	y / m	θ / rad
Absolute value of max error	0.068	0.045	0.026	0.023	0.026	0.024
Absolute value of min error	0.017	0.006	0.005	0.001	0.002	0.002
Absolute value of mean error	0.045	0.028	0.013	0.010	0.012	0.012

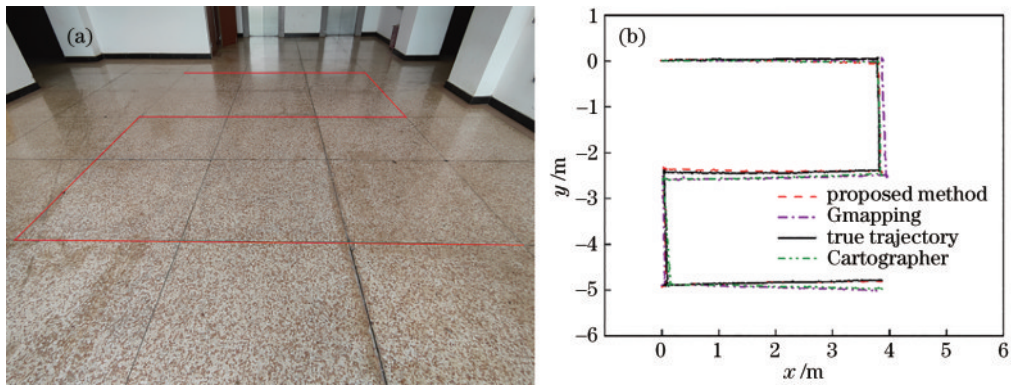


图 10 定位估计实验。(a) 机器人路径;(b) 不同算法定位轨迹

Fig. 10 Location estimation experiment. (a) Robot path; (b) positioning trajectory of different algorithms

实验中通过 ros 命令节点记录移动机器人的位姿信息,绘制图 10(b)所示曲线。图 10(b)中坐标原点为移动机器人运动的起点,可以直观地看出:所提方法得到的路径更加贴合实际路径,与实际位置没有过大的偏差;而现在主流的 Cartographer 算法进行定位时,会对位姿进行非线性优化,定位精度略低于所提算法;Gmapping 算法进行定位时,严重依赖里程计,存在累计误差,与真实的位置信息存在相对较大的误差。所提

方法对后端进行了稀疏矩阵位姿优化,提高了移动机器人的定位精度。

为了可以精确地描述所提方法与其他方法在移动机器人定位中的误差,记录 x 和 y 方向上产生的误差,同时为了避免数据量过大,通过 ros 命令节点以每秒 10 次的频率记录位置信息,沿 S 形路径总耗时为 150 s,最终的误差如图 11 所示。机器人的运行平均速度为 0.15 m/s。从图 11 可以看出:所提方法具有

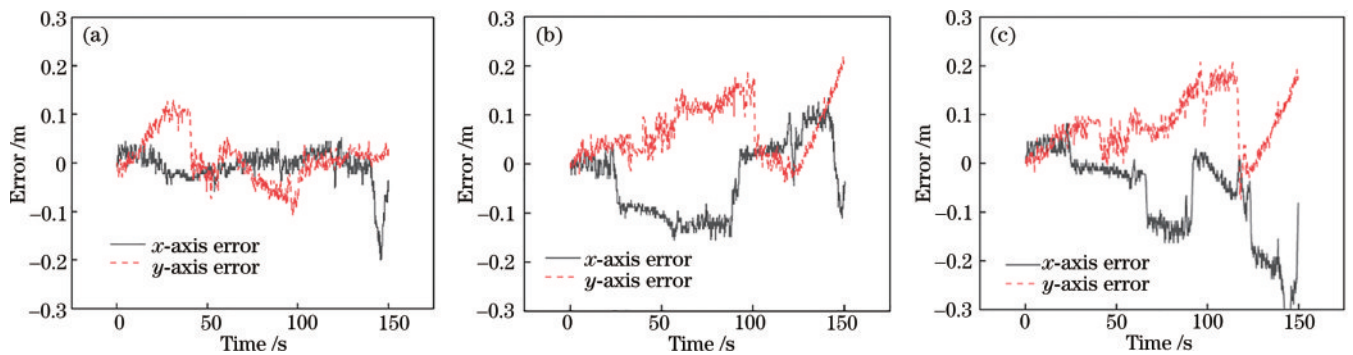


图 11 定位精度误差。(a) 所提方法;(b) Cartographer;(c) Gmapping

Fig. 11 Positioning accuracy error. (a) Proposed method; (b) Cartographer; (c) Gmapping

较高的鲁棒性,误差峰值在 x 方向控制在 0.2 m 以内, y 方向误差控制在 0.12 m 以内; Gmapping 和 Cartographer 进行定位的误差和波动相对偏大, Gmapping 算法中定位最大误差超过了 0.3 m。因此所提方法可以较高精度地对移动机器人进行定位。从表 5 数据可以得出:所提方法的定位精度相比另外两种方法有明显的提高;相比 Cartographer 方法,所提方法在 x 和 y 轴方向的均方根误差降低了 0.0470 m 和 0.0409 m, 平均误差分别降低了 0.0486 m 和

0.0357 m; 相比 Gmapping 方法,最大误差、平均值、均方根误差都有明显的降低;同时也对三种算法的测试时间进行了对比,因为引入了多分辨率结合多点云密度的方式,剔除了大量的非优位姿,同时减少了 ICP 算法的计算量,所提方法的算法效率分别是 Cartographer 和 Gmapping 算法的 3.59 倍和 2.97 倍,极大地提高了运算效率,减少了电脑 CPU 的占用率。因此所提方法在实际应用中有着明显的优势,定位信息更接近于真实值。

表 5 三种算法定位误差综合对比

Table 5 Comprehensive comparison of positioning errors among three algorithms

Location method	Maximum error along x -axis /m	Average error along x -axis /m	RMSE along x -axis /m	Maximum error along y -axis /m	Average error along y -axis /m	RMSE along y -axis /m	Running time /s
Gmapping	0.385	0.0786	0.1147	0.207	0.0819	0.0975	11.0
Cartographer	0.156	0.0707	0.0827	0.222	0.0713	0.0894	13.3
Proposed method	0.2	0.0221	0.0357	0.133	0.0356	0.0485	3.7

通过上述实验,结果表明所提算法在移动机器人定位过程中的误差相比另外两种算法更小,能够处理机器人在复杂环境中的定位情况,定位结果更加精确。

6 结 论

为实现移动机器人在移动过程中实时感知自身的位置并取得高精度的定位效果,提出了一种改进的 ICP 定位方式,建立多分辨率位姿节点搜索空间,分辨率由粗到精,每一层分辨率的搜索空间均在上一层搜索位姿最优值附近建立,提高定位精度,加速搜索;同时结合多点云密度的方式,以不同密度的点云与所构建的欧氏距离子图进行 ICP 算法匹配,在这个过程中只需要对部分点云进行匹配和位姿剔除,减少了实际点云匹配量,保证了定位的实时性。对于前端得到的较优位姿,通过 LM 方法进行稀疏矩阵位姿优化,确定机器人位置信息,并与其他主流激光 SLAM 进行对比。结果表明,所提方法具有较高定位精度的同时保证了实时性,可以广泛地用于工业或者服务型机器人实时的位置估计。

参 考 文 献

- [1] 塞巴斯蒂安·特龙, 沃尔弗拉姆·比加尔, 迪特尔·福克斯. 概率机器人[M]. 曹红玉, 谭志, 史晓霞, 等, 译. 北京: 机械工业出版社, 2019.
Thrun S, Burgard W, Fox D. Probabilistic robotics[M]. Cao H Y, Tan Z, Shi X X, et al., Transl. Beijing: China Machine Press, 2019.
- [2] 宋桂岭, 明安龙. 移动机器人开发技术: 激光 SLAM 版[M]. 北京: 机械工业出版社, 2022: 152-181.
Song G L, Ming A L. Development technology for autonomous mobile robots[M]. Beijing: China Machine Press, 2022: 152-181.
- [3] 朱凯, 刘华峰, 夏青元. 基于单目视觉的同时定位与建

图算法研究综述[J]. 计算机应用研究, 2018, 35(1): 1-6.

Zhu K, Liu H F, Xia Q Y. Survey on monocular visual SLAM algorithms[J]. Application Research of Computers, 2018, 35(1): 1-6.

- [4] 毕京学, 汪云甲, 宁一鹏, 等. 顾及 BLE 信标几何优化的室内测距定位方法[J]. 中国矿业大学学报, 2021, 50(2): 411-416.

Bi J X, Wang Y J, Ning Y P, et al. Indoor range-based positioning method considering geometry optimization of BLE beacons[J]. Journal of China University of Mining & Technology, 2021, 50(2): 411-416.

- [5] 何佳泽, 张寿明. 2D 激光雷达移动机器人 SLAM 系统研究[J]. 电子测量技术, 2021, 44(4): 35-39.

He J Z, Zhang S M. Research on SLAM system of 2D lidar mobile robot[J]. Electronic Measurement Technology, 2021, 44(4): 35-39.

- [6] 申文杰, 任永杰, 朱博源, 等. 结合人工信标和视觉 SLAM 的自主快速定位方法[J]. 机械工程学报, 2022, 58(14): 137-145.

Shen W J, Ren Y J, Zhu B Y, et al. Autonomous fast localization method combining artificial beacons and visual SLAM[J]. Journal of Mechanical Engineering, 2022, 58(14): 137-145.

- [7] 秦岭, 王东星, 史明泉, 等. 基于遗传算法优化 ELM 神经网络的室内可见光定位系统[J]. 中国激光, 2022, 49(21): 2106001.

Qin L, Wang D X, Shi M Q, et al. Indoor visible light localization system based on genetic algorithm-optimized extreme learning machine neural network[J]. Chinese Journal of Lasers, 2022, 49(21): 2106001.

- [8] 赵霞, 张君毅, 龙倩倩. 基于 Circle 混沌映射的 ISSA-ELM 神经网络室内可见光定位方法[J]. 光学学报, 2023, 43(2): 0206004.

Zhao X, Zhang J Y, Long Q Q. ISSA-ELM neural network indoor visible light location method based on Circle chaotic mapping[J]. Acta Optica Sinica, 2023, 43(2): 0206004.

- [9] Qin C Y, Zhang H, Li S, et al. Mapping non-laminar proton acceleration in laser-driven target normal sheath field[J]. High Power Laser Science and Engineering, 2021, 10(1): e2.
- [10] 高翔, 张涛, 刘毅. 视觉 SLAM 十四讲: 从理论到实践[M]. 2 版. 北京: 电子工业出版社, 2019: 119-148.
Gao X, Zhang T, Liu Y. Fourteen lectures on visual SLAM: from theory to practice[M]. 2nd ed. Beijing: Publishing House of Electronics Industry, 2019: 119-148.
- [11] 沈斯杰, 田昕, 魏国亮, 等. 基于 2D 激光雷达的 SLAM 算法研究综述[J]. 计算机技术与发展, 2022, 32(1): 13-18, 46.
Shen S J, Tian X, Wei G L, et al. Review of SLAM algorithm based on 2D lidar[J]. Computer Technology and Development, 2022, 32(1): 13-18, 46.
- [12] 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, November 1-5, 2011, Kyoto, Japan. New York: IEEE Press, 2011: 155-160.
- [13] 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), May 16-21, 2016, Stockholm, Sweden. New York: IEEE Press, 2016: 1271-1278.
- [14] 冯佳萌, 裴东, 邹勇, 等. 基于机器人激光定位的一种改进 AMCL 算法[J]. 激光与光电子学进展, 2021, 58(20): 2028003.
Feng J M, Pei D, Zou Y, et al. An improved AMCL algorithm based on robot laser localization[J]. Laser & Optoelectronics Progress, 2021, 58(20): 2028003.
- [15] 白崇岳, 王建军, 程霄霄, 等. 融合激光 SLAM 实现无人驾驶轮椅空间定位优化[J]. 激光与光电子学进展, 2022, 59(2): 0228007.
Bai C Y, Wang J J, Cheng X X, et al. Spatial positioning optimization of driverless wheelchair by fusion of laser SLAM[J]. Laser & Optoelectronics Progress, 2022, 59(2): 0228007.
- [16] 王子润, 燕必希, 董明利, 等. 基于激光雷达与改进 AMCL 的爬壁机器人定位方法[J]. 仪器仪表学报, 2022, 43(12): 220-227.
Wang Z R, Yan B X, Dong M L, et al. Localization method of wall-climbing robot based on lidar and improved AMCL[J]. Chinese Journal of Scientific Instrument, 2022, 43(12): 220-227.
- [17] 黄永新. 基于复合点云帧改进的激光 SLAM 闭环检测算法[J]. 自动化与仪表, 2020, 35(6): 42-47.
- Huang Y X. Improved loop closure detection algorithm of laser SLAM based on multi-scans-to-map matching method[J]. Automation & Instrumentation, 2020, 35(6): 42-47.
- [18] Besl P J, McKay N D. A method for registration of 3-D shapes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239-256.
- [19] Akca D, Gruen A. Fast correspondence search for 3D surface matching[EB/OL]. [2023-03-02]. <https://www.isprs.org/proceedings/XXXVI/3-W19/papers/186.pdf>.
- [20] Olson E B. Real-time correlative scan matching[C]//2009 IEEE International Conference on Robotics and Automation, May 12-17, 2009, Kobe, Japan. New York: IEEE Press, 2009: 4387-4393.
- [21] Jost T, Hugli H. A multi-resolution scheme ICP algorithm for fast shape registration[C]//Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission, June 19-21, 2002, Padua, Italy. New York: IEEE Press, 2002: 540-543.
- [22] 丁元浩, 吴怀宇, 陈洋. 基于 2D 激光扫描到子图匹配的 SLAM 方法[J]. 计算机工程与设计, 2020, 41(12): 3458-3463.
Ding Y H, Wu H Y, Chen Y. SLAM method based on 2D laser scan-to-submap matching[J]. Computer Engineering and Design, 2020, 41(12): 3458-3463.
- [23] 李鑫. 基于全局优化匹配的激光 SLAM 方法研究[D]. 厦门: 厦门大学, 2021: 26-41.
Li X. Research on laser SLAM method based on global optimization matching[D]. Xiamen: Xiamen University, 2021: 26-41.
- [24] 申东, 徐雨航, 李强, 等. 基于稀疏位姿优化的激光 SLAM 算法研究[J]. 激光与光电子学进展, 2021, 58(24): 2428003.
Shen D, Xu Y H, Li Q, et al. Research on laser SLAM algorithm based on sparse pose optimization[J]. Laser & Optoelectronics Progress, 2021, 58(24): 2428003.
- [25] Kümmerle R, Steder B, Dornhege C, et al. On measuring the accuracy of SLAM algorithms[J]. Autonomous Robots, 2009, 27(4): 387-407.
- [26] 李鑫, 仲训昱, 彭侠夫, 等. 基于多分辨率搜索与多点云密度匹配的快速 ICP-SLAM 方法[J]. 机器人, 2020, 42(5): 583-594.
Li X, Zhong X Y, Peng X F, et al. Fast ICP-SLAM method based on multi-resolution search and multi-density point cloud matching[J]. Robot, 2020, 42(5): 583-594.