Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **1**, February **2025**

Automatic Positioning of Intelligent Mobile Robot Based on Improved Front-End Scanning Matching

Hong Wang*

Department of Computer and Software Engineering
Shandong College of Electronic Technology, Jinan 250200, P.R. China
wh02202024@163.com

Yi-Gui Wang

School of Computer Science and Technology
Shandong Jianzhu University, Jinan 250101, P.R. China
35214763@qq.com

Jie Zhang

College of Agricultural Science and Technology
Yana Royal Polytechnic University, Chiang Mai 50000, Thailand
wh0511@21cn.com

*Corresponding author: Hong Wang

Received May 27, 2024, revised September 3, 2024, accepted November 29, 2024.

Abstract. *Traditional autonomous navigation methods for mobile robots mainly rely on geometric feature-based LiDAR scan-matching algorithms, but in complex environments, this method is often affected due to the presence of moving objects, occlusions, and other interfering factors, resulting in a decrease in positioning accuracy. With the growing demand for robotics applications in logistics, security, exploration and other fields, the need for robust autonomous navigation and high-precision mapping in highly dynamic and complex environments is becoming more and more urgent. To solve this problem, an improved front-end scan matching algorithm based on Frequency Modulated Continuous Wave (FMCW) LiDAR is proposed in this paper. Firstly, by using the target Doppler velocity information provided by the FMCW LiDAR, we design a novel point cloud segmentation algorithm based on velocity clustering, which is able to effectively distinguish between stationary and moving objects, and avoid the dynamic interference affecting the position estimation. Secondly, we introduce the Gaussian Mixture Model Sampling Consistency (GMMSC) algorithm, which is more robust to reject the mis-matched pairs in the scanning matching process and improve the alignment accuracy. Finally, based on the residual high-quality matched pairs, we combine the classical ICP algorithm with the robust kernel function to further enhance the stability of the position estimation in the case of occlusion and local mismatch. The experimental results show that the proposed improved algorithm significantly improves the mapping capability of mobile robots in complex environments compared with the existing techniques. The average relative error of the improved front-end scanning matching algorithm is reduced by 40.6 % compared with the pre-improved one.*

Keywords: Mobile robot; FMCW LiDAR; Scan matching; Point cloud segmentation; False match rejection

1. **Introduction.** With the continuous development of artificial intelligence, sensors and robotics, mobile robots have gradually come out of the laboratory and have been widely used in more and more fields. In logistics and distribution [1, 2, 3], mobile robots can

autonomously complete the handling and sorting of goods in warehouses, terminals and other scenarios with high efficiency, which greatly improves the operational efficiency. In the field of security [4, 5], unmanned robots can replace manual patrolling and monitoring, providing 7 × 24-hour uninterrupted coverage to safeguard public places. In exploration [6, 7], a variety of specialised mobile robots can enter narrow, treacherous or high-risk environments that are difficult for personnel to reach to perform exploration and sampling tasks. In addition, home service robots [8], assistive robots [9] and other fields also have a broad application prospect. The popularity of mobile robots will not only improve work efficiency and safety, but also promote the overall improvement of the productivity of the whole society.

Autonomous navigation and map building capability of mobile robots is one of the key technologies to achieve unmanned operation [10], which has a wide range of demands in various application areas, such as logistics and distribution, security patrol, and exploration operations. In order to achieve high-precision autonomous localisation and mapping, mobile robots need to fuse a variety of sensor information, of which LiDAR scanning matching is one of the most commonly used and effective methods [11]. Traditional LiDAR scanning matching algorithms mainly rely on the geometric features of the point cloud for alignment, but in complex environments, due to the presence of interference factors such as occlusion and moving objects, this geometric feature-based method is often affected, resulting in a decrease in positioning accuracy.

In addition, compared to the traditional pulsed LiDAR, Frequency Modulated Continuous Wave (FMCW) LiDAR can not only accurately measure the target distance, but also obtain the target radial velocity information through Doppler frequency shift [12]. This unique velocity measurement capability gives FMCW LiDAR a natural advantage in detecting and tracking moving targets, which helps to improve the perception and map building accuracy of mobile robots in dynamic and complex environments. At the same time, FMCW technology can provide higher ranging resolution and anti-interference ability, which makes the performance of LiDAR better in narrow space and occluded environment [13]. Therefore, fully exploiting the characteristics of FMCW LiDAR will be expected to promote the new development of autonomous navigation technology for mobile robots. The main research objective of this paper is to propose an improved front-end scan matching algorithm based on FMCW LiDAR to enhance the autonomous localisation and map building capabilities of mobile robots in dynamic and complex environments. Through the improvement measures, we expect to significantly enhance the autonomous navigation performance of mobile robots in complex environments, and lay a solid technical foundation for their applications in logistics, security, exploration and other fields.

1.1. **Related work.** For the problem of autonomous localisation and map building of mobile robots in complex environments, researchers have proposed a variety of improved algorithms, such as introducing machine learning techniques to improve the robustness of data correlation, optimising the position estimation using semantic a priori constraints, and fusing the Doppler velocity information from the FMCW LiDAR to detect the moving objects.

Kostavelis and Gasteratos [14] proposed a 3D LiDAR-based semantic KITTI dataset for evaluating object recognition and motion estimation performance of mobile robots in complex environments. The dataset contains rich scene annotations and provides a good benchmark for algorithm evaluation. However, the scenes in this dataset are still limited to outdoor environments such as streets and car parks, and may not be comprehensive enough for more complex indoor dynamic scenes. Wang et al. [15] designed a novel machine learning-based LiDAR motion compensation algorithm to correct point cloud aberrations by predicting robot motion during scanning. The method significantly improves the accuracy of LiDAR image construction under high-speed motion. However, this algorithm requires high training data, and the generalisation performance may suffer in the absence of sufficient motion data. Park et al. [16] proposed a LiDAR scan matching algorithm based on Bayesian thinking, which improves the robustness of the alignment process by taking into account the non-Gaussian characteristics of the measurement noise. The results show that this algorithm improves the positioning accuracy in the presence of partial occlusion and moving objects. Wong et al. [17] proposed an adaptive Iterative Closest Point (ICP) improvement method based on enhancement learning to address the poor matching effect of the traditional ICP algorithm in smooth regions and unstructured environments. Through the reward mechanism, this algorithm can automatically adjust the data association weights, which effectively improves the matching accuracy in the region with inconspicuous local features. However, this algorithm requires a large amount of training data and computational resources, and is difficult to deploy and migrate. Su et al. [18] proposed a dynamic object detection and removal algorithm based on deep learning for the problem of SLAM system's localisation offset in complex environments with dynamic objects. Using recurrent neural networks to process the point cloud sequence, the interference of dynamic objects on the position estimation can be effectively removed. However, this algorithm needs to process each pair of point cloud frames in sequence, which is computationally inefficient and may be a bottleneck in scenes with high real-time requirements. Jin et al. [19] designed a dynamic point cloud segmentation method using the velocity information provided by FMCW LiDAR. By effectively separating the static background and moving objects, the accuracy of LiDAR odometry and map building is substantially improved. However, the segmentation effect of this algorithm may be affected for targets moving at low or uniform speeds. Renaut et al. [20] proposed a LiDAR position estimation algorithm based on Normal Distribution Transform (NDT). This algorithm suppresses the effects of occlusion and moving objects by introducing Gaussian kernel weights, which improves the robustness of LiDAR scan matching in the environment. However, this algorithm relies on accurate motion and noise models and is more sensitive to changes in model assumptions.

1.2. **Motivation and contribution.** Although the above methods have improved the perception and mapping ability of mobile robots to a certain extent, there are still some deficiencies in real-time, computational efficiency and model generalisation, which are difficult to fully meet the demands of highly dynamic and complex practical application scenarios. Aiming at the shortcomings of the existing techniques, an improved scan-matching algorithm that integrates FMCW LiDAR Doppler velocity measurements, Gaussian mixture model mis-matching rejection and robust ICP-based position estimation is proposed, aiming to further enhance the autonomous navigation and mapping capabilities of mobile robots in dynamic and complex environments. The main innovations and contributions of this work include:

(1) By utilising the target radial velocity information measured by FMCW LiDAR, we propose a novel velocity clustering-based point cloud segmentation algorithm, which is able to effectively differentiate between stationary and moving objects, thus avoiding the influence of dynamic interference on the position estimation and map construction.

(2) To address the problem of mis-matching in the scanning matching process, we introduce the Gaussian Mixture Model Sampling Consistency (GMMSC) algorithm, which eliminates the mis-matched pairs in a more robust way by fitting Gaussian mixture models to the matching residuals, thus improving the quality of alignment and the accuracy of the position estimation.

(3) On the basis of the above point cloud segmentation and mis-match rejection, we apply the classical ICP algorithm to the residual high-quality matched point pairs, and introduce a robust kernel function to further improve the stability of the position estimation in the presence of occlusion and local mismatch.

Some existing algorithms introduce complex models or deep learning techniques to improve accuracy, but this leads to a significant increase in the computational complexity of this algorithm, which makes it difficult to meet the real-time requirements. The improved scanning matching algorithm proposed in this paper maintains good computational efficiency while improving accuracy, and is suitable for deployment on resource-constrained mobile platforms. Some algorithms are built on scene-specific assumptions with poor generalisation ability. The method in this paper is mainly based on the geometric and motion features of the point cloud, with fewer a priori assumptions on the scene, which can be better generalised to a wider range of complex environments.

2. Analysis of relevant principles.

2.1. FMCW LIDAR measurement principle.

FMCW LIDAR, as an advanced sensor, has attracted a lot of attention because of its ability to provide both distance and speed information. The FMCW LIDAR measurement principle uses a laser transmitter to emit a FMCW, and determines the distance between the target object and the radar by using the difference in the frequency of the laser signals reflected back from the target object. distance from the radar. When the laser beam is emitted, the frequency changes continuously with time, and the frequency of the signal reflected back from the target also changes with time. By measuring the difference of the frequency change, the distance between the radar and the target can be calculated, so as to realise range measurement. Simply put, FMCW is based on transmitting a continuously modulated laser signal and receiving the reflected signal back, and obtaining the distance and speed information of the target by calculating the frequency difference between the transmitted signal and the returned signal (i.e. beat frequency signal).

LIDAR emits a continuous wave signal with a time-dependent frequency, usually in a triangular or sawtooth wave FM mode.

$$f_t(t) = f_0 + k(t) \tag{1}$$

where f_0 is the initial frequency and $k(t)$ is the frequency modulation term over time.

The transmitted signal encounters the target object and is reflected back. The receiver collects the reflected signal and mixes it locally with the transmitted signal to produce a beat signal. Let the frequency of the transmitted signal be $f_t(t)$, then the frequency of the received signal be $f_r(t)$ and the frequency of the beat signal be $f_d = |f_t(t) - f_r(t)|$.

$$f_r(t) = f_0 + k(t - \tau) \tag{2}$$

where τ is the propagation delay of the signal to and from the target.

The distance and speed information of the target can be obtained by calculating the frequency of the beat signal. The distance can be determined by calculating the phase difference of the beat frequency signal.

$$d = \frac{c \cdot \tau}{2 \cdot f_0} \tag{3}$$

where c is the speed of light and d is the distance of the LiDAR from the target.

Velocity information can be obtained through the Doppler shift. When the target object has a velocity v with respect to the LIDAR, a Doppler shift Δf_d is generated, which is related to the velocity as $\Delta f_d = \frac{2v}{\lambda}$, where λ is the wavelength of the laser.

$$v = \frac{\lambda \cdot \Delta f_d}{2} \tag{4}$$

2.2. **Principle of Laser SLAM Algorithm.** Laser SLAM (Simultaneous Localisation and Mapping) algorithm obtains the map information of the environment through LIDAR sensor and locates the robot's position in real time based on it, so as to achieve the tasks of simultaneous localization and map building [21, 22]. The principle is that the robot scans the surrounding environment through the carried LiDAR, and builds a map based on the distance data and angle information acquired by the LiDAR; meanwhile, the robot estimates its position in real time based on its own motion information and the feature point information acquired by the LiDAR. The laser SLAM algorithm mainly includes front-end data processing (e.g. feature extraction, scan matching), back-end data processing (attitude estimation, optimisation), as well as alignment, closed-loop detection and other modules, and through the collaborative work of these modules, it achieves the robot's autonomous localisation and map construction in an unknown environment.

Scan matching at the front end is the main method for localisation of the robot. Compared with the method of using odometers and IMUs to calculate the position, the scanning matching method has a higher accuracy. The basic idea of scanning matching is to align the target frame with the reference frame, and the rotations and translations required in the alignment process are the relative positions between the two frames. The position of the reference frame is known and the position of the target frame can be obtained from the relative position [23]. In the local localisation of SLAM systems, IMUs or odometers are often used first to provide a priori positions for scan matching, and then the target positions are solved iteratively using the scan matching method and the constructed nonlinear least squares function. Currently, the more commonly used scan matching methods are ICP and scan-to-submap in Correlative Scan Matching (CSM).

3. Analysis of the motion process of mobile robots.

3.1. **Mobile robot positional projection.** Positioning is a method used to express the position and attitude of a mobile robot. Since the object of this paper is an indoor planar mobile robot, its position in the z-axis direction generally does not change, and its attitude only exists in the angle of rotation around the z-axis. Therefore, in the world coordinate system W, the position p of a planar mobile robot can be expressed as (x, y, θ), where x, y represent the absolute position coordinates of the robot in the world coordinate system, and θ represents the robot's position in the world coordinate system, i.e., the angle of rotation around the z-axis [23, 24]. The robot's coordinate system follows the right-hand rule. For the attitude angle θ, counterclockwise rotation is positive and clockwise rotation is negative.

The position of the robot at a certain moment in the W system cannot be obtained directly, but needs to be calculated based on the position of the previous moment, as shown in Figure 1. Assuming that the coordinates x_B of the vector x in the B system are (dx, dy), I need to solve for the coordinates x_W of the vector x in the W system.

$$x_B = \begin{bmatrix} dx \\ dy \end{bmatrix} = dx \cdot OX' + dy \cdot OY' \tag{5}$$

$$OX' = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{6}$$

$$OY' = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \tag{7}$$

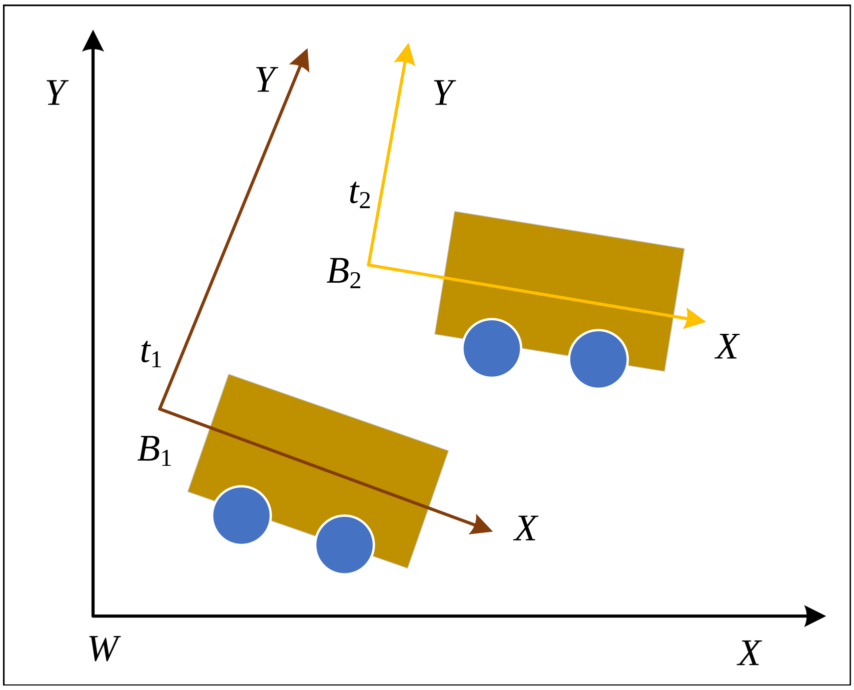

Figure 1. The robot's position at different moments in time.

It can be shown that the coordinate expression of the vector x in the W system is:

$$x_W = dx \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} + dy \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \tag{8}$$

3.2. **Two-wheel differential kinematic model.** The kinematic chassis control for the ROS mobile robot used in this paper is an Ackermann structure, as shown in Figure 2. The Ackermann structure is well known for its simple and efficient steering system, which allows the robot to maintain pure rolling contact between the wheels and the ground during cornering, thus reducing friction and improving manoeuvrability [25, 26]. In this configuration, the two rear wheels are driven by motors with rotational feedback from encoders for precise speed and direction control. The two front wheels, on the other hand, are controlled by servos and are responsible for steering the robot. This design allows the robot to steer flexibly within a small turning radius, which is ideal for navigating in narrow or crowded spaces.

Let the number of light-transmitting areas possessed by the code disc be k, and the number of pulses generated by the photoelectric encoder per unit time be n, then the linear velocity of the wheel v and the distance Δs travelled per unit time are:

$$v = \Delta s = \frac{2\pi r_c n}{k} \tag{9}$$

Based on the above analysis, the linear velocities v_R and v_L of the right and left rear wheels can be obtained. The radii of the rear wheels, r_R and r_L, can be obtained directly, so the angular velocities of the rear wheels can also be obtained, and are expressed as w_R and w_L, respectively. The distance of the wheels from the centre of the chassis, d, and the distance between the wheels, b, can also be obtained directly. The relationship between b and d is as follows:

$$b = 2 \cdot d \tag{10}$$

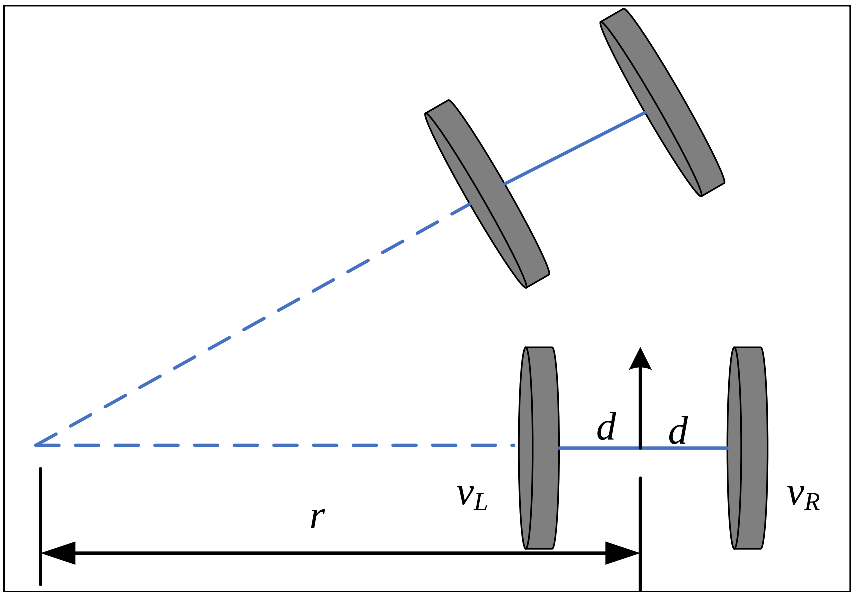

Figure 2. Ackermann steering mechanism.

Since the rotation of the Ackermann structure is in circular motion around a centre, the magnitude of the angular velocity w can be obtained [27, 28] as follows:

$$w = w_r = w_l \tag{11}$$

where w_r, w_l are the angular velocities of the right and left rear wheels relative to the centre of the arc.

According to the velocity relation we can get as follows:

$$w_r = w_l = \frac{v_L}{r - d} = \frac{v_R}{r + d} \tag{12}$$

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \frac{r_L}{2} & \frac{r_R}{2} \\ -\frac{r_L}{b} & \frac{r_R}{b} \end{bmatrix} \begin{bmatrix} w_L \\ w_R \end{bmatrix} \tag{13}$$

3.3. **Rotational representation of mobile robots.** In the positional projection of mobile robots, rotational representations are used to describe the orientation of the robot with respect to some reference coordinate system. Rotation can be expressed through a variety of mathematical tools, including rotation matrices, Euler angles, and quaternions. These representations play a crucial role in SLAM systems because they are able to accurately describe the robot's motion in space.

A rotation matrix is a mathematical tool for describing the rotation of a rigid body in three-dimensional space [29]. For a mobile robot on a two-dimensional plane, the rotation matrix can be reduced to a two-dimensional matrix representing the rotation around the Z-axis. A 2D rotation matrix R can be represented by the angle θ.

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{14}$$

The main advantage of rotated matrices is their simplicity and ease of performing matrix multiplication operations, which makes them very useful in robot kinematics and SLAM.

Euler angles are another way of describing rotation, which describes the rotation of an object relative to a fixed coordinate system through three angles. These three angles are usually expressed as rotation about the X-axis, rotation about the Y-axis and rotation about the Z-axis. However, Euler angles suffer from gimbal deadlock, which may limit their use in some cases.

A quaternion is an extended complex number that provides a rotational representation that avoids the gimbal deadlock problem. A quaternion consists of one real part and three imaginary parts and can be expressed as follows:

$$q = w + xi + yj + zk \tag{15}$$

where w is the real part of the quaternion and x, y, z are the imaginary parts, which satisfy specific rules for multiplying quaternions. Quaternions are more compact than Euler angles in representing rotations and have advantages in interpolation and avoiding gimbal deadlock.

In SLAM systems, the choice of rotational representation is crucial for the performance and stability of this algorithm. Rotation matrices are widely used due to their simplicity and ease of manipulation, whereas quaternions are more appropriate in applications where high accuracy is required and gimbal deadlock is avoided.

4. **Improvement of front-end scanning matching algorithm.**

4.1. **Doppler velocity based FMCW point cloud segmentation.** Traditional LiDAR recognition of dynamic objects usually requires joint two to three frames of timing information to judge, while FMCW LiDAR can not only measure the 3D position information of the point cloud, but also capture the Doppler velocity information of the point cloud. Therefore, this paper proposes a Doppler velocity-based FMCW point cloud segmentation method, which can simultaneously measure the distance and velocity of a target in a dynamic scene to improve the accuracy and efficiency of point cloud data processing. The proposed method mainly utilises Doppler velocity information to distinguish between stationary and moving objects, thus improving the robustness of the SLAM system in the environment.

Firstly, the raw point cloud data are denoised and filtered to eliminate outliers and invalid measurements. Using the frequency modulation characteristics of the FMCW LiDAR, the Doppler velocity information of the reflected objects is extracted by frequency analysis of the received signals. Let the point cloud set P have n data points in the current LiDAR coordinate system $p_i = \left[x_i, y_i, z_i, v_i\right]^T$. Using the measured Doppler shift f_d, the radial velocity $v_{r,i}$ of each point p_i can be estimated.

$$v_{r,j} = \frac{\lambda_0 f_{d,j}}{2} \tag{16}$$

To achieve velocity clustering, points with similar radial velocities are gathered into the same cluster. We use the density-based spatial clustering algorithm DBSCAN, where the distance metric takes into account the velocity differences of the points:

$$dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha(v_{r,j} - v_{r,j})^2} \tag{17}$$

where α is the weighting coefficient for speed differences.

Since static and dynamic objects have different Doppler characteristics, we can further segment the point cloud based on the Doppler characteristics. We can set a threshold value v_{th}. For each point in the point cloud, the point cloud is segmented into stationary and moving point clouds based on the comparison of its Doppler velocity v_i with the preset threshold value v_{th}. Clusters with velocities close to 0 are identified as stationary objects and a stationary object point cloud P_s is constructed.

$$P_s = p_i \mid v_{r,j} < v_{th} \tag{18}$$

Clusters with significantly non-zero velocities are identified as moving objects and a moving object point cloud P_m is constructed.

$$P_m = p_i \mid v_{r,j} \geq v_{th} \tag{19}$$

The key innovation of this algorithm is the use of radial velocity measurements provided by the FMCW LiDAR to cluster the point cloud by velocity, enabling more precise separation of stationary and moving objects in complex environments.

4.2. **False matching rejection of feature points based on GMMSC algorithm.** When performing LiDAR scanning matching, the presence of measurement noise, occlusion and dynamic objects can lead to mis-matches during the matching process. These mis-matches can seriously affect the accuracy of odometer position estimation. To solve this problem, we propose a feature point mismatch rejection method based on the GMMSC algorithm.

In traditional ICP algorithms, the Euclidean distance is usually used as a data term and the measurement noise is assumed to obey a Gaussian distribution. However, in real scenarios, due to the presence of interfering factors such as occlusions, dynamic objects, etc., the noise distribution may take the form of heavy tails or multiple peaks, which do not conform to the assumption of a single Gaussian distribution. The GMMSC algorithm improves the capability of mis-match rejection by using Gaussian Mixture Models (GMMs) to better fit the noise distribution.

Firstly, feature point sets P and Q of the current frame and the reference frame are extracted using feature descriptors with rotational invariance (e.g., FPFH, SHOT, etc.). Initial coarse matching is performed after feature extraction is completed. Coarse matching is performed on the feature point sets P and Q to obtain an initial set of matched point pairs M.

Then, a GMM is fitted to the Euclidean distance d_i of the matched point pairs to obtain the probability density function.

$$p(d \mid \theta) = \sum_{k=1}^{K} \pi_k N\left(d \mid \mu_k, \sigma_k^2\right) \tag{20}$$

where $\theta = \pi_k, \mu_k, \sigma_k$ is a GMM parameter that can be estimated by the Expected Maximum Algorithm (EM).

In order to achieve false match rejection, this paper uses a GMM-based probability density function to compute the probability weight of each matched point pair.

$$w_i = \frac{\pi_k N\left(d_i \mid \mu_k, \sigma_k^2\right)}{\sum_{j=1}^{K} \pi_j N\left(d_i \mid \mu_j, \sigma_j^2\right)} \tag{21}$$

For matching point pairs whose weight w_i is less than a certain threshold w_{th}, they are excluded from the matching set M and treated as mismatches.

Compared with the traditional method based on a single Gaussian distribution, the GMMSC algorithm is more robust and is able to effectively reject false matches introduced by noise, occlusion and dynamic objects, thus improving the accuracy and stability of the LiDAR scan matching. The remaining set of matched point pairs will be subsequently used to estimate the positional transformation of the current frame with respect to the reference frame using the ICP algorithm.

4.3. **Robust ICP-based position estimation.** After the above steps of point cloud segmentation and feature point mismatch rejection, we obtain a set of high-quality matched point pairs. Next, using these matched point pairs, we can use the iterative closest point algorithm (ICP) to accurately estimate the positional changes of the mobile robot. Compared with the traditional ICP, this paper proposes a novel position estimation algorithm based on ICP, and the main innovations are the introduction of false match rejection and robust kernel function, which improves the stability and accuracy in complex environments.

The goal of the ICP algorithm is to find a rigid body transformation that aligns the point cloud of the current frame as well as possible with the point cloud of the reference frame. The transformation usually consists of a rotation R and a translation t. We define the following energy function to measure the distance between the transformed point cloud and the reference point cloud.

$$E(R,t) = \sum_{(p_i,q_j)\in M} w_{ij}\rho(||Rp_i + t - q_j||^2) \tag{22}$$

where M is the set of matched point pairs, w_{ij} is the matching weights computed in Section 4.2, and $\rho(x)$ is the robust kernel function (Huber kernel) used to suppress the effect of outliers.

By minimising the above energy function, we can solve for the optimal transformation T^*.

$$T^* = \arg\min_{R,t} E(R,t) \tag{23}$$

Since the energy function is nonlinear, we use an iterative algorithm to solve it, and gradually converge the current frame to the reference frame by solving the incremental transformation through linearised approximation in each iteration. The specific steps are as follows.

1) Calculate the centre of mass $\bar{p}$ and $\bar{q}$ of the current set of matched point pairs.

2) Construct initial values of the rotation matrix R_0 and the translation vector t_0 such that $R_0p_i + t_0$ is aligned with the centre of mass of q.

3) For each point pair (p_i, q_i), compute the error vector.

$$e_i = q_i - (R_0p_i + t_0) \tag{24}$$

4) Stack the error vectors to construct the Jacobi matrix J.

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial \phi} & \frac{\partial e_1}{\partial t} \\ \vdots & \vdots \\ \frac{\partial e_n}{\partial \phi} & \frac{\partial e_n}{\partial t} \end{bmatrix} \tag{25}$$

5) Calculate the incremental rotation $\Delta\phi$ and translation Δt.

$$\begin{bmatrix} \Delta\phi \\ \Delta t \end{bmatrix} = H^{-1}\sum_{i=1}^{n} w_ie_i \tag{26}$$

where $H = J^TWJ$ and W is the diagonal weight matrix.

6) Update the transformation: $R \leftarrow R(I + \Delta\phi)R_0, t \leftarrow t + \Delta t$.

7) Repeat steps 3-6 until convergence or the maximum number of iterations is reached.

With the above ICP iterations, we then obtain the optimal rigid-body transformation T^* of the current frame with respect to the reference frame, i.e., the change in the position of the mobile robot.

5. Results and analyses of indoor scene experiments.

5.1. Map construction. In order to verify the effectiveness of the proposed improved front-end scanning matching algorithm, a simulated indoor corridor environment is constructed in order to experimentally analyse its map building effect. The environment consists of several rooms and corridors as shown in Figure 3. The modelling software (Gazebo) was used to create the corridor models required for the experiments. The advantage of this simulated environment is the complete control of the environmental parameters and the Ground Truth of the motion trajectory, which allows quantitative analysis of the performance of this algorithm.

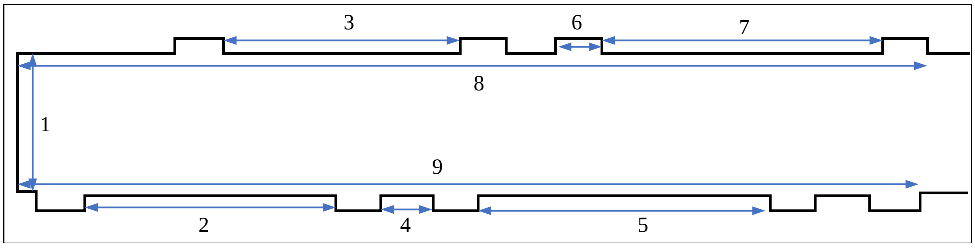

Figure 3. Indoor simulation environment

The above environment simulation map is labelled with 9 features that are more important in the scene, and the real length of the features is obtained through measurements. We verify whether the improved algorithm can improve the accuracy of map building by comparing some details of the map generated before and after the improvement of this algorithm and the accuracy of these 9 features in the map. In building the simulation environment, we set the following parameters for the FMCW LiDAR to simulate the actual sensor used, as shown in Table 1. By simulating a LiDAR with Doppler measurement capability, we can evaluate the effectiveness of the proposed Doppler velocity-based point cloud segmentation algorithm in real scenarios. In the simulated environment, the LiDAR will be mounted on top of a mobile robot and continuously acquire point cloud data during the robot's motion. These data will be fed into our improved scan-matching algorithm for processing to estimate the robot's positional changes and construct a map of the environment.

Table 1. Experimental datasets information

Parametric	**Numerical value**
Wavelength	1550 nm
Scanning angle range	270°
Angular resolution	0.25°
Maximum range	30 m
Distance resolution	2 cm
Ranging error	Gaussian noise ($\mu = 0$, $\sigma = 2$ cm)
Doppler velocity resolution	0.1 m/s
Modulation bandwidth	2 GHz
Sampling rate	10 Hz

5.2. Map building accuracy analysis. The collected point cloud data is passed into our improved front-end scanning matching algorithm for processing, which includes the steps of point cloud segmentation, mismatch rejection, and position estimation. The estimated robot position is compared with the reference trajectory to evaluate this algorithm's localisation accuracy and map building quality in this simulated environment. In order to more accurately compare the map building accuracy of this algorithm before and after the improvement, the relative and absolute errors of the nine feature lengths are calculated, as shown in Table 2 and Table 3, respectively.

Table 2. Absolute and relative errors of feature lengths before improvement

Feature number	**True value/mm**	**Measured value/mm**	**Absolute error/mm**	**Relative error/%**
1	2344	2409	65	2.7730
2	6775	7438	663	9.7860
3	6686	7407	721	10.7837
4	1064	1092	28	2.6316
5	8094	7369	725	8.9573
6	1029	1012	17	1.6521
7	7901	8837	936	11.8466
8	26681	25967	714	2.6761
9	25534	24679	855	3.3485
Average	-	-	-	6.0505

Table 3. Absolute and Relative Errors of Improved Feature Lengths

Feature number	**True value/mm**	**Measured value/mm**	**Absolute error/mm**	**Relative error/%**
1	2344	2366	22	0.9386
2	6775	7081	306	4.5166
3	6686	7115	429	6.4164
4	1064	1039	25	2.3496
5	8094	8607	513	6.3380
6	1029	1047	18	1.7493
7	7901	8411	510	6.4549
8	26681	26266	415	1.5554
9	25534	25021	513	2.0091
Average	-	-	-	3.5919

According to Tables 2 and 3, it can be seen that the improved algorithm is higher than the pre-improved algorithm in terms of accuracy for all nine features. The average relative error of the improved algorithm is reduced by 40.6% compared to the pre-improved algorithm. The errors of features 2, 3, 5, and 7 are more obvious when building the map, which is related to the fact that the environment structure at features 2, 3, 5, and 7 is relatively simple. The single environment structure is more likely to cause the scanning matching to have a large error, which leads to the most prominent relative error in these four areas before the improvement, while the improved algorithm can effectively alleviate this problem. At feature 4 and 6, the relative errors are not obvious before and after the improvement because the environmental structure is relatively more complex. Since the improvement of this algorithm does not involve the vertical orientation of the corridor, the relative error at feature 1 does not change significantly. The relative errors of 9 features before and after improvement are shown in Figure 4.

6. **Conclusion.** In this paper, an improved front-end scanning matching algorithm based on FMCW LiDAR is proposed for the autonomous positioning and mapping of mobile robots in complex environments. The traditional scanning matching algorithm based on LiDAR is difficult to effectively distinguish between stationary objects and moving objects, and dynamic objects will lead to positioning offset and map distortion. In this paper, by using the Doppler velocity measurement of FMCW LiDAR, the detection and segmentation of moving objects are achieved, thus avoiding the influence of dynamic

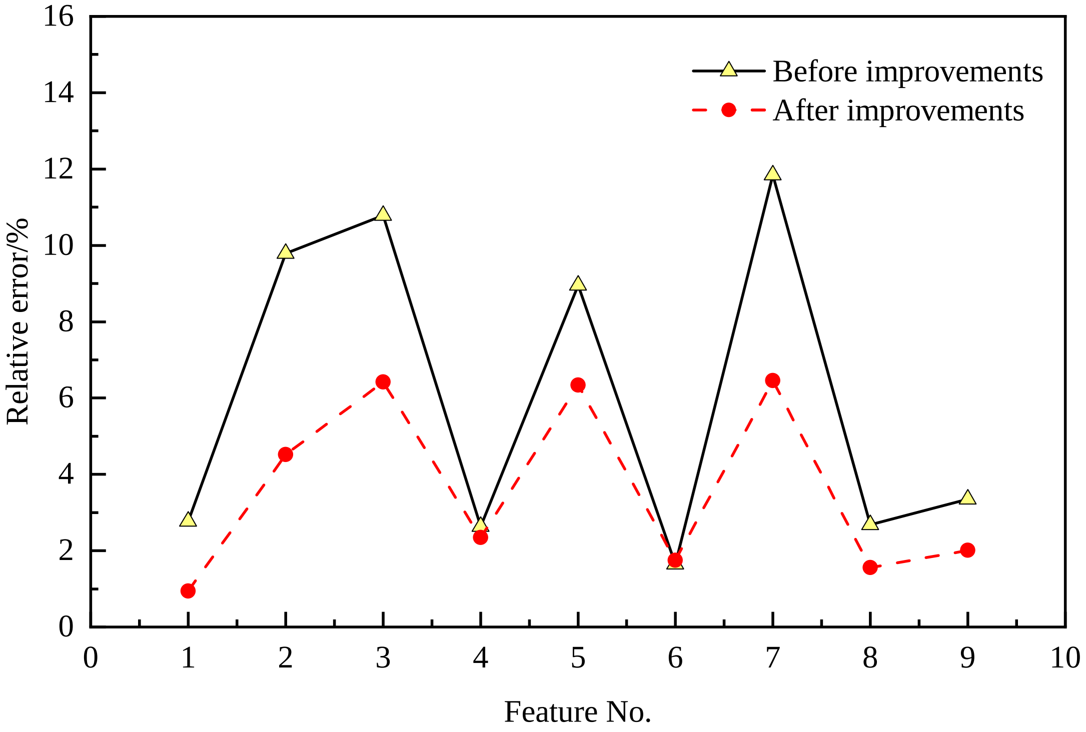

Figure 4. Comparison of relative errors for 9 features before and after improvement

interference on the position estimation and map construction. Due to the measurement noise, occlusion and other factors, the scanning matching process often produces some mismatched pairs of points, and these mismatches will seriously reduce the accuracy and stability of the position estimation. In this paper, the GMMSC algorithm is introduced to eliminate the mismatched points in a more robust way and improve the alignment quality. We comprehensively evaluate the proposed algorithm in both simulated and real scenarios. The experimental results show that our algorithm significantly improves the autonomous navigation and mapping capabilities of mobile robots in complex environments compared with the existing techniques, especially in dynamic object detection, mis-match rejection and robustness. In addition, this algorithm has a broad application prospect because of its strong generalisation ability, which does not depend on specific scene a priori. In the future, we plan to fuse this algorithm with information from other sensory sensors (e.g., vision cameras) to further improve the robustness and accuracy of the system.

Funding. This research was supported by the Shandong vocational education teaching reform research project (Project No.: 2023081).

REFERENCES

[1] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231-249, 1997.

[2] S. G. Tzafestas, "Mobile robot control and navigation: A global overview," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 35-58, 2018.

[3] J. Crowley, "Navigation for an intelligent mobile robot," *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 31-41, 1985.

[4] M. B. Alatise, and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830-39846, 2020.

[5] G. N. DeSouza, and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, 2002.

[6] Y. Nakamura, and A. Sekiguchi, “The chaotic mobile robot,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 898-904, 2001.

[7] K. Zhu, and T. Zhang, “Deep reinforcement learning based mobile robot navigation: A review,” *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674-691, 2021.

[8] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, “A comparative review on mobile robot path planning: Classical or meta-heuristic methods?,” *Annual Reviews in Control*, vol. 50, pp. 233-252, 2020.

[9] L. Wu, X. Huang, J. Cui, C. Liu, and W. Xiao, “Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot,” *Expert Systems with Applications*, vol. 215, 119410, 2023.

[10] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, “A Spectral Convolutional Neural Network Model Based on Adaptive Fick’s Law for Hyperspectral Image Classification,” *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19-46, 2024.

[11] T.-Y. Wu, A. Shao, and J.-S. Pan, “CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm,” *Mathematics*, vol. 11, no. 10, 2339, 2023.

[12] T.-Y. Wu, H. Li, and S.-C. Chu, “CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps,” *Mathematics*, vol. 11, no. 9, 1977, 2023.

[13] Z. Li, Z. Zang, Y. Han, L. Wu, and H. Fu, “Solid-state FMCW LiDAR with two-dimensional spectral scanning using a virtually imaged phased array,” *Optics Express*, vol. 29, no. 11, pp. 16547-16562, 2021.

[14] I. Kostavelis, and A. Gasteratos, “Semantic mapping for mobile robotics tasks: A survey,” *Robotics and Autonomous Systems*, vol. 66, pp. 86-103, 2015.

[15] S. Wang, X. Chen, G. Ding, Y. Li, W. Xu, Q. Zhao, Y. Gong, and Q. Song, “A lightweight localization strategy for LiDAR-guided autonomous robots with artificial landmarks,” *Sensors*, vol. 21, no. 13, 4479, 2021.

[16] C. Park, S. Kim, P. Moghadam, C. Fookes, and S. Sridharan, ”Probabilistic surfel fusion for dense LiDAR mapping,” *Sensors*, vol. 21, no. 11, pp. 2418-2426.

[17] C.-C. Wong, H.-M. Feng, and K.-L. Kuo, “Multi-Sensor Fusion Simultaneous Localization Mapping Based on Deep Reinforcement Learning and Multi-Model Adaptive Estimation,” *Sensors*, vol. 24, no. 1, 48, 2023.

[18] P. Su, S. Luo, and X. Huang, “Real-time dynamic SLAM algorithm based on deep learning,” *IEEE Access*, vol. 10, pp. 87754-87766, 2022.

[19] F. Jin, A. Sengupta, S. Cao, and Y.-J. Wu, ”Mmwave radar point cloud segmentation using gmm in multimodal traffic monitoring,” in *2020 IEEE International Radar Conference (RADAR)*, IEEE, 2020, pp. 732-737.

[20] L. Renaut, H. Frei, and A. Nüchter, “Lidar pose tracking of a tumbling spacecraft using the smoothed normal distribution transform,” *Remote Sensing*, vol. 15, no. 9, 2286, 2023.

[21] S. Gao, and R. Hui, “Frequency-modulated continuous-wave lidar using I/Q modulator for simplified heterodyne detection,” *Optics Letters*, vol. 37, no. 11, pp. 2022-2024, 2012.

[22] D. J. Lum, S. H. Knarr, and J. C. Howell, “Frequency-modulated continuous-wave LiDAR compressive depth-mapping,” *Optics Express*, vol. 26, no. 12, pp. 15420-15435, 2018.

[23] S. Roehr, P. Gulden, and M. Vossiek, “Precise distance and velocity measurement for real-time locating in multipath environments using a frequency-modulated continuous-wave secondary radar approach,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, no. 10, pp. 2329-2339, 2008.

[24] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31-43, 2010.

[25] H. Taheri, and Z. C. Xia, “SLAM; definition and evolution,” *Engineering Applications of Artificial Intelligence*, vol. 97, 104032, 2021.

[26] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, “A comprehensive survey of visual SLAM algorithms,” *Robotics*, vol. 11, no. 1, 24, 2022.

[27] B. Alsadik, and S. Karam, “The simultaneous localization and mapping (SLAM)-An overview,” *Journal of Applied Science and Technology Trends*, vol. 2, no. 02, pp. 147-158, 2021.

[28] Z. Teed, and J. Deng, “Droid-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 16558-16569, 2021.

[29] S. Macenski, and I. Jambrecic, “SLAM Toolbox: SLAM for the dynamic world,” *Journal of Open Source Software*, vol. 6, no. 61, 2783, 2021.