

Research Article

An adaptive compensation strategy for sensors based on the degree of degradation

Yanbin Li^a, Wei Zhang^{a,*}, Zhiguo Zhang^a, Xiaogang Shi^a, Ziruo Li^b, Mingming Zhang^c, Wenzheng Chi^{d,*}^a School of Electronics Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China^b Key Lab of Smart Agriculture Systems, Ministry of Education, China Agricultural University, Beijing 100083, China^c School of Integrated Circuit Science and Engineering, Beihang University, Beijing 100191, China^d Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, China

ARTICLE INFO

Article history:

Received 18 January 2025

Revised 14 April 2025

Accepted 18 April 2025

Available online 30 April 2025

Keywords:

Anti-degradation

Multilayer perceptron

Lidar SLAM

Particle filter

ABSTRACT

Simultaneous Localization and Mapping (SLAM) is widely used to solve the localization problem of unmanned devices such as robots. However, in degraded environments, the accuracy of SLAM is greatly reduced due to the lack of constrained features. In this article, we propose a deep learning-based adaptive compensation strategy for sensors. First, we create a dataset dedicated to training a degradation detection model, which contains coordinate data of particle swarms with different distributional features, and endow the model with degradation detection capability through supervised learning. Second, we design a lightweight network model with short computation time and good accuracy for real-time degradation detection tasks. Finally, an adaptive compensation strategy for sensors based on the degree of degradation is designed, where the SLAM is able to assign different weights to the sensor information according to the degree of degradation given by the model, to adjust the contribution of different sensors in the pose optimization process. We demonstrate through simulation experiments and real experiments that the robustness of the improved SLAM in degraded environments is significantly enhanced, and the accuracy of localization and mapping are improved.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Simultaneous localization and mapping (SLAM) allows robots or self-driving vehicles to simultaneously perform localization and map construction in unknown environments. It processes sensor data to sense the surrounding environment, updating the position and environment map in real time [1]. SLAM based on particle filter is widely used because of its high efficiency, among which GMapping has become the first choice for sweeping robots and service robots because the two-dimensional grid map constructed is suitable for robot navigation tasks [2].

However, SLAM may suffer from degradation problems in certain situations. For example, in feature-sparse scenarios, such as narrow corridors or tunnel environments, the matching relationship between sensor observations and map features fails, resulting in drift of localization. The degradation issues of monocular camera SLAM mainly stem from its inability to directly measure the depth information of objects [3–9], leading to scale uncertainty, which means it cannot accurately determine the

actual size and distance of objects. In environments with sparse features, monocular SLAM struggles to extract sufficient feature points for effective matching and tracking, thereby affecting the accuracy of localization and mapping. At the same time, the lack of sufficient geometric features results in sparse point cloud data captured by the lidar, and the lidar ranging information will be consistent in the long straight corridor, the constraints provided by lidar are relatively insufficient in these situations, making it difficult to establish a match with the map features, which in turn affects the accuracy of localization and mapping [10–13].

The key challenge is to adaptively compensate for the degraded state of SLAM by dynamically adjusting the contribution weights of different sensors to pose optimization based on their respective strengths. Existing multi-sensor fusion strategies often employ a tightly coupled approach, which results in insufficient dynamic fusion capabilities [13–16]. In degraded environments, erroneous information from failed sensors can propagate through the pose matching process to the entire system, leading to localization failure. Moreover, dynamic switching mechanisms may cause information loss between multimodal data, resulting in large variations in pose optimization and consequently affecting overall localization accuracy [17].

* Corresponding authors.

E-mail addresses: weizhang13@bupt.edu.cn (W. Zhang), wzchi@suda.edu.cn (W. Chi).

In recent years, deep learning has been used to improve SLAM performance in complex environments. Liu et al. [18] used Convolutional Neural Network (CNN) for feature extraction and processing of lidar data to enhance the performance of SLAM in dynamic urban environments. Zhang et al. [19] used Generative Adversarial Network (GAN) to generate realistic image data to supplement sensor information to enhance the performance of SLAM in low-light conditions. Li et al. [20] used transformer architecture to learn global features for loopback detection. Lian et al. [21] built neuroimplicit representations through multilayer perceptron (MLP) to improve the adaptability of SLAM in dynamic environments.

This article proposes an adaptive sensor compensation strategy for particle filter-based SLAM using deep learning. First, we design a lightweight neural network model for degradation detection, which is trained via supervised learning. The model can assess the degree of environmental degradation based on the distribution characteristics of the particle set and outputs a degradation factor. Subsequently, the contribution weights of different sensors are adjusted according to this degradation factor. By influencing the likelihood scores, the degradation factor steers the pose optimization towards relying more on the sensor constraints that are more reliable in the current environment, thereby achieving dynamic compensation for degradation. This article is extended from our previous conference work [22] and the main contributions of this article are summarized as follows:

- A coordinate dataset of particle swarms for the degradation detection task distinguishes particle swarms with various distributional features by different labels;
- A lightweight network model with the ability to analyze the distributional characteristics of the particle swarms to enable degradation detection;
- An adaptive compensation strategy for sensors that is able to change the contribution of different sensors to the SLAM system depending on the degree of degradation.

2. Related work

Multi-sensor fusion strategy is used to reduce degradation by integrating the advantages of different sensors. Rahman et al. [23] developed SVIn2, a multi-sensor fusion SLAM system for underwater environments. It integrates sonar, Inertial Measurement Unit (IMU), and other sensors to provide geometric information and motion constraints, improving localization and mapping accuracy. Jia et al. [24] created Lvio-fusion, a self-adaptive SLAM framework using an actor-critic method to adjust sensor weights. It fuses stereo camera, LiDAR, IMU, and Global Positioning System (GPS) data, employing graph optimization and reinforcement learning for dynamic weight adjustment and robust pose estimation. Frosi et al. [25] introduced MCS-SLAM, a system that integrates multiple sensor cues like color, depth, and normals into a graph SLAM framework. Li et al. [26] proposed AVM-SLAM, using multi-sensor fusion in a bird's-eye view. It combines fisheye cameras, wheel encoders, and IMU, with CNN for semantic feature extraction and semantic Iterative Closest Point (ICP) for loop closure, ensuring accurate mapping and robustness. Wang et al. [27] developed mvil-fusion, a monocular visual-inertial-lidar SLAM system optimized for challenging environments. It tightly integrates data from a monocular camera, IMU, and LiDAR, leveraging LiDAR structural information and IMU motion constraints for high-precision localization and mapping. Tao et al. [28] presented a multi-sensor fusion localization strategy for intelligent vehicles using global pose graph optimization. It combines data from LiDAR, IMU, GPS, and visual sensors to achieve high-precision vehicle localization. The proposed method achieves multi-sensor

fusion by means of tight coupling, resulting in the error information of the failed sensor in the degraded environment propagating to the whole system through the pose matching process, increasing the positioning error. At the same time, in complex environments, it may not be possible to quickly respond to sensor failure or degradation through the adaptive mechanism, which affects the positioning accuracy. In this article, the contribution weight of different sensors to pose optimization is dynamically adjusted by the degradation factor based on the model output, so as to achieve adaptive sensor information fusion.

Some researches have improved the degradation resistance of SLAM systems by optimizing methods such as feature extraction and closed-loop detection. Gao et al. [29] proposed FELC-SLAM, which employs advanced ground segmentation methods and feature segmentation strategies, including ground features, edge features, planar features, and spherical features. Additionally, by introducing a robust decoupled global registration method for loop closure detection in the backend, it effectively addresses the sparsity of long-distance point clouds and the degradation issues caused by the reduction of inner layers in point cloud registration. Lin et al. [30] presented a Lidar SLAM algorithm tailored using dynamic feature point extraction for coal mine environments. The algorithm adapts to the unique challenges of coal mine environments, such as limited visibility and complex structures, to improve SLAM performance. Tsai et al. [31] introduced a point cloud feature extraction method based on ground segmentation. By improving the segmentation algorithm for ground points and the extraction of edge features, the method ensures the stability and geometric characteristics of features. Implementing the proposed point cloud preprocessing techniques on LeGO-LOAM achieves higher accuracy while reducing rotation and translation errors in most dataset sequences. Xiao et al. [32] proposed SL-SLAM, a visual-inertial SLAM system based on deep feature extraction and matching. The system uses the Superpoint network for feature point detection and LightGlue for feature matching to improve the accuracy and robustness of matching. But these methods struggle to adapt to dynamic environments and can be affected by fast-moving objects, leading to incorrect feature matching and impacting localization accuracy. And their robustness varies across environments, with feature extraction being less effective in scenes with changing lighting or lack of texture. Moreover, these algorithms heavily rely on sensors, so sensor noise and failures can accumulate and compromise the SLAM stability and reliability. In this article, degradation factors are used to influence the likelihood score in the pose optimization process, so as to achieve smooth and robust constraint compensation.

3. Rao-Blackwellized particle filter

In this section, we outline the updating process of the Rao-Blackwellized Particle Filter (RBPF) within the SLAM algorithm and substantiate the feasibility of the approach advocated in this article. In this article, Gmapping is used as a research object, and its core mechanism relies on the RBPF to deal with the uncertainty and nonlinearity of the pose. The RBPF achieves the state estimation by maintaining a set of particles representing the probability distribution of the poses, where the dispersion of the particles directly reflects the uncertainty in the estimation of the poses.

The updating process of RBPF is shown in Algorithm 1. First, a set of particles is initialized according to the probability distribution $p(s_0)$ of the initial state of the robot. Subsequently, the particles are updated by using odometry information of motion model in Eq. (1).

$$s_t^{(i)} = f(s_{t-1}^{(i)}, u_{t-1}) + \eta_{t-1}^{(i)} \quad (1)$$

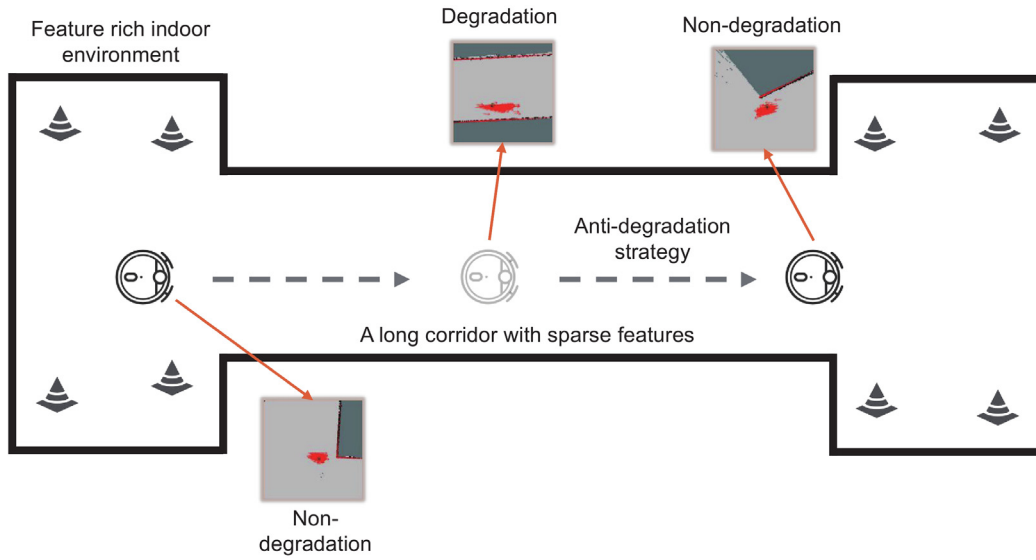


Fig. 1. The schematic diagrams show the divergent and convergent states of the particle swarm in the degraded and non-degraded environments, respectively.

where $s_{t-1}^{(i)}$ denotes the distribution of i th particle in the particle swarm at the moment of $t-1$; u_{t-1} denotes the odometry data at the moment of $t-1$; and $\eta_{t-1}^{(i)}$ denotes the noise to which the i th particle is exposed.

After that, the particles are further updated by using the lidar observation data and the weights of the particles are calculated according to Eq. (2).

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_{t-1}|x_{t-1}^{(i)}) \quad (2)$$

Herein, $w_{t-1}^{(i)}$ denotes the weight of i th particle at the moment of $t-1$; z_{t-1} denotes the lidar scan data at the moment of $t-1$; and $x_{t-1}^{(i)}$ denotes the positional attitude of the i th particle at the moment of $t-1$. This step reflects the adjustment of particle weights by the observation model. If particles exhibit a clustered distribution, it indicates high localization accuracy. Conversely, a dispersed particle distribution suggests lower localization accuracy and SLAM degradation, as shown in Fig. 1, which supports the feasibility of using neural networks for degradation detection.

In order to maintain the robustness of pose estimation, we utilize N_{eff} as a metric to assess the variety within the particle set. Based on the specific value of N_{eff} , particles undergo resampling, where particles with higher weights are retained while those with lower weights are eliminated, thus facilitating the restoration of the representativeness of the particle set. Finally, the position and map of the particles are updated, and this step involves the reconstruction of the map and the refinement of the position based on the particles with the latest weight adjustments.

4. Degradation detection model

4.1. Dataset for the model

We utilize an Ackermann-type wheeled robot for data collection and validation of anti-degradation system. The robot is equipped with an LSlidar M10P, which is powered by an ARM Cortex-A72 64-bit CPU and a Broadcom VideoCore VI GPU, as shown in Fig. 3.

In our experimental setup, we expose the robot to both degraded and non-degraded environments. Specifically, the outdoor scene is designated as the degraded environment, while the indoor scene is considered the non-degraded environment. Upon

Algorithm 1: The process of updating RBPF

Input: The particle swarm at the previous moment S_{t-1} , the laser scan z_{t-1} and the odometry data u_{t-1}

Output: The particle swarm at the current moment S_t

- 1 Initialize a set of random particles $s_0^{(i)} \sim p(s_0)$
- 2 **while** True **do**
- 3 **for all** $s_{t-1}^{(i)}$ in S_{t-1} **do**
- 4 Initial update of particles $s_{t-1}^{(i)}$ based on the motion model
- 5 Optimization of particle weights $w_{t-1}^{(i)}$ based on observation models
- 6 Normalize the weights according to

$$w_{t-1}^{(i)} = \frac{w_{t-1}^{(i)}}{\sum_{i=1}^N w_{t-1}^{(i)}}$$
- 7 Calculate the effective sample size used to measure particle diversity according to $N_{eff} = \frac{1}{\sum_{i=1}^N (\omega^{(i)})^2}$
- 8 **if** $N_{eff} \leq \text{threshold}$ **then**
- 9 Remove low weight particles and keep high weight particles
- 10 **else**
- 11 Keep all particles
- 12 Update the estimated position of the particle $x_t^{(i)}$
- 13 Update the map for each particle $m_t^{(i)}$
- 14 **return** S_t

activating the SLAM node, we meticulously record the coordinates $(x, y) \in \mathbb{R}^2$ of the particle in each frame. Specifically, we take particle coordinate $(x, y) \in \mathbb{R}^2$ as the input feature of the model, and the model analyzes the distribution of particle swarm according to the coordinate information, so as to predict the degree of degradation. The localization accuracy is closely related to the distribution of particles. The localization accuracy can be measured by the diagonal elements $(\sigma_x^2, \sigma_y^2, \sigma_\theta^2) \in \mathbb{R}^{1 \times 3}$ of the pose covariance matrix. These three elements are essentially variances, representing the accuracy of the particle coordinates (x, y) and the orientation angle θ . Theoretically, the smaller the values of $(\sigma_x^2, \sigma_y^2, \sigma_\theta^2)$, the higher the localization accuracy. In this case, the particles are concentrated near the ground truth of the

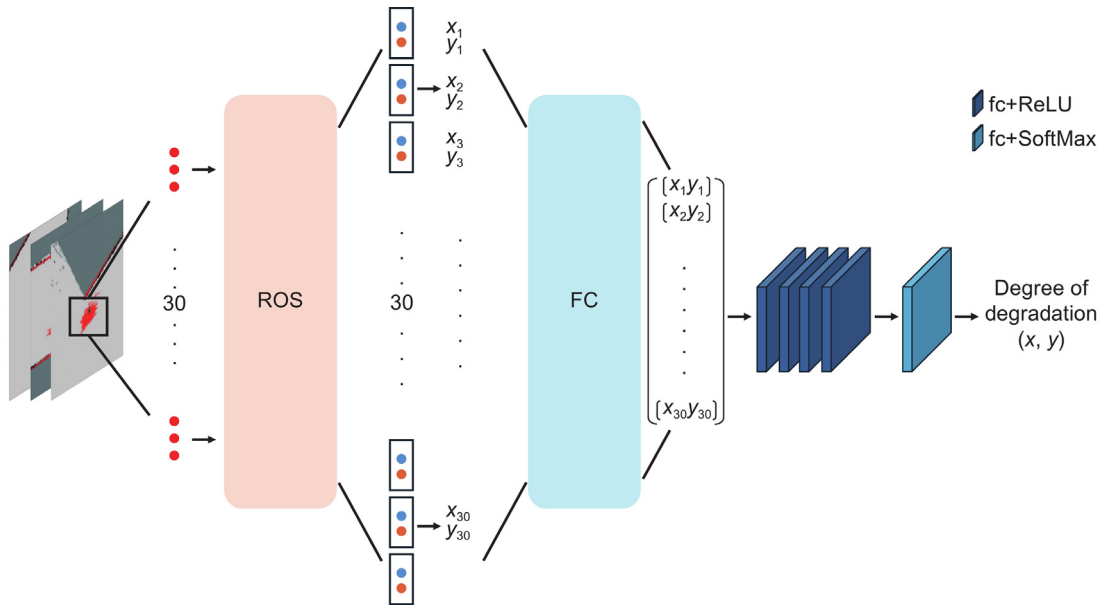


Fig. 2. Schematic diagram of the model architecture.

pose, showing a more concentrated distribution. Conversely, the accuracy of localization will decline when degradation occurs, the values of $(\sigma_x^2, \sigma_y^2, \sigma_\theta^2)$ become larger, causing the particles to diverge and show a more dispersed distribution. At this point, the model should be able to detect particle divergence and determine that degradation occurs.

Subsequently, we remove any data entries that contained missing values. To mitigate the influence of outliers on the distribution of the particle swarms, we employ Z-SCORE normalization. This statistical method standardizes the data by subtracting the mean and dividing by the standard deviation for each feature, effectively reducing the impact of extreme values. Finally, we add different labels to particle coordinate data (x, y) collected in degraded environment and non-degraded environment respectively.

The entire dataset consists of 587 degraded and 562 non-degraded data, each consisting of all particle coordinates of the current frame with a label representing whether degraded or not, and the ratio of training set to test set is 7:3. The network model is able to perform supervised learning of the distributional characteristics of the particles in different scenarios with this dataset dedicated to degradation detection.

4.2. Model architecture

In this article, we present a neural network model based on Multilayer Perceptron (MLP) designed to predict the degree of SLAM degradation by analyzing particle coordinate data, as depicted in Fig. 2. The input consists of 60 values representing the 2D coordinates of 30 particles after the spreading process. The initial fully connected layer accurately reconstructs these 60 input features into the 2D coordinates of the 30 particles using a pre-set weight matrix, merging them two by two. This reconstruction is crucial, enabling the network to process input data as 2D coordinates, thus facilitating subsequent feature extraction and learning.

The particle coordinate data then pass through a series of fully connected layers interspersed with ReLU activation functions. Each fully connected layer performs a linear transformation on the previous output, progressively reducing the dimensionality from 30 to 2. The ReLU activation function introduces non-linearity, allowing the model to capture complex patterns in the

input data. This dimensionality reduction not only minimizes the parameters to mitigate overfitting but also focuses the model on the salient characteristics of particle distribution.

Ultimately, the model outputs the degradation degree through a (4, 2) fully connected layer followed by a SoftMax activation function, yielding a two-dimensional vector. The first dimension represents the non-degradation degree, while the second dimension indicates the degradation degree.

During feature transformation, the model first reorganizes the 60 input spreading coordinate values into 30 particle 2D coordinates via the initial fully connected layer and weight matrix. This step preserves the original data structural information and provides a clear coordinate representation for further feature extraction. Subsequently, the data undergo multiple fully connected layers and ReLU activation functions, abstracting and transforming the input features. As the network progresses, the feature dimensionality decreases, and the model extracts higher-level, discriminative features from the raw coordinate data. These multi-layered features are then used to predict the SLAM degradation level, producing a two-dimensional vector that accurately reflects the SLAM current state. The first dimension of the vector indicates the degree of degradation, while the second dimension signifies the degree of non-degradation. When the value of the first dimension exceeds that of the second dimension, the system is deemed to be in a degraded state.

The feature transformation process is meticulously designed to enable the model to effectively learn and predict the SLAM degradation level from complex particle coordinate data, offering a novel approach for degradation detection in SLAM systems.

5. Adaptive compensation strategies for sensors

After detecting the degradation state of the SLAM system, corresponding strategies must be adopted to mitigate the impact of the degradation phenomenon on the SLAM performance, so as to enhance the accuracy of localization and the quality of maps.

Pose optimization, as a key aspect of SLAM, finely adjusts the pose by fusing multiple sensor data to find the globally optimal pose, which in turn enhances the overall performance of SLAM. Especially in degraded environments, it is necessary to compensate for the pose optimization process by using odometry

Algorithm 2: Global position optimization based on the degree of degradation

Input: Initial position p_o , the maps carried by the particles m , lidar beam b , degree of degradation c , noise n , difference in odometer position δ

Output: Optimized position p_n and score s_n of optimized particles

- 1 **for** all particles **do**
- 2 Update the position using the motion model and add noise n_x, n_y, n_θ :

$$\begin{cases} \delta = (\delta_x, \delta_y, \delta_\theta) = p_{new} - p_{old} \\ n_x = + \text{Gaussian}(|\delta_x| + |\delta_\theta| + |\delta_y|) \\ n_y = + \text{Gaussian}(|\delta_y| + |\delta_\theta| + |\delta_x|) \\ n_\theta = + \text{Gaussian}(|\delta_\theta| + \sqrt{\delta_x^2 + \delta_y^2}) \end{cases}$$
- 3 Calculate the difference ψ between the predicted hit point of the laser beam p_{hit} and the mean value of its corresponding point e_{mean} in the map:

$$\psi = p_{hit} - e_{mean}$$
- 4 Calculate the initial score s_i of particles:

$$s_i = (s_i + e^{(-\psi^2/\mu)})$$
- 5 **while** ScanMatching **do**
- 6 **if** motion is translation **then**
- 7 Move δ_i distance in the corresponding direction
- 8 **else**
- 9 Rotate δ_a in the corresponding direction
- 10 Update the score s_i of particles based on sensor impact factor φ :

$$\begin{cases} \varphi = -e^{0.7c} + e^{0.35} + 1 \\ s_i = (s_i + e^{(-\psi^2/\mu)}) \cdot \varphi \end{cases}$$
- 11 Assign the position with the highest score to p_n
- 12 **return** p_n

data due to the insufficient constraints in the ranging information provided by Lidar.

We design an adaptive compensation strategy for sensors to improve the pose optimization process, the flow of which is shown in Algorithm 2. In the pose optimization process, the position of robot is firstly adjusted initially using information from the odometer based on the motion model, and then further optimized by scanmatch using lidar data based on the observation model. During the scanmatch process of the lidar, the robot is fine-tuned by different degrees of translation and rotation, and the one with the highest matching score is selected as the optimal pose of the robot among the many fine-tuned positions. Then an influence factor φ is introduced to control the dependence on different sensors, which adjusts the matching score in each pose optimization process to determine whether the optimized pose of lidar or the initial pose of odometer is preferred. As the degradation confidence increases, the fraction of matches per scan decreases, so the pose given by the odometer is favored. This approach reduces the effect of differences in sensor properties on the SLAM system by adaptively adjusting the dependency weights of different sensors.

6. Experiments

This section demonstrates the optimality of the model through ablation experiments and verifies the anti-degradation of the system in different environments through simulations and real-world experiments. We use a laptop computer with an Intel(R)

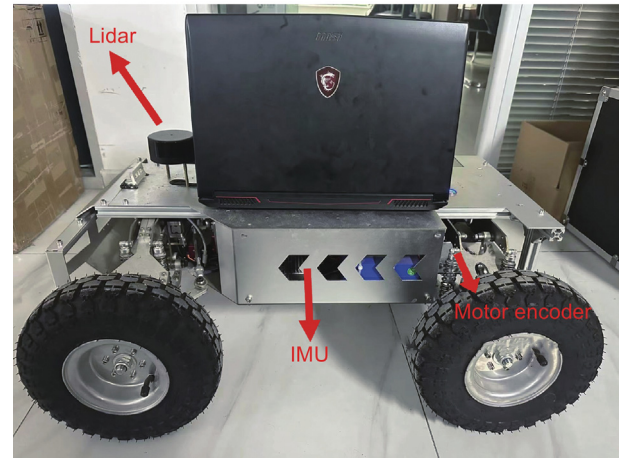


Fig. 3. Experimental platforms.

Core(TM) i7-7700HQ CPU @ 2.80 GHz, an NVIDIA GeForce GTX 1060 graphics card, and 16G of operating memory to remotely maneuver an Ackerman wheeled robot for experiments, as shown in Fig. 3. The robot is equipped with an LSLidar M10P, which is powered by an ARM Cortex-A72 64-bit CPU and a Broadcom VideoCore VI GPU. In simulations, we adopt a computer with ubuntu 20.04 system with a CPU configuration of 13th Gen Intel(R) Core(TM) i7-13700K, GPU of NVIDIA GeForce RTX 2060 super, and 16G of RAM running on the computer.

The experimental framework is depicted in Fig. 4. In this setup, we control the robot chassis using a laptop. The SLAM node publishes the particle coordinates to the MLP node via ROS. After making projections, the model sends the degradation level back to the SLAM node, the SLAM system then uses the information from the different sensors for adaptive constraint compensation.

6.1. Ablation experiments

We demonstrate the effectiveness of the proposed methods with ablation experiments. We choose different pairings of optimizers and loss functions to find the best combination, where the better performing choices are Binary CrossEntropy Loss (BCE), BCEWithLogitsLoss, Adaptive Moment Estimation (Adam), Root Mean Square Propagation (RMSprop). In addition, we have designed the CNN consisting of two 1D convolutional layers, plus a maximum pooling layer, and two fully connected layers. Then we compare the performance of the two models, CNN and MLP, to prove that our choice is optimal.

The results of the experiment are presented in Table 1, where Average Precision (AP) reflects the average of precision and recall of the model, F1 Score reflects the balanced performance of the model between precision and recall, Parameters reflects the complexity of the model, and Latency reflects the computational efficiency and speed of the model, respectively. It can be seen that CNN and MLP models perform similarly on AP and F1 Score, but the complexity of CNN is several orders of magnitude higher than that of MLP and the computational speed is slow. Considering the demanding real-time requirements of SLAM, we finally choose the MLP model with BCELoss and Adam.

6.2. Performance of anti-degradation systems in simulation environments

We create simulation environments through gazebo to verify the performance of the system in various scenarios. A two-wheel

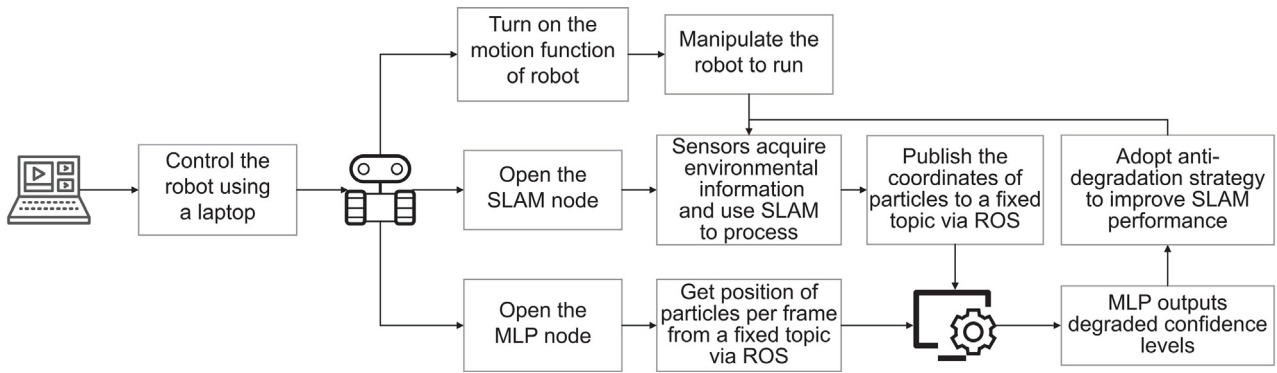


Fig. 4. Experimental procedure.

Table 1
Results of ablation experiments.

Model	Loss	Optimizer	AP (%)	F1 Score	Parameters	Latency (ms)
MLP	BCELoss	Adam	95.07	0.9199	1608	17
MLP	BCELoss	RMSprop	94.04	0.8881	1608	32
MLP	BCEWithLogitsLoss	Adam	91.41	0.8925	1608	28
MLP	BCEWithLogitsLoss	RMSprop	90.45	0.8846	1608	33
CNN	BCELoss	Adam	95.15	0.8955	135 618	33
CNN	BCELoss	RMSprop	94.15	0.9085	135 618	27
CNN	BCEWithLogitsLoss	Adam	94.04	0.9272	135 618	36
CNN	BCEWithLogitsLoss	RMSprop	95.74	0.9145	135 618	47

Table 2
Positioning errors in simulation experiments.

Method	Indoor environment			Circular corridor		
	ATE (m)	Error _x (m)	Error _y (m)	ATE (m)	Error _x (m)	Error _y (m)
Gmapping(120)	0.1067	0.0362	0.0388	1.1766	0.2989	0.3768
Gmapping(100)	0.124	0.0445	0.046	1.283	0.4407	0.5356
Gmapping(80)	0.138	0.059	0.0557	1.8079	0.5624	0.7372
Cartographer [33]	0.0968	0.0423	0.0439	4.7844	5.984	4.779
Ours	0.0877	0.0264	0.022	0.25335	0.093	0.142

differential car equipped with RPLidar is used in the simulation experiment. The scanning range of Lidar is 0.1 m to 30 m, and the scanning frequency is 5.5 Hz. All model configurations are from [34]. Theoretically, the higher the number of particles, the better the performance of the SLAM, but the required computational resources will also rise sharply. We set the particle count of improved gmapping to 80, and conducted a comparative test with gmapping with particle count of 100 and 120 respectively, to highlight that our method can overcome the problem of poor fit of the estimated pose caused by small particle count, and reduce the impact of degradation on SLAM.

In terms of map accuracy, we compare the maps of the improved SLAM (with a particle count of 80) with those of the pre-improved SLAM (with a particle count of 120) and cartographer [33] to perform a qualitative analysis. In terms of localization accuracy, we plot the trajectories given by SLAM and qualitatively analyze them by comparing their fit to groundtruth given by gazebo. In addition, in the experiments, we select three evaluation metrics for quantitative analysis, which are absolute trajectory error (ATE), as shown in Eq. (3), the average of the absolute value of the difference between the localization trajectory and the true value in the x -direction, $Error_x$, and the average of the absolute value of the difference in the y -direction, $Error_y$.

$$ATE_{trans} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|T_{gt,i} - T_{esti,i}\|^2} \quad (3)$$

where N represents the number of sampling points, $T_{gt,i}$ represents the true value of the coordinates of the sampling points, and $T_{esti,i}$ represents the estimate of the coordinates of the sampling points.

Fig. 5(a) is a 10m*10 m indoor scene, which enriches texture features by setting obstacles. Fig. 5(b) is the map constructed by the pre-improved SLAM using 120 particles, (c) is the map constructed by the improved SLAM using 80 particles, (d) is the map given by [33]. It can be observed that the indoor maps generated by both SLAM systems are quite similar and accurately represent the structure of the environment. Furthermore, the trajectories output by the different SLAM systems are close to the groundtruth, as shown in Fig. 6. However, in Table 2, it can be seen that the improved SLAM has a reduction in ATE, with a decrease in error of roughly 36.45% compared to Gmapping(80) and 9.4% compared to [33], and both $Error_x$ and $Error_y$ are also in the lower range, suggesting that the improved SLAM performs well in indoor environments.

Fig. 5(e) is a 40 m*40 m circular corridor which consists of smooth walls with sparse features, from (f), it can be seen that the maps constructed by the pre-improvement SLAM do not present the correct closure and are misaligned at the point where the loopback occurs. The map (h) obtained from [33] has a greater deviation, the length of the vertical corridor is shorter, and the corners are skewed to varying degrees. The improved SLAM is able to construct a correct map of the environment. In Fig. 6, the highest fit between the trajectory of the improved SLAM and the

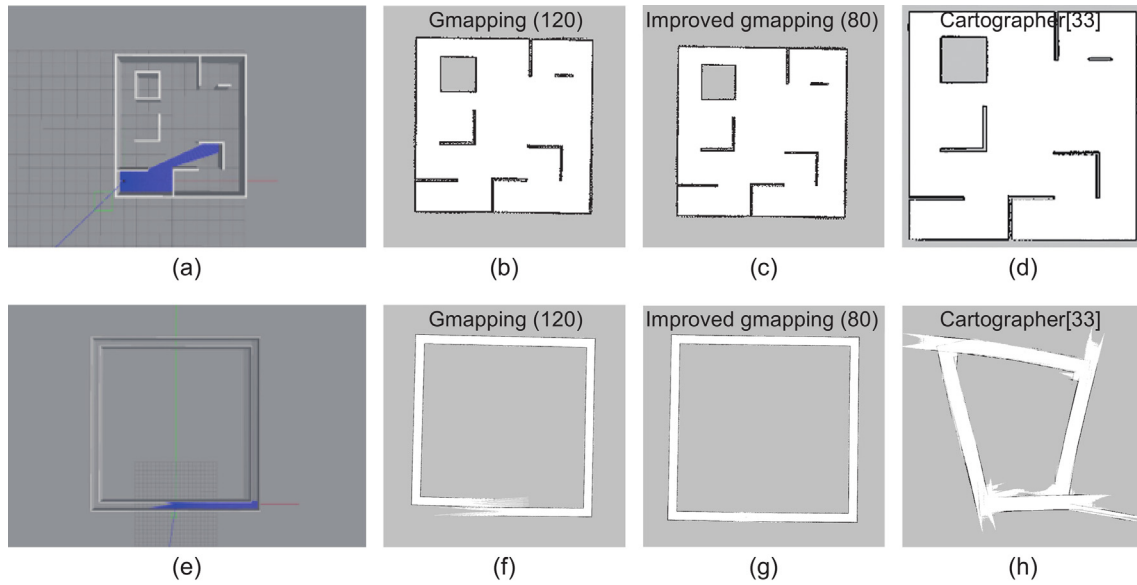


Fig. 5. The truth values of the experimental scenarios and the constructed maps.

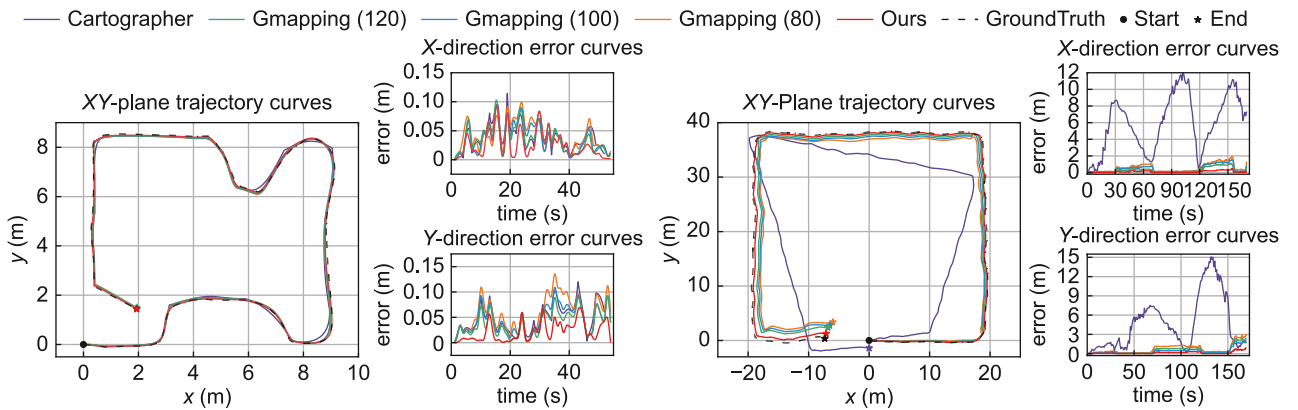


Fig. 6. Evaluation plot of trajectory accuracy for simulation experiments.

groundtruth is observed, the variation curves of both $Error_x$ and $Error_y$ fluctuate in a lower range, while the rest of the trajectories show a larger offset. In Table 2, the improved SLAM performs well in terms of ATE, with a 94.7% improvement in accuracy over [33] and a 78.5% improvement over Gmapping(120).

The above experiments show that the improved SLAM performs well in indoor environments, while the enhancement in degraded environments is more obvious, presenting more accurate localization and map building results. In addition, we record that the average inference time of the model is about 18.7 ms, which cannot affect the real-time performance of SLAM.

6.3. Performance of anti-degradation systems in real-world environments

We use the experimental equipments in Fig. 3 to test the improved SLAM in real scenarios, choosing two long straight corridors with sparse environmental features and evaluating the SLAM through the same steps and metrics as in the simulation experiments, where the groundtruth is derived from manual measurements.

Fig. 7(a) shows a 50m*2.5 m long straight corridor with a vertical corner, where the smooth walls on both sides make the SLAM lack constraints and thus degradation, leading to the skewed map shown in map (b) given by Gmapping(120), while

(c) demonstrates that the improved SLAM is able to construct a roughly accurate map, although still with a small degree of skew. Although the map (d) given by [33] does not appear skew, it can be seen from Table 3 that its localization error on the X-axis is up to 10.78 m, and the corner part also appears double shadow. As shown in the localization trajectory in Fig. 8, the improved SLAM has the highest fit to groundtruth and the most significant decreasing trend in $Error_y$, indicating an improvement in SLAM localization accuracy in the direction of lack of constraints. In Table 3, it can be seen that the improved SLAM reduces the ATE, $Error_x$ and $Error_y$ metrics, and its ATE is reduced by about 38.83% compared to Gmapping(80) and 97.9% compared to [33].

Fig. 7(e) shows a 60m*3 m long straight corridor made of smooth walls with a vertical corner, as depicted in (f) and (g), the map generated by the improved SLAM exhibit a slight skew, yet it represent a significant enhancement over the map produced by Gmapping(80) . As illustrated in Fig. 8, the localization trajectory accuracy of [33] is the worst, especially on the X-axis, and the error is 10.6225 m different from the groundtruth, the fit between the improved SLAM output trajectory and groundtruth is improved. Regarding the quantitative metrics, Table 3 shows that the ATE of the improved SLAM is reduced by approximately 32.04% compared to Gmapping(80) and 94.43% compared to [33], and both $Error_x$ and $Error_y$ are at significantly lower levels.

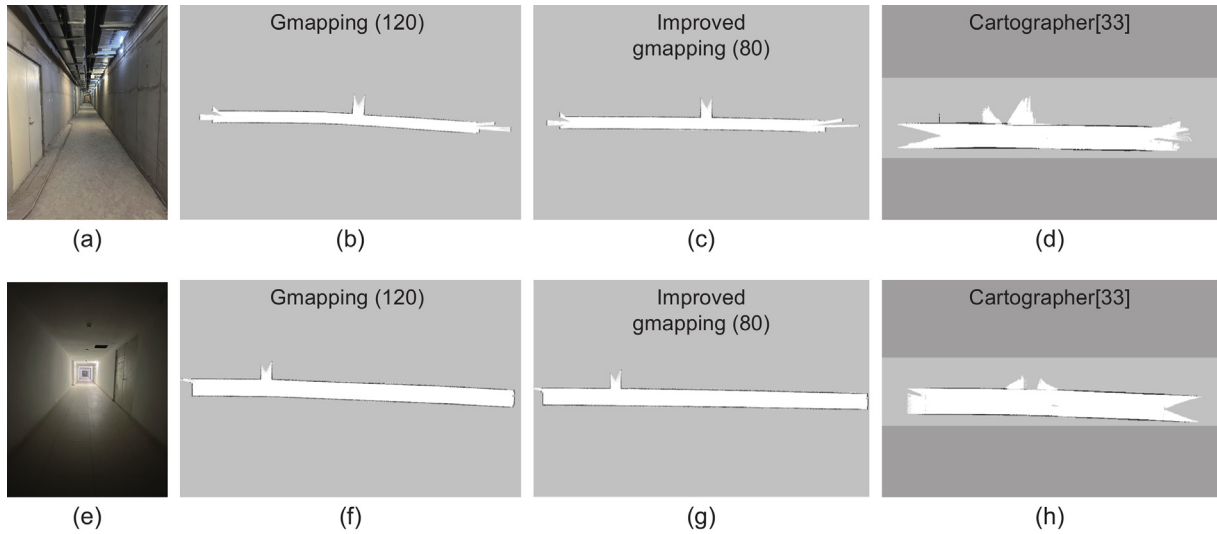


Fig. 7. The truth values of the experimental scenarios and the constructed maps.

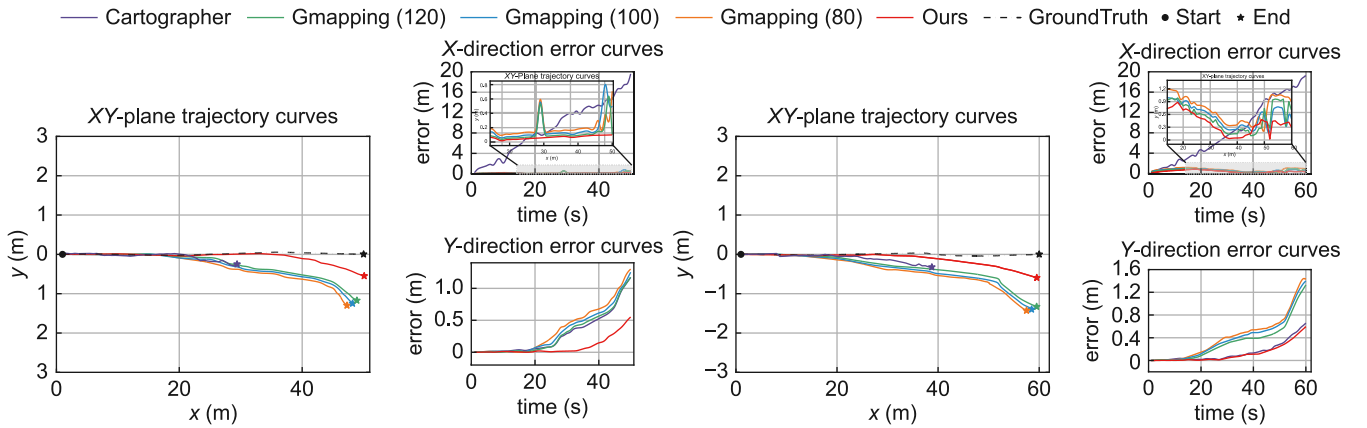


Fig. 8. Evaluation plot of trajectory accuracy for actual experiments.

Table 3
Positioning errors in actual experiments.

Method	Straight corridor A			Straight corridor B		
	ATE (m)	Error _x (m)	Error _y (m)	ATE (m)	Error _x (m)	Error _y (m)
Gmapping(120)	0.655	0.1183	0.593	0.8528	0.5606	0.7325
Gmapping(100)	0.8642	0.1653	0.628	0.956	0.6062	0.7868
Gmapping(80)	0.9564	0.1834	0.664	1.0687	0.7599	0.8228
Cartographer [33]	28.4572	10.78	0.968	13.0365	10.6225	0.885
Ours	0.585	0.0676	0.186	0.72627	0.4069	0.1167

The above experiments show that the improved SLAM performs particularly well in degraded environments, demonstrating better robustness. It is able to accurately detect the degraded state and compensate the SLAM system with information from different sensors, thus improving the performance of SLAM. In the real world experiments, the average inference time of the model is 20.6 ms, which can guarantee the real-time performance of SLAM.

7. Conclusion

In this article, we have proposed a deep learning-based adaptive compensation strategy for sensors in degraded environments. First, we created a dataset dedicated to train a degradation detection model, which contains coordinate data of particle populations with different distributional characteristics, and empowered

the model with degradation detection capabilities through supervised learning. Second, we designed a lightweight network model based on the MLP architecture, which has a short inference time to ensure real-time SLAM and at the same time good accuracy for real-time degradation detection tasks. Finally, after the model detecting degradation, we adaptively compensated the SLAM system with a specific strategy to assign different weights to the sensor information according to the degree of degradation, so as to improve the robustness of SLAM in degraded environments. We demonstrated through simulation experiments and real experiments that the improved SLAM system had good performance in various environments, and the improvement was especially obvious in degraded environments, where localization accuracy improved by up to 97.9% compared to cartographer [33] that relied on parameter adjustment. In the future, we will develop

a complete system to overcome the long-term accumulation of errors in the odometry.

CRedit authorship contribution statement

Yanbin Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Wei Zhang:** Writing – original draft, Funding acquisition, Conceptualization. **Zhiguo Zhang:** Supervision, Funding acquisition. **Xiaogang Shi:** Project administration, Methodology. **Ziruo Li:** Software, Data curation. **Mingming Zhang:** Software, Methodology. **Wenzheng Chi:** Writing – review & editing, Writing – original draft, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Science Foundation of China (62273246), Science and Technology Research Foundation of State Grid Co. Ltd (5700-202318270A-1-1-ZN).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.birob.2025.100235>.

References

- [1] W. Chen, W. Chi, S. Ji, H. Ye, J. Liu, Y. Jia, J. Yu, J. Cheng, A survey of autonomous robots and multi-robot navigation: Perception, planning and collaboration, *Biomim. Intell. Robot.* (2024) 100203.
- [2] G. Grisetti, Improved techniques for grid mapping with rao-blackwellized particle filters, *IEEE Trans. Robot.* 23 (1) (2007) 34–46.
- [3] L. Xu, H. Yin, T. Shi, D. Jiang, B. Huang, EPLF-VINS: Real-time monocular visual-inertial SLAM with efficient point-line flow features, *IEEE Robot. Autom. Lett.* 8 (2) (2023) 752–759, <http://dx.doi.org/10.1109/LRA.2022.3231983>.
- [4] Z. Chen, Z. Miao, M. Liu, C. Wu, Y. Wang, A fast and accurate visual inertial odometry using hybrid point-line features, *IEEE Robot. Autom. Lett.* 9 (12) (2024) 11345–11352, <http://dx.doi.org/10.1109/LRA.2024.3490406>.
- [5] J. Wang, Y. Ren, Z. Li, X. Xie, Z. Chen, T. Shen, H. Liu, K. Wang, USD-SLAM: A universal visual SLAM based on large segmentation model in dynamic environments, *IEEE Robot. Autom. Lett.* 9 (12) (2024) 11810–11817, <http://dx.doi.org/10.1109/TRO.2024.3366815>.
- [6] Y. Ge, L. Zhang, Y. Wu, D. Hu, PIPO-SLAM: Lightweight visual-inertial SLAM with preintegration merging theory and pose-only descriptions of multiple view geometry, *IEEE Trans. Robot.* 40 (2024) 2046–2059, <http://dx.doi.org/10.1109/TRO.2024.3415814>.
- [7] B. Zhang, Y. Dong, Y. Zhao, X. Qi, DynPL-SLAM: A robust stereo visual SLAM system for dynamic scenes using points and lines, *IEEE Trans. Intell. Veh.* (2024) 1–13, <http://dx.doi.org/10.1109/TIV.2024.3415814>.
- [8] H. Xu, P. Liu, X. Chen, S. Shen, D²SLAM: Decentralized and distributed collaborative visual-inertial SLAM system for aerial swarm, *IEEE Trans. Robot.* 40 (2024) 3445–3464, <http://dx.doi.org/10.1109/TRO.2024.3422003>.
- [9] C. Campos, Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam, *IEEE Trans. Robot.* 37 (6) (2021) 1874–1890.
- [10] T. Shan, Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 5135–5142.
- [11] W. Xu, Fast-lio2: Fast direct lidar-inertial odometry, *IEEE Trans. Robot.* 38 (4) (2022) 2053–2073.
- [12] H. Wang, F-loam: Fast lidar odometry and mapping, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 4390–4396.
- [13] C. Zheng, FAST-LIVO2: Fast, direct LiDAR–Inertial–Visual odometry, *IEEE Trans. Robot.* 41 (2025) 326–346, <http://dx.doi.org/10.1109/TRO.2024.3502198>.
- [14] T. Wen, LIVER: A tightly coupled LiDAR-inertial-visual state estimator with high robustness for underground environments, *IEEE Robot. Autom. Lett.* 9 (3) (2024) 2399–2406, <http://dx.doi.org/10.1109/LRA.2024.3355778>.
- [15] J. Xu, Intermittent VIO-assisted LiDAR SLAM against degeneracy: Recognition and mitigation, *IEEE Trans. Instrum. Meas.* 74 (2025) 1–13, <http://dx.doi.org/10.1109/TIM.2024.3507053>.
- [16] Q.H. Hoang, G.-W. Kim, IMU augment tightly coupled lidar-visual-inertial odometry for agricultural environments, *IEEE Robot. Autom. Lett.* 9 (10) (2024) 8483–8490, <http://dx.doi.org/10.1109/LRA.2024.3440728>.
- [17] J. Lee, R. Komatsu, M. Shinozaki, T. Kitajima, H. Asama, Q. An, A. Yamashita, Switch-SLAM: Switching-based LiDAR-inertial-visual SLAM for degenerate environments, *IEEE Robot. Autom. Lett.* 9 (8) (2024) 7270–7277, <http://dx.doi.org/10.1109/LRA.2024.3421792>.
- [18] W. Liu, Dloam: Real-time and robust lidar slam system based on cnn in dynamic urban environments, *IEEE Open J. Intell. Transp. Syst.* (2021).
- [19] Q. Zhang, GAN-SLAM: GAN based monocular visual-inertial simultaneous localization and mapping in dark environments, in: 2022 5th International Symposium on Autonomous Systems, ISAS, 2022, pp. 1–6, <http://dx.doi.org/10.1109/ISAS55863.2022.9757267>.
- [20] C. Li, Tlcd: A transformer based loop closure detection for robotic visual slam, in: 2022 International Conference on Advanced Robotics and Mechatronics, ICARM, IEEE, 2022, pp. 261–267.
- [21] X. Lian, DMN-SLAM: Multi-MLPs neural implicit representation SLAM for dynamic environments, *IEEE Access* (2025).
- [22] Y. Li, Q. Xiong, Z. Li, X. Shi, W. Chi, A unimodal degradation detection method for particle filter-based slam algorithms, *Procedia Comput. Sci.* 250 (2024) 265–273.
- [23] S. Rahman, A. Quattrini Li, I. Rekleitis, SVIn2: A multi-sensor fusion-based underwater SLAM system, *Int. J. Robot. Res.* 41 (11–12) (2022) 1022–1042.
- [24] Y. Jia, H. Luo, F. Zhao, G. Jiang, Y. Li, J. Yan, Z. Jiang, Z. Wang, Lvio-fusion: A self-adaptive multi-sensor fusion slam framework using actor-critic method, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 286–293.
- [25] M. Frosi, M. Matteucci, MCS-SLAM: Multi-cues multi-sensors fusion SLAM, in: 2022 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2022, pp. 1423–1429.
- [26] Y. Li, W. Yang, D. Lin, Q. Wang, Z. Cui, X. Qin, AVM-SLAM: Semantic visual SLAM with multi-sensor fusion in a bird's eye view for automated valet parking, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2024, pp. 7937–7943.
- [27] Y. Wang, H. Ma, mvil-fusion: Monocular visual-inertial-lidar simultaneous localization and mapping in challenging environments, *IEEE Robot. Autom. Lett.* 8 (2) (2022) 504–511.
- [28] X. Tao, B. Zhu, S. Xuan, J. Zhao, H. Jiang, J. Du, W. Deng, A multi-sensor fusion positioning strategy for intelligent vehicles using global pose graph optimization, *IEEE Trans. Veh. Technol.* 71 (3) (2021) 2614–2627.
- [29] R. Gao, Y. Li, B. Li, G. Li, FELC-SLAM: feature extraction and loop closure optimized lidar SLAM system, *Meas. Sci. Technol.* 35 (11) (2024) 115112.
- [30] X. Lin, W. Yao, T. Yan, A LiDAR SLAM algorithm considering dynamic extraction of feature points in underground coal mine, *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* 48 (2024) 659–664.
- [31] T.-C. Tsai, C.-C. Peng, Ground segmentation based point cloud feature extraction for 3D LiDAR SLAM enhancement, *Measurement* 236 (2024) 114890.
- [32] Z. Xiao, S. Li, SL-SLAM: A robust visual-inertial SLAM based deep feature extraction and matching, 2024, arXiv preprint [arXiv:2405.03413](https://arxiv.org/abs/2405.03413).
- [33] W. Hess, Real-time loop closure in 2D LiDAR SLAM, in: 2016 IEEE International Conference on Robotics and Automation, ICRA, 2016, pp. 1271–1278, <http://dx.doi.org/10.1109/ICRA.2016.7487258>.
- [34] Stanford Artificial Intelligence Laboratory, et al., Robotic operating system, 2018, URL <https://www.ros.org>.