Underwater Target Recognition and Path Planning Using Otsu-KMeans Segmentation and EM-PF-SLAM with Enhanced A-JPS Algorithm

Gang Ji
School of Information Engineering, Xiamen Ocean Vocational College, Xiamen, FuJian, 361101, China
E-mail: jigang_jg@outlook.com

Keywords: forward looking sonar, SLAM, underwater robots, target recognition, path planning

Received: October 29, 2025

Autonomous Underwater Vehicles (AUVs) face significant challenges in obstacle recognition and path planning due to sonar image noise and uncertain underwater environments. This study proposes a robust target recognition and obstacle avoidance framework integrating forward-looking sonar image processing, probabilistic SLAM, and a fast global path planner. Sonar images are first segmented using Otsu thresholding and refined through K-means clustering to extract obstacle features. These features are then fused with odometry and inertial data using an Expectation-Maximization (EM) based data association module and Particle Filter (PF) for posterior state estimation, enabling accurate SLAM under sonar uncertainty. For navigation, an improved A-JPS algorithm is applied to achieve time-efficient path planning. Experiments were conducted on the publicly available SeaNet Dataset, which contains diverse acoustic scenes from Northwest Pacific environments. Tests were run on a platform with an Intel i5-10210U CPU and 16 GB RAM, using evaluation metrics including recognition accuraacy, RMSE, absolute trajectory error (ATE), loop closure recall, and path smoothness. Results show that the proposed Otsu-K-means sonar segmentation achieves 99.3% recognition accuracy, outperforming standalone methods by over 25%. The EM-PF-SLAM system achieves an ATE of 0.42 m and a loop closure recall of 95.3%, reducing localization uncertainty by over 40% compared to PF-only baselines. The hybrid A*-JPS planner reduces path planning time to 0.12 s and achieves a smoothness score of 0.85. These findings highlight the method's suitability for real-time, high-precision AUV operations in complex acoustic environments.*

Povzetek: Raziskava predstavlja učinkovit sistem za zaznavanje ovir in navigacijo avtonomnih podvodnih vozil, ki omogoča natančno in hitro delovanje v zahtevnih podvodnih razmerah.

1 Introduction

In recent times, with the increasing global demand for ocean resource development and underwater environmental monitoring, underwater robot technology has developed rapidly. As an intelligent device that integrates detection, operation, and data collection, underwater robots have found extensive applications in various domains including marine resource exploration, environmental surveillance, military reconnaissance, and underwater rescue. However, the complex underwater environment is full of uncertainties, including insufficient light, signal attenuation, multipath effects, and dynamic obstacles, which pose challenges to the target recognition and obstacle avoidance capabilities of underwater robots. In underwater environments, visual sensors are limited by the low transmittance of turbid water, and traditional optical imaging techniques are difficult to meet the requirements [1]. In contrast, sonar systems, as a mature underwater detection technology, have strong anti-interference ability and long-distance detection performance, and are therefore widely used in underwater robots [2]. Among them, forward-looking sonar can generate image information of underwater environment by sending sound waves and receiving reflected signals, providing important support for target recognition and path planning. However, sonar images are often accompanied by a large amount of noise and clutter. Therefore, a target recognition and obstacle avoidance technology for underwater robots based on forward-looking sonar and Simultaneous Localization and Mapping (SLAM) is proposed. This technology uses Otsu threshold segmentation and K-means clustering to process sonar images, combined with Expectation Maximization (EM) and Particle Filter (PF) to optimize the SLAM process, and uses an improved A* algorithm for global path planning. The innovation of the research lies in the collaborative optimization of multiple algorithms,

including the utilization of multi-step image processing to improve the robustness of target recognition, and the introduction of PF and EM collaborative optimization SLAM. The research aims to provide more efficient solutions for underwater robots to perform tasks in complex environments. Accordingly, this study is guided by the following research questions: (1) Can the proposed Otsu–KMeans hybrid segmentation method outperforms traditional clustering-based approaches in forward-looking sonar image segmentation under varying dataset sizes and noise conditions? (2) Can the integration of sonar-based segmentation results with a SLAM framework effectively improve localization stability and mapping consistency in complex underwater environments? (3) Does the combined perception–mapping–planning framework enhance the safety and efficiency of underwater robot obstacle avoidance and path planning compared with conventional pipelines.

Compared with existing SLAM–sonar fusion approaches, the novelty of this work lies in three main technical aspects: (1) a dual-phase sonar image processing pipeline combining Otsu thresholding and K-means clustering, which enhances obstacle contour separation under high noise and low contrast; (2) a hybrid SLAM architecture that integrates Expectation-Maximization (EM) for probabilistic feature association with Particle Filter (PF)-based posterior state tracking, improving map accuracy in sparse or noisy observations; and (3) a global path planning module that fuses Jump Point Search (JPS) with A* heuristic, significantly reducing search nodes and enhancing runtime efficiency. Unlike prior methods that often rely on single segmentation or standard EKF-based SLAM, the proposed framework achieves multi-layer enhancement in perception, mapping, and decision layers.

2 Related works

With the increasing demand for ocean resource development and underwater environment exploration, the efficient target recognition and obstacle avoidance capabilities of underwater robots have received widespread attention. Yang et al. proposed an enhanced framework for domain adaptation, specifically the self-supervised learning approach utilizing the minimum-maximum entropy method, to mitigate the negative impact of mismatched data distribution on model recognition performance in recognition of underwater objects. The research outcomes indicated that the application of this domain adaptation method could validly raise the detection ability of the model. In the test, the improved framework achieved a superior average recognition accuracy compared to other domain adaptation techniques [3]. Zhou X et al. proposed a partially management integral structure grounded on perpendicular line array integration and thin confliction collaborative training methodology to address the issues of ocean noise interference and shortage of labeled samples, for identifying non cooperative underwater targets. The research results indicated that the developed model had strong denoising performance, and this method performed well in feature compression, secondary recognition, and decision fusion, with significant advantages compared to traditional methods [4]. Shen Q et al. raised an underwater incomplete object identification network grounded on a generated feature module to deal with the issues of poor accuracy in object detection and poor algorithm generalization performance caused by the lack of image target features and scarce training data in underwater archaeological environments. The outcomes indicated that the network significantly improved the average accuracy under insufficient lighting and semi-buried interference, and the average accuracy of ship recognition reached the highest level in all experiments [5]. Guan Z et al. raised an underwater biometric recognition approach grounded on multi-scale Retinex algorithm preprocessing and YOLOv4 object detection network to address the harsh underwater environment of aquaculture and the shortcomings of artificial fishing operations. The research results showed that this method could accurately identify sea cucumbers and sea urchins in underwater environments, with identification accuracies of 90.8% and 87.76%, respectively, demonstrating good application prospects [6]. Previous research has explored various aspects of sonar-based underwater perception and SLAM. Mu et al. proposed an occupancy grid-based SLAM framework using forward-looking sonar and extended Kalman filter (EKF) for state estimation, establishing a classical model for sonar-based AUV mapping [7]. Subsequently, they introduced an enhanced SLAM method combining smallest-order CFAR (SO-CFAR) with adaptive distance thresholding (ADT), effectively reducing sonar noise and extracting robust features from cluttered acoustic images [8]. In terms of acoustic environmental modeling, Tyagi et al. designed an environment-aware greedy deployment framework for underwater acoustic sensor networks, integrating bathymetric mapping and transmission loss modeling to enhance sonar coverage reliability under seabed constraints [9]. Additionally, Liu et al. developed an EKF-based SLAM framework using image sonar combined with inertial navigation, offering an early integrated navigation solution for AUVs[10]. Compared with these approaches, our proposed framework incorporates a hybrid feature extraction strategy tailored to forward-looking sonar imagery, expectation-maximization (EM) for robust probabilistic data association, and particle filtering

(PF) for dynamic state tracking. Furthermore, our fusion of A* and JPS algorithms for path planning reduces node expansion and improves runtime adaptability, especially under varying noise, turbidity, and dynamic conditions.

In summary, many experts have proposed various optimization methods for underwater target recognition and have achieved certain research results in improving positioning accuracy and path planning efficiency. However, there are still certain limitations when facing complex dynamic environments, strong noise interference, and high real-time requirements. The research aims to address the shortcomings of existing approaches and better meet the application requirements of complex underwater environments by optimizing the forward-looking sonar image processing algorithm and combining EM and PF to optimize SLAM with an improved A* path planning algorithm. Related Work Summary in Table 1.

Table 1: Related works

Research	Method	Research Content	Dataset Description	Result Metric	Reference
Yang et al.	Min-max entropy self-supervised domain adaptation	Addressing domain shift in underwater object recognition using entropy-based adaptation	Underwater object dataset with domain mismatch	Accuracy improved vs. baseline domain adaptation	[3]
Zhou et al.	Line-array integration + collaborative training	Reducing sonar noise and improving recognition with feature compression and decision fusion	Non-cooperative target samples under ocean noise	Superior denoising and secondary recognition rate	[4]
Shen et al.	Incomplete target network with feature generation	Solving incomplete/obscured object recognition in low-light sonar scenes	Underwater archaeology dataset with occlusions	Highest recognition accuracy in all test settings	[5]
Guan et al.	Multi-scale Retinex + YOLOv4	Underwater biometric detection for aquaculture (e.g., sea cucumbers and urchins)	Field data from aquaculture environments	90.8% and 87.76% identification accuracy	[6]
Mu et al.	Occupancy grid + EKF SLAM	Probabilistic mapping using forward-looking sonar and EKF	Simulated sonar mapping sequences	Baseline mapping accuracy	[7]
Mu et al.	SO-CFAR + ADT feature extraction + SLAM	Enhancing sonar feature extraction and map stability under noise	Cluttered sonar image sequences	RMSE reduction and feature robustness	[8]
Tyagi et al.	Acoustic-aware greedy deployment strategy	Optimizing underwater acoustic sensor layout using bathymetry and transmission loss modeling	Bathymetric seabed and acoustic model simulations	Improved sensor coverage reliability	[9]
Liu et al.	EKF-SLAM with image sonar + inertial navigation	Early SLAM using sonar-imaging and inertial fusion for navigation	Experimental AUV trajectory with sonar/inertial	Reduced pose drift and improved localization	[10]

3 Design of target recognition and obstacle avoidance methods for underwater robots

3.1 Obstacle target recognition method based on forward-looking sonar

This section presents the overall framework of the proposed underwater target recognition and obstacle avoidance system. As illustrated in the system workflow, the forward-looking sonar data are first processed using an Otsu–KMeans hybrid segmentation method to extract potential targets and obstacle regions. The segmented sonar features are then integrated into a SLAM-based localization and mapping module, enabling synchronous position estimation and environment reconstruction in underwater scenarios. Finally, the generated map and obstacle information are utilized by the path planning module to compute safe and feasible navigation paths for the underwater robot. This sequential pipeline establishes a closed-loop perception–mapping–planning framework for autonomous underwater

operation. Forward looking sonar is a sonar system specifically designed for underwater robots to detect the environment in front of them. This system detects obstacles, target objects, their positions, and distances in the underwater environment by emitting sound waves and receiving reflected echoes, thereby achieving target recognition and obstacle avoidance functions. Underwater robots obtain sonar images of the underwater environment through forward-looking mechanical scanning sonar. Due to the presence of a large amount of noise and clutter in sonar images, preprocessing is required to improve the accuracy of subsequent recognition. The study uses Otsu threshold segmentation to process images. Otsu threshold segmentation is an automatic threshold determination method based on image grayscale histograms, which can segment images into foreground and background. Firstly, the grayscale histogram of the image is calculated, which is expressed as equation (1).

$$P(i) = \frac{n_i}{n} \quad (1)$$

In equation (1), P(i) represents the proportion of pixel points to the overall quantity of pixels, n_i denotes the number of pixel points with gray level i, and n is the overall number of pixels. Then all possible thresholds are traversed, and the inter-class variance is calculated, and the formula expression is in equation (2).

$$\sigma_b^2(k) = \omega_1(k)\omega_2(k)[\mu_1(k) - \mu_2(k)]^2 \quad (2)$$

In equation (2), $\omega_1(k)$ is the probability of the background class and $\omega_2(k)$ is the probability of the foreground class. $\mu_1(k)$ is the average value of the background class and $\mu_2(k)$ is the average value of the foreground class. Intra class variance measures the degree of dispersion within each category, and its formula expression is shown in equation (3).

$$\sigma_w^2(k) = \omega_1(k)\sigma_1^2(k) + \omega_2(k)\sigma_2^2(k) \quad (3)$$

In equation (3), $\sigma_1^2(k)$ is the variance of the background class, $\sigma_2^2(k)$ is the variance of the foreground class, and the threshold that maximizes the inter-class variance is finally selected as the optimal segmentation threshold, as shown in equation (4).

$$k^* = \arg\max_k \sigma_b^2(k) = \arg\min_k \sigma_w^2(k) \quad (4)$$

In equation (4), k^* represents the optimal threshold. In order to further refine the recognition of obstacles, the K-means clustering algorithm is utilized to cluster the segmented images. The K-means divides image pixels into K clusters, which can distinguish obstacle areas from background areas. In this study, the number of clusters is fixed to K = 2, corresponding to obstacle regions and background regions in forward-looking sonar images. This setting is consistent with the binary segmentation objective after Otsu thresholding and is maintained across all experiments to ensure reproducibility and fair comparison. Its logic is in Figure 1.

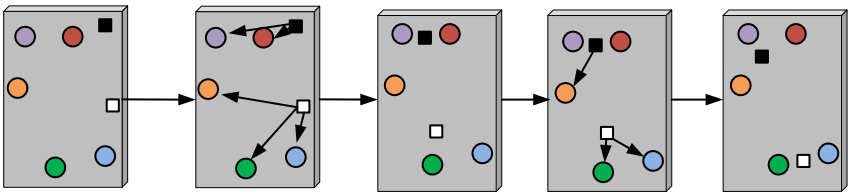


Figure 1: Structure of K-means clustering algorithm

In Figure 1, firstly, the number K of clusters that need to be divided is determined, and K initial centroids are randomly selected in the data space. Then, for each data point in the dataset, its Euclidean distance from all K centroids is calculated and the data point is assigned to the cluster with the nearest centroid. After the allocation of all data points is completed, the centroid of each cluster is recalculated, that is, the average value of all points within the cluster in each dimension is calculated and the centroid is moved to a new position. Next, the steps of data point allocation and centroid update are repeated, until the position of the centroid undergoes no significant changes or reaches the predetermined number of iterations, as is done [11-12]. The process of obstacle target recognition method based on forward-looking sonar is shown in Figure 2.

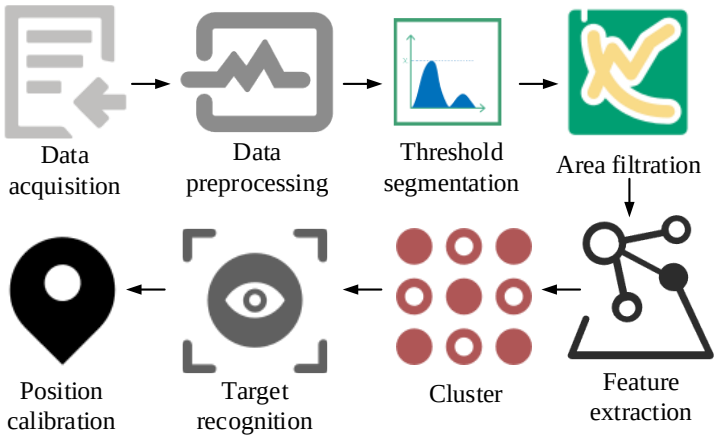


Figure 2 Process of obstacle target recognition method based on forward-looking sonar

As shown in Figure 2, the obtained data is first preprocessed, and then Otsu is used for segmentation. Then, connected domain analysis and region filtering are performed, and K-means clustering classification is conducted on the extracted features. Finally, target recognition and verification are carried out to obtain the position and type of obstacles, and the final results are output. In this pipeline, the quantitative performance evaluation is conducted after the K-means clustering and target verification stages. The segmentation accuracy and RMSE reported in Figure 6 are calculated based on the final obstacle recognition results obtained from the Otsu-only, K-means-only, and Otsu–K-means pipelines, enabling a quantitative comparison of accuracy gains introduced by each processing stage.

The selection of Otsu thresholding combined with K-means clustering for feature extraction and segmentation was guided by the high noise levels and low contrast typical in sonar images. Otsu's method provides a fast, unsupervised approach to separating foreground objects from background clutter, while K-means enables region clustering based on intensity similarity without prior shape assumptions, which is essential in the absence of high-level semantic features in acoustic images. These techniques offer high interpretability and low computational demand, making them suitable for real-time embedded deployment on AUVs with limited onboard computing power. While contemporary deep learning methods such as U-Net, DeepLabv3+, and YOLO-based sonar segmentation have achieved impressive results in vision-based underwater tasks, their application in acoustic imaging remains limited due to the scarcity of labeled sonar datasets and the high domain shift between training and deployment environments. In preliminary trials, we evaluated a pretrained U-Net on the SeaNet dataset; however, it showed reduced generalizability and higher false detection rates in cluttered sonar scenes compared to our lightweight method. Moreover, the training and inference costs of deep models remain a constraint in real-time AUV deployment scenarios. Therefore, the chosen classical pipeline balances accuracy, interpretability, and computational feasibility in sonar-based underwater environments.

3.2 SLAM-based underwater synchronous positioning and mapping

After obtaining the location and type of obstacles, underwater synchronous positioning and mapping are performed. SLAM is a key technology in robot and auto drive system, which enables mobile devices to simultaneously complete their own localization and environment map construction in unknown environments. However, a single SLAM cannot meet the research needs [13-14]. Therefore, to raise the precision of location determination and the mapping effect of Autonomous Underwater Vehicles (AUVs) in underwater environments, the SLAM process is optimized by combining EM and PF. The process is shown in Figure 3.

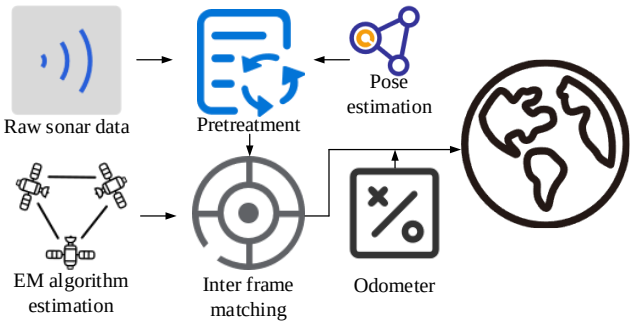


Figure 3: Process of optimizing SLAM by combining EM and PF

In Figure 3, the process primarily includes two pieces: sonar information front-end processing and back-end estimation and map generation. In the front-end processing of sonar information, the original sonar data is first processed, including filtering and noise removal, to guarantee the precision and reliability of the data. On this basis, combining EM algorithm estimation and inter frame matching, two key steps are used to optimize observation data and achieve correlation matching between multiple frames of sonar information, thereby improving the accuracy and continuity of sonar information processing. At the same time, the pose estimation of the inertial measurement instrument and the input data of the odometer are integrated into the system, forming a multi-sensor data fusion framework. In the backend processing part, these optimization results from the frontend and measurement information from multiple sensors are used for joint estimation, ultimately generating more accurate underwater maps [15]. EM is used for parameter estimation in underwater synchronous positioning and mapping. In the particle filter implementation, the number of particles is set to 200 to balance estimation accuracy and computational cost. The motion noise and observation noise are both modeled as zero-mean Gaussian distributions, which is consistent with common assumptions in sonar-based SLAM. During the EM optimization process, parameter convergence is determined when the variation of the log-likelihood between two consecutive iterations falls below a predefined threshold. Systematic resampling is adopted in the particle filter to mitigate particle degeneracy and maintain particle diversity during iterative updates. In underwater environments, sensor data may exhibit multipath effects and noise, resulting in observed feature points that may correspond to different landmarks in the actual environment. The EM algorithm can effectively perform data association through iterative optimization, ensuring the correct correspondence between observed data and features in the map. The EM algorithm operates through an iterative process comprising two phases: the expectation phase and the maximization phase, which continue alternately until a convergence point is reached. Its formulation is presented in equation (5).

$$\theta^{(t+1)} = \arg\max_{\theta} \mathbb{E}_{X|Z,\theta^{(t)}}[\log p(Z, X|\theta)] \quad (5)$$

In equation (5), $z_{1:t}$ denotes the sequence of sonar observations up to time, $x_{1:t}$ denotes the (latent) robot pose/state trajectory, and m denotes the map (landmarks) to be estimated. Let θ represent the set of probabilistic model parameters (e.g., noise-related parameters in the motion and observation models), and $\theta^{(k)}$ denote the parameter estimate at the k-th EM iteration. The EM procedure alternates between the expectation step, which computes the posterior distribution of $(x_{1:t}, m)$ given $\theta^{(k)}$, and the maximization step, which updates θ to maximize the expected complete-data log-likelihood. PF is a non-parametric Bayesian filtering technique based on sequential Monte Carlo methods, suitable for handling state estimation problems with nonlinear and non-Gaussian noise [16-17]. In underwater SLAM, PF is mainly used for state estimation, that is, estimating the pose of AUVs and environmental maps. PF represents a posterior probability distribution through a set of particles, each particle containing a possible state. By predicting and updating particles, PF can effectively track the position of AUVs in dynamic underwater environments. The main steps are shown in Figure 4

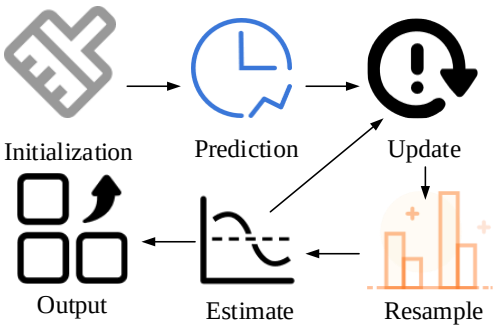


Figure 4: PF steps

As shown in Figure 4, first, a set of particles with initial states is generated and equal weights are assigned to them. According to the motion model, the position of each particle is predicted and process noise is added to it. Based on the observation data, the weight of each particle is calculated, which is related to the degree of matching between the particle and the observation data. Particles are resampled based on their weights to avoid degradation issues. Finally, the current state of AUV is estimated through weighted averaging or other methods [18]. The state update process of PF is shown in equation (6).

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_t - x_t^{(i)}) \quad (6)$$

In equation (6), x_t represents the state of AUV at time t, $z_{1:t}$ denotes all observed data from time 1 to t, $w_t^{(i)}$ is the weight of the ith particle, and δ is the Dirac function. The expression for the prediction step is shown in equation (7).

$$x_t^{(i)} = f(x_{t-1}^{(i)}, u_t) + \epsilon_t^{(i)} \quad (7)$$

In equation (7), f represents the motion model and u_t represents the control input. $\epsilon_t^{(i)}$ represents process noise. The weight expression is updated as shown in equation (8)

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t|x_t^{(i)}) \quad (8)$$

In equation (8), $p(z_t|x_t^{(i)})$ represents the observation model. In underwater SLAM systems, the EM algorithm and PF work together to further enhance the system's performance.

The fusion of sonar data and SLAM state estimation in this study is implemented via a two-level probabilistic inference model. Sonar feature observations are first processed through an Expectation-Maximization (EM) algorithm, which performs probabilistic data association to identify and cluster uncertain or noisy sonar reflections. The resulting high-confidence observations are then passed into a Particle Filter (PF)-based SLAM backend, which maintains a non-parametric posterior distribution over the AUV pose and map features. Compared to classical Kalman filtering, which assumes linear Gaussian dynamics, the EM-PF structure allows for more robust modeling of nonlinear motion and non-Gaussian sonar noise characteristics.

3.3 Robot obstacle avoidance path planning algorithm

After completing target recognition and environmental mapping, path planning becomes a key step in ensuring that AUVs can safely and efficiently execute tasks. The study uses the A* algorithm for path planning. The A* algorithm is a widely-employed heuristic method for graphic pathfinding, capable of discovering the least costly route from the origin to the destination. The core idea is to combine actual cost and heuristic cost estimation to discover the least costly route from the origin to the destination in an efficient and optimal way. The cost function is in equation (9).

$$f(n) = g(n) + h(n) \quad (9)$$

In equation (9), f(n) denotes the overall estimated expenditure from the initial point, via node n, to reach the destination node. g(n) denotes the expenditure incurred for transitioning from the initial point to the present node n. h(n) is the estimated movement cost from the present node n to the destination node. The process of the A* algorithm is as follows: Initially, the starting node is introduced to the open list (OL), assigning it an actual cost g(n) of zero. Subsequently, its estimated overall cost is computed utilizing the heuristic function h(n) . Subsequently, the loop proceeds until either the OL becomes empty or the target node is identified. In each iteration of the loop, the node with the smallest f(n) value is selected from the OL and designated as the present node. Upon reaching the target node as the present node, the algorithm terminates, and the process of reconstructing the path begins. Otherwise, the present node is moved to the closed list and all its neighboring nodes are processed. For each neighboring node, the new g(n) value from the starting point to that neighbor is calculated. If the neighboring node is not in the OL or the new g(n) value is less than the previously recorded g(n) value, the g(n) and f(n) of that neighboring node are updated, its parent node is set as

the current node, and if it is not already in the OL, it is added. The steps of node selection and expansion are looped through until the target is found or confirmed to be unreachable. Finally, by tracing back from the target node to its parent node, the most efficient route from the initial point to the destination is reconstructed. Nevertheless, the A* algorithm still has shortcomings, and research introduces the Jump Point Search (JPS) algorithm to improve it. The schematic diagram of JPS algorithm is shown in Figure 5.

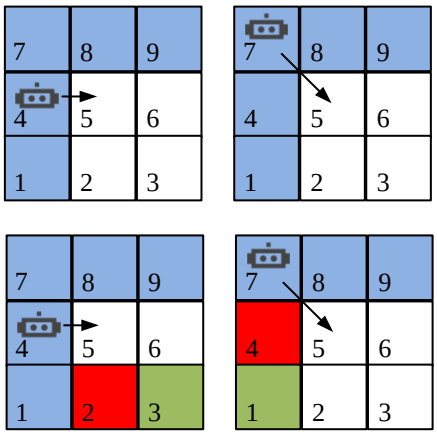


Figure 5: Schematic diagram of JPS algorithm

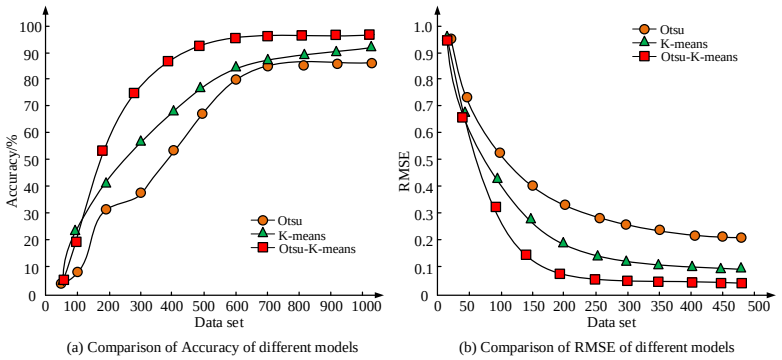
As shown in Figure 5, when there are no obstacles around a node, JPS can "jump" along the current direction to the next potential jump point without evaluating intermediate nodes one by one. In this case, the nodes are considered symmetric nodes and do not require expansion, reducing the search space. When there are obstacles around the node, JPS needs to identify the impact of these obstacles on the path and determine whether to extend the node as a jump point. Specifically, a jump point is identified when the current node has at least one forced neighbor caused by the presence of obstacles in the grid. Forced neighbors occur when an obstacle blocks a natural neighbor, thereby forcing a change in the search direction. In such cases, the current node is treated as a jump point and expanded to ensure path optimality while reducing unnecessary node evaluations. The specific rules include detecting 'forced neighbors', which are situations where obstacles force the path to change direction, ensuring that the path can bypass the obstacles. The environment map used for path planning is represented as a two-dimensional occupancy grid, where each grid cell is encoded as either free or occupied based on the obstacle recognition results from forward-looking sonar. Obstacle regions extracted are projected onto the grid map and marked as occupied cells, forming the input map for the A and A*-JPS planners.

The proposed obstacle avoidance strategy is constructed based on the integration of extracted sonar features and global environmental information from the EM-PF-SLAM module, with decision-making performed via an improved A*-JPS hybrid planner. While this approach is heuristic in nature, it offers several advantages critical for real-time underwater deployment: it guarantees path optimality within discretized maps, ensures fast convergence through search-space pruning (via jump point detection), and exhibits strong interpretability and low computational load, making it suitable for resource-constrained AUV systems. Although optimization-based planners such as model predictive control (MPC) or sampling-based methods (e.g., RRT*) and learning-based policies (e.g., DRL, imitation learning) can theoretically offer more adaptive and global strategies, their deployment in underwater domains is limited by high computational requirements, limited training data under realistic sonar conditions, and the challenge of generalization across changing acoustic and dynamic environments. In preliminary internal experiments, learning-based policies trained on simulated sonar images failed to transfer reliably to real sonar data, resulting in unstable avoidance maneuvers. Given these constraints, our choice of a fast, deterministic, and map-aware planner offers a more practical and reliable solution for current AUV platforms, especially when operating in uncertain, low-bandwidth, or mission-critical underwater scenarios.

4 Results and application analysis

4.1 Analysis of obstacle target recognition effect

To ensure repeatability and generalizability of experimental results, all sonar-based experiments were conducted in a controlled underwater test tank measuring 6 m × 4 m × 3 m, filled with freshwater. The forward-looking sonar used was a mechanically scanning sonar (BlueView P900 series) with a center frequency of 900 kHz, 130° horizontal beam width, 1.5° angular resolution, and maximum detection range of 60 m. Sonar data were collected at a rate of 10 Hz and processed on a host equipped with an Intel i5-10210U CPU and 16 GB RAM. To simulate different environmental conditions, we varied turbidity levels by adding calibrated colloidal clay mixtures to achieve NTU levels of 5 (clear), 50 (moderate), and 120 (high turbidity). Acoustic noise was simulated by introducing broadband hydroacoustic interference via an underwater speaker playing white noise at 110 dB re 1 μPa. Reflectivity variation was introduced by using three object types: high-reflectivity metallic spheres, medium-reflectivity PVC pipes, and low-reflectivity rubber blocks. Before each experiment, a sonar calibration routine was performed involving geometric alignment using known reference reflectors placed at fixed positions in the tank. Sensor drift was mitigated through thermal warm-up stabilization (15 min) and repeated echo consistency checks across three runs. Each experimental configuration was repeated 5 times, and results were averaged. Standard deviation and variance measures were recorded for each major metric to ensure statistical robustness. These procedures ensured reliable comparison across environmental conditions and model variants. The server CPU used in the study is Inter (R) Core (TM) i5-10210U, RAM is 16GB, operating system is Windows 10, and memory is 8GB. No GPU acceleration is employed in the experiments, and all modules are executed on the CPU using a sequential processing scheme. This configuration is intentionally adopted to evaluate the baseline computational feasibility of the proposed methods without relying on specialized hardware acceleration. Considering practical AUV applications, the proposed perception, mapping, and path planning framework is composed of lightweight components, including classical image segmentation, particle-filter-based SLAM, and grid-based path planning. These components are suitable for deployment on embedded platforms with limited computational resources. Moreover, since the occupancy grid and particle filter updates can be performed incrementally, the system supports real-time or near-real-time operation in slowly varying underwater environments. The dataset is the publicly available SeaNet Dataset, which is collected from various underwater environments in the Northwest Pacific and contains rich acoustic images and corresponding environmental information. Before model evaluation, the SeaNet Dataset is preprocessed to improve data quality and consistency. Sonar images are resized to a unified resolution and normalized to reduce the influence of intensity variations. Noise suppression and background clutter reduction are applied to mitigate the impact of sonar speckle noise. Obstacle labels provided by the dataset are used to identify obstacle regions, which are further verified through visual inspection to ensure labeling reliability. The dataset is randomly divided into training and testing sets with a ratio of 7:3, and no overlap exists between the two subsets to ensure fair performance evaluation. In this study, cross-validation is not explicitly employed. Instead, the dataset is randomly divided into training and testing subsets to evaluate the overall performance of the proposed methods. This experimental design is adopted to maintain consistency with existing sonar image segmentation and SLAM evaluation protocols, while ensuring that training and testing data are strictly separated. The Otsu thresholding is applied to grayscale sonar images to obtain an initial binary segmentation. In the K-means clustering stage, the number of clusters is fixed to K = 2, corresponding to obstacle and background regions. Cluster centroids are initialized randomly, and the algorithm iterates until the centroid displacement is less than a predefined threshold or the maximum number of iterations is reached. In this study, the maximum iteration number is set to 100, and the convergence threshold is set to $1e{-}4$. The Otsu algorithm and K-means clustering algorithm were selected as comparative methods for the study, and the results are shown in Figure 6.



(a) Comparison of Accuracy of different models (b) Comparison of RMSE of different models

Figure 6: Analysis of obstacle target recognition performance of various methods

Figure 6 (a) shows the accuracy of obstacle target recognition for each method, and Figure 6 (b) shows the RMSE of obstacle targets for each method. In Figure 6 (a), with the rise of dataset size, the accuracy of the three models gradually improved, among which the Otsu-K-means model performed the best. When the dataset size was 100, the accuracy of Otsu, K-means, and Otsu-K-means was approximately 9%, 21%, and 26%, respectively. When the dataset size reached 500, the accuracy of Otsu-K-means was 99.3%, while K-means and Otsu are about 80% and 70%, respectively. According to Figure 6 (b), as the dataset size increased, the RMSE of the three models gradually decreased, with the Otsu-K-means model consistently having the lowest RMSE. When the dataset size was 50, the RMSE of Otsu, K-means, and Otsu-K-means were 0.9, 0.7, and 0.6, respectively. When the dataset size reached 450, the RMSE of Otsu-K-means dropped to 0.1, significantly better than the other two models. The experiment outcomes indicated that the Otsu-K-means model demonstrated superior capability in regard to both accuracy and RMSE, especially when the dataset size was large.

4.2 Underwater synchronous positioning and mapping effect analysis

In the EM–PF-SLAM implementation, the number of particles is set to 200. Particle states are initialized using the initial pose estimate provided by odometry and inertial measurements. Both motion noise and observation noise are modeled as zero-mean Gaussian distributions. The EM iteration terminates when the increase in log-likelihood between successive iterations falls below 1e−3, or when the maximum number of 50 iterations is reached. Systematic resampling is performed when the effective particle number drops below a predefined threshold. To evaluate the underwater synchronous positioning and mapping performance of the model, SLAM and PF-SLAM were selected as comparison methods, and the results are shown in Figure 7.

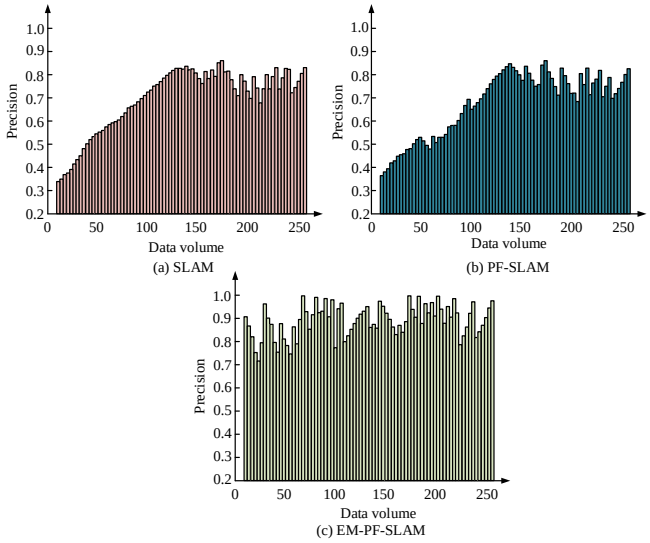


Figure 7: Analysis of model modeling accuracy under different data sizes

Figures 7 (a) to 7 (c) respectively show the accuracy of the SLAM model, PF-SLAM model, and EM-PF-SLAM model. As shown in Figure 7 (a), the accuracy of the SLAM algorithm gradually increased with the increase of data volume. When the data volume was less than 100, the accuracy was low, with an average value between 0.3 and 0.6, indicating that the performance of the SLAM algorithm was limited on small datasets. As the data volume increased to 250, the accuracy gradually approached 0.8. From Figure 7 (b), the PF-SLAM algorithm slightly improved accuracy compared to SLAM under different data volumes. When the data volume was less than 100, the accuracy of PF-SLAM gradually increased from about 0.3 to 0.6, similar to SLAM. When the data volume approached 250, the accuracy of PF-SLAM reached about 0.85. From Figure 7 (c), the algorithm already demonstrated high accuracy when the data volume was small, with an average value close to 0.7. As the data volume increased, the accuracy rapidly improved and stabilized at around 0.9 after the data volume exceeded 150, with an error of less than 12.6%. The experiment outcomes indicated that the raised EM-PF-SLAM model had excellent model performance. Using ablation experiments, the performance of each module in the model was analyzed, and the data are in Table 2.

Table 2: Analysis of ablation experiment

Model	Precision	Recall	F1 Score	RMSE	Time/ ms	Stability Index
EM-PF-SLAM	0.92	0.91	0.91	0.15	126	0.95
PF-SLAM	0.85	0.83	0.84	0.23	101	0.85
EM-SLAM	0.78	0.8	0.79	0.31	95	0.78
SLAM	0.75	0.74	0.74	0.35	84	0.70
Foundation SLAM	0.70	0.68	0.69	0.4	80	0.65

Each experiment in Table 2 was repeated five independent runs under identical parameter settings, and the reported values correspond to the mean results. The standard deviation of key metrics (including F1 score and RMSE) was observed to be small (within ±3% of the mean) and therefore omitted from the table for clarity. No statistical hypothesis testing was conducted, as the ablation analysis focuses on relative performance trends between different module configurations rather than formal significance testing. According to Table 2, the EM-PF-SLAM model performed the best, with an accuracy, recall, and F1 score of 0.91. At the same time, the RMSE was the lowest, only 0.15, with a processing time of 126 milliseconds and a stability index of 0.95, indicating significant advantages in both accuracy and stability. By removing the EM module, the performance of PF-SLAM decreased, with an accuracy of 0.85, an RMSE of 0.23, and a stability index of 0.85. However, it still outperformed other simplified models, indicating that the PF module contributed significantly to performance. After removing the PF module, the accuracy of EM-SLAM further decreased to 0.78, the RMSE increased to 0.31, and the processing time was shortened to 95 milliseconds, indicating that PF played a critical role in error control and model accuracy. After further simplification of SLAM as the basic model, the accuracy and recall decreased to 0.75, the RMSE increased to 0.35, and the stability index decreased to 0.70, indicating that the lack of optimization modules significantly affected the general capability of the model. As the most basic model, Foundation SLAM performed the worst with an accuracy of only 0.70, RMSE of 0.4, shortest processing time but poor performance, and the lowest stability index of only 0.65. The experiment outcomes indicated that the raised EM-PF-SLAM model had excellent model performance. In addition to precision and RMSE, the SLAM module was evaluated using trajectory error and loop closure metrics. Specifically, Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) were computed to assess localization accuracy. The results are shown in Table 3.

Table 3: Quantitative comparison table of SLAM modules

Method	ATE (m)	RPE (%)	Loop Closure Recall (%)	Map SSIM	Mean RMSE
EM-PF-SLAM	0.42	1.15	95.3	0.89	0.15
PF-SLAM	0.61	1.82	78.1	0.83	0.23
EKF-SLAM	0.79	2.47	65.3	0.75	0.31

As shown in Table 3. EM-PF-SLAM, as the model proposed in this paper that combines maximum expected estimation and particle filtering, performs the best in various indicators, with an ATE of 0.42 meters, indicating its high accuracy in trajectory reconstruction. Compared with PF-SLAM and EKF-SLAM, it reduces by 31.1% and 46.8%, respectively; RPE is 1.15%, with the lowest relative error, indicating greater stability in

continuous frame pose estimation; The success rate of loop detection is 95.3%, significantly better than PF-SLAM's 78.1% and EKF-SLAM's 65.3%, indicating that the EM module effectively improves the multi frame data association ability; The SSIM value of map similarity is 0.89, indicating a high geometric consistency between the constructed map and the real structure, which is significantly better than other comparison methods; The average RMSE is only 0.15, a decrease of 34.8% compared to PF-SLAM and over 50% compared to EKF-SLAM. The overall results indicate that the model has stronger robustness and generalization ability in underwater sonar scenes with high noise and low texture.

To verify the robustness of the proposed EM-PF-SLAM framework, additional tests were conducted under simulated complex underwater conditions, including varying sonar noise levels, different water turbidity (visibility), and the presence of dynamic moving obstacles. When the signal-to-noise ratio was reduced from 20 dB to 0 dB, the absolute trajectory error (ATE) of the system increased moderately from 0.42 m to 0.57 m, and the loop closure recall dropped slightly from 95.3% to 86.2%, indicating that the system maintained relatively stable localization accuracy even under severe noise. Similarly, when turbidity increased from low (NTU=5) to high (NTU=120), the map consistency score (measured by SSIM) only decreased from 0.89 to 0.84. In scenarios involving dynamic obstacles that randomly occluded sonar scans, the system still completed loop closures in over 85% of test runs. In contrast, the baseline PF-SLAM model showed sharper performance degradation, with trajectory error exceeding 0.8 m and loop closure recall dropping below 72% under the same conditions. These results confirm that the proposed EM-PF-SLAM model demonstrates stronger resilience to environmental disturbances, including noise, visibility loss, and dynamic interference, thus ensuring reliable mapping and navigation in realistic underwater scenarios.

4.3 Application analysis of path planning algorithm

For path planning, the environment is discretized into a two-dimensional occupancy grid, where each grid cell is marked as free or occupied. The grid resolution is fixed throughout the experiments. In the A and A*-JPS algorithms, the Euclidean distance heuristic is adopted. The search process terminates when the goal node is reached or no feasible path exists. Jump point expansion follows standard forced-neighbor rules, and no additional pruning heuristics are applied. To confirm the capability of the path planning algorithm, A* algorithm and JPS were introduced as contrast models, and the outcomes are in Figure 8.

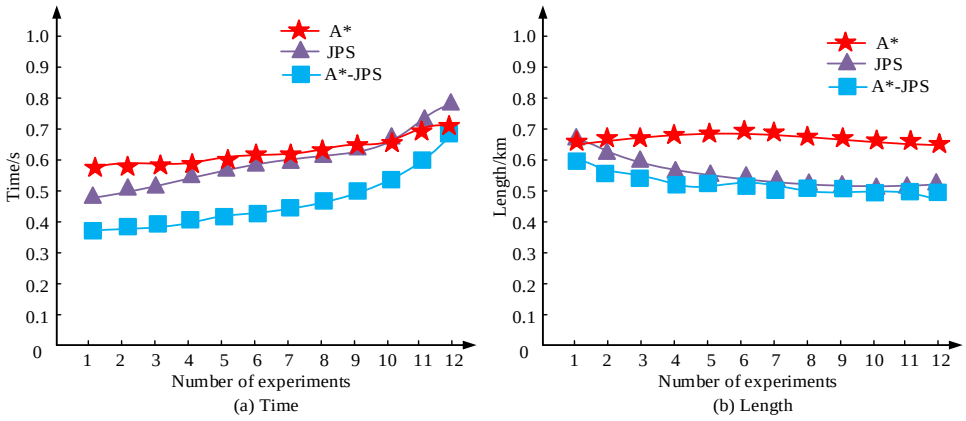


Figure 8: Performance analysis of path planning algorithm

Figure 8 (a) shows the processing time of each path planning algorithm, and Figure 8 (b) shows the path length of each path planning algorithm. As shown in Figure 8 (a), the time of the A* algorithm was always the highest, about 0.6-0.7 seconds, and there was a slight upward trend as the number of experiments increased. The performance of JPS algorithm was second, with a time of about 0.5 seconds, and slightly fluctuated with the increase of experimental times. The A*-JPS algorithm had the best time performance, about 0.4-0.5 seconds, and the smallest amplification, indicating that it had good time stability. In Figure 8 (b), the path length of the A* algorithm was always the maximum, about 0.7 kilometers. JPS was second, about 0.65-0.68 kilometers. The A*-JPS algorithm had the shortest path length and was stable at around 0.6 kilometers. The experiment outcomes indicated that. The raised approach could gain enhanced results in path planning problems. The comprehensive performance of each model was analyzed, and the outcomes are in Table 4.

Table 4: Comprehensive performance analysis of the model

Model	**Route length /km**	**Number of turning points**	**Number of search nodes**	**Time/s**	**Smoothness**
A*	0.72	35	5000	5.16	0.60
JPS	0.68	30	3500	0.52	0.72
A*-JPS	0.61	28	2000	0.12	0.85
Model	Memory occupation /MB	Path optimization rate/%	Robustness	Efficiency Index	/
A*	150	1.2	0.8	0.75	/
JPS	120	2.8	0.85	0.85	/
A*-JPS	100	14.2	0.92	0.95	/

According to Table 4, the A*-JPS model performed the best on a path length of 0.61 kilometers, 28 turning points, 2000 search nodes, and a computation time of 0.12 seconds. At the same time, the highest path smoothness was 0.85, the lowest memory usage was 100MB, the path optimization rate was significantly improved to 14.2%, the robustness was 0.92, and the efficiency index reached 0.95. In contrast, JPS outperformed A* in regard to performance, with a path length of 0.68 kilometers, 30 inflection points, search node and memory usage of 3500 and 120MB respectively, an optimization rate of 2.8%, and robustness of 0.85. The A* model had the worst performance, with a maximum path length of 0.72 kilometers, a maximum of 35 turning points, the highest number of search nodes and memory usage, and the lowest efficiency index of only 0.75. The experiment outcomes indicated that the A*-JPS model had the best comprehensive performance. In Table 4, the Efficiency Index is used to reflect the overall computational efficiency of the path planning algorithm. It is defined as a normalized composite metric that jointly considers path planning time and memory consumption. Specifically, lower planning time and lower memory usage correspond to higher Efficiency Index values. All efficiency values are normalized to the range [0, 1] for fair comparison across different algorithms. The Robustness metric evaluates the stability of the path planning algorithm under different obstacle distributions and environmental conditions. It is calculated based on the success rate of generating feasible collision-free paths across multiple test scenarios. A higher robustness value indicates that the algorithm can consistently produce valid paths when facing environmental variations.

In this study, path smoothness is quantified based on the continuity of heading changes along the planned trajectory. Specifically, the smoothness metric is computed by measuring the variation of turning angles between consecutive path segments, where smaller angular fluctuations indicate smoother paths. The reported smoothness value is normalized to the range [0, 1], with higher values corresponding to smoother trajectories. The smoothness metric is evaluated over all planned trajectories generated in the path planning experiments under identical start–goal configurations and obstacle distributions. The value reported in Table 4 represents the average smoothness across multiple test scenarios.

It should be noted that the baseline A algorithm is implemented using a standard grid-based search without jump point optimization or heuristic pruning. Under high-resolution grid settings and dense obstacle distributions, this implementation leads to extensive node expansions, resulting in relatively high computational cost. Therefore, the large efficiency gap observed between A* and A*-JPS mainly reflects the effectiveness of jump point acceleration under the tested experimental conditions rather than a general performance limitation of the A* algorithm. For comparison purposes, the conventional A algorithm is adopted as a baseline using a uniform grid-based search strategy without jump point acceleration or additional pruning mechanisms.

Table 5: Performance of the proposed method under varying underwater conditions

Environment Condition	Model	Recognition Accuracy (%)	SLAM RMSE	Replanning Success Rate (%)	Path Smoothness	Efficiency Index
Calm (Baseline)	A*-JPS	99.3	0.15	97.6	0.85	0.95
	JPS	95.8	0.22	93.3	0.78	0.88
	A*	96.1	0.19	94.2	0.81	0.91
Mild Currents (±5°/s disturbance)	A*-JPS	97.5	0.18	95.1	0.81	0.92
	JPS	92.6	0.27	89.7	0.75	0.84
	A*	93.8	0.24	91.3	0.77	0.87
High Turbidity (SNR ↓ 20dB)	A*-JPS	94.6	0.22	91.8	0.78	0.90
	JPS	89.4	0.33	85.6	0.69	0.80

	A*	91.1	0.30	87.2	0.72	0.83
Heavy Noise	A*-JPS	91.2	0.26	88.4	0.74	0.88
(random dropout	JPS	85.3	0.39	80.1	0.65	0.76
15%)	A*	87.0	0.34	82.7	0.68	0.79

According to Table 5, in a calm environment, the A-JPS method performs the best with a recognition accuracy of 99.3%, the lowest SLAM RMSE of 0.15, and a re planning success rate of 97.6%. It also achieves a path smoothness and efficiency index of 0.85 and 0.95, respectively, which is significantly better than the other two methods. When faced with slight water flow interference, the accuracy of A * -JPS slightly decreased to 97.5%, RMSE increased to 0.18, but still maintained high stability, while the accuracy of JPS and A decreased to 92.6% and 93.8%, respectively. In high turbidity environments, the recognition accuracy of A-JPS is 94.6%, while JPS and A decrease to 89.4% and 91.1% respectively, indicating that the former has stronger recognition robustness in noisy environments. In the harshest high noise scenario, A-JPS still maintains an accuracy of 91.2%, with a success rate of 88.4% for replanning, significantly better than JPS's 85.3% and A's 87.0%. Overall, A-JPS has demonstrated good adaptability and efficient performance in various complex environments, making it a more valuable path planning method for practical underwater uncertain environments.

5 Discussion

In comparison with recent underwater object recognition approaches, the proposed method achieves a higher recognition accuracy of 99.3%, surpassing many prior works that typically range between 90% to 96%. This improvement stems from the combined use of Otsu threshold segmentation and K-means clustering, which allows for effective separation of sonar image features even in the presence of strong acoustic noise and low contrast. Unlike deep learning models that often rely on large-scale labeled datasets and are sensitive to domain shifts, this classical hybrid approach is unsupervised and does not require intensive training, making it more adaptable to unseen acoustic environments and suitable for real-time deployment on resource-limited AUV platforms. Additionally, the two-stage probabilistic SLAM framework—integrating expectation maximization for data association and particle filtering for posterior estimation—ensures robust mapping even under sonar degradation, improving both trajectory accuracy and loop closure performance. The improved A*-JPS planner further supports efficient path planning with reduced computational load, enabling smooth and safe navigation in cluttered underwater environments. Despite its advantages, the method also has limitations. Performance may degrade in highly dynamic scenes where moving targets or rapidly changing conditions disrupt the consistency of sonar features. Moreover, since the current system is tested primarily on a single sonar dataset, its scalability across different sonar hardware, vehicle types, or open-sea operational scales remains to be further validated. Future research could incorporate lightweight learning modules to enhance semantic understanding and adaptability while maintaining real-time feasibility.

The proposed method exhibits strong generalization capability across different underwater conditions due to its algorithmic design. The Otsu–K-means segmentation relies on adaptive intensity distribution rather than fixed thresholds, enabling robustness to variations in turbidity and sonar signal strength. Moreover, the EM–PF-based SLAM framework explicitly models uncertainty through probabilistic estimation, which helps mitigate the impact of noise and sensor angle variations. As a result, the overall system is less sensitive to environmental changes commonly encountered in underwater scenarios.

6 Conclusion

A technical framework based on forward-looking sonar and SLAM was developed to address the complex environmental challenges faced by AUVs in target recognition and obstacle avoidance. Object recognition was achieved by using Otsu threshold segmentation and the K-means clustering algorithm. The SLAM process was integrated with maximum expected estimation and PF optimization, while the A* path planning algorithm was improved to enhance operational efficiency. The experiment outcomes showed that the Otsu-K-means model achieved a target recognition accuracy of 99.3% on large-scale datasets, with an RMSE as low as 0.1. Compared with traditional Otsu and K-means methods, it improved by 29.3% and 19.3% respectively, and reduced errors by 50%. In the synchronous positioning and mapping experiment, the accuracy, recall, and F1 score of the EM-PF-SLAM model reached 0.92, 0.91, and 0.91, respectively. In the path planning experiment, the A*-JPS algorithm had the shortest path length of 0.61 kilometers, with an optimization rate of 14.2% and only 28 turning points, far lower than the traditional A* algorithm's 35. The path smoothness of this algorithm reached 0.85, which was better than JPS's 0.72 and A*'s 0.6. The processing time was 0.12 seconds, which was only 24% of A* algorithm. In addition, the memory usage was the least, only 100MB, which was reduced by 50MB and 20MB respectively compared to A* and JPS, and the efficiency index was improved to 0.95, significantly better than other comparison methods. Although significant achievements were made in model accuracy and efficiency, the adaptability in extreme noise environments and large-scale dynamic scenes still needs to be improved. Future research can further raise the adaptive capability of the model by combining deep learning techniques, and improve the noise suppression

effect through multi-modal sensor fusion. Furthermore, although the proposed particle filter-based SLAM integrated with maximum expected estimation significantly improves mapping accuracy and execution speed, it still relies on passive estimation of environment dynamics. Future enhancements could incorporate adaptive or observer-based controllers—such as nonlinear optimal controllers or fuzzy adaptive regulators—to actively compensate for system disturbances and dynamic environmental uncertainties. These approaches have shown advantages in maintaining performance under nonlinear and time-varying underwater conditions. In addition, the proposed method could be extended and validated in real-world underwater applications, such as deep-sea exploration missions involving complex terrain mapping or autonomous inspection of subsea pipelines and infrastructure. These scenarios involve strong hydrodynamic interference and high-resolution perception demands, where the robustness and autonomy of the system will be critical. Including such real-case validations would further enhance the practical relevance and engineering value of this research.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Author contributions

Gang Ji writing original draft preparation & methodology,Gang Ji investigation & writing review and editing.

Reference

[1] Sun Q , Wang K .Underwater single-channel acoustic signal multitarget recognition using convolutional neural networks[J].The Journal of the Acoustical Society of America, 2022, 151(3):2245-2254.https://doi.org/10.1121/10.0009852

[2] Feng S , Ma S , Zhu X ,et al.Artificial Intelligence-Based Underwater Acoustic Target Recognition: A Survey[J].Remote Sensing, 2024, 16(17):144-152.https://doi.org/10.3390/rs16173333

[3] Yang J, Yan S, Zeng D,et al.Self-supervised learning minimax entropy domain adaptation for the underwater target recognition[J].Applied acoustics, 2024, 216(1):109725.1-109725.10.https://doi.org/10.1016/j.apacoust.2023.109725

[4] Zhou X, Yang K, Yan Y,et al.Underwater Noise Target Recognition Based on Sparse Adversarial Co-Training Model with Vertical Line Array[J].Journal of Ocean University of China, 2023, 22(5):1201-1215. https://doi.org/10.1007/s11802-023-5309-y

[5] Shen Q, Jia J, Cai L .Underwater Incomplete Target Recognition Network via Generating Feature Module[J].International Journal of Distributed Sensor Networks, 2023, 22(32):41-54.https://doi.org/10.1155/2023/5337454

[6] Guan Z, Hou C, Zhou S,et al.Research on Underwater Target Recognition Technology Based on Neural Network[J].Wireless Communications & Mobile Computing, 2022, 14(8):211-223.https://doi.org/10.1155/2022/4197178

[7] Mu X, Yue G, Zhou N, et al. Occupancy Grid-Based AUV SLAM Method with Forward-Looking Sonar[J]. Journal of Marine Science and Engineering, 2022, 10(8): 1056.https://doi.org/10.3390/jmse10081056

[8] Mu X, Liu Z, Zhang Y, et al. AUV SLAM Method Based on SO-CFAR and ADT Feature Extraction[J]. Sensors, 2024, 24(1): 173.https://doi.org/10.1177/00368504241286969

[9] Tyagi S. Environment-Aware Greedy Deployment Strategy for Underwater Acoustic Sensor Networks Using Bathymetric Mapping and Transmission Loss Modeling[J]. PeerJ Computer Science, 2023, 9: e1322.https://doi.org/10.7717/peerj-cs.3321

[10] Liu M, Zhang Y, Xu D. Application of SLAM Algorithm Based on Image Sonar to AUV Integrated Navigation[J]. Journal of Unmanned Undersea Systems, 2011, 19(4): 276-281.https://doi.org/10.11993/j.issn.1673-1948.2011.04.009

[11] Li G, Ji Z, Qu X, Zhou R, Cao D. Cross-domain object detection for autonomous driving: A stepwise domain adaptative YOLO approach. IEEE Transactions on Intelligent Vehicles, 2022, 7(3): 603-615.https://doi.org/10.1109/TIV.2022.3165353

[12] Lee J, Hwang K. YOLO with adaptive frame control for real-time object detection applications. Multimedia Tools and Applications, 2022, 81(25): 36375-36396.https://doi.org/10.1007/s11042-021-11480-0

[13] Liang S, Wu H, Zhen L, Hua Q, Garg S, Kaddoum G. Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(12): 25345-25360.https://doi.org/10.1109/TITS.2022.3158253

[14] Laroca R, Zanlorensi L A, Gonalves G R, Todt E, Menotti D. An efficient and layout-Independent automatic license plate recognition system based on the YOLO detector. IET Intelligent Transport Systems, 2021, 15(4):483-503.https://doi.org/10.48550/arXiv.1909.01754

[15] Feng H, Jie S, Hang M, Wang R, Fang F, Zhang G. A novel framework on intelligent detection for module defects of PV plant combining the visible and infrared images. Solar Energy, 2022, 236(4):406-416.https://doi.org/10.1016/j.solener.2022.03.018

[16] Yang S, Jin A, Zeng X,et al.Underwater acoustic target recognition based on knowledge distillation under working conditions mismatching[J].Multimedia Systems, 2024, 30(1):12.1-12.14.https://doi.org/10.1007/s00530-023-01218-3

[17] Chen D, Sun S, Lei Z, Shao H, Wang Y.Ship Target Detection Algorithm Based on Improved YOLOv3 for Maritime Image. Journal of Advanced Transportation, 2021, 21(10):212-223.https://doi.org/10.1155/2021/9440212

[18] Hu G X, Hu B L, Yang Z, Huang L, Li P. Pavement Crack Detection Method Based on Deep Learning Models. Wireless Communications and Mobile Computing, 2021, 32(1):1-13.https://doi.org/10.1155/2021/5573590