IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2021, accepted June 7, 2021, date of publication June 11, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3088541

Bathymetric Particle Filter SLAM With Graph-Based Trajectory Update Method

QIANYI ZHANG, YE LI, TENG MA, ZHENG CONG, AND WENJUN ZHANG
Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China

Corresponding authors: Ye Li (liyeheu103@163.com) and Teng Ma (mateng_heu@hrbeu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0309400; in part by the National Natural Science Foundation of China under Grant U1806228, Grant 52001093, Grant 51879057, and Grant 51909245; in part by the Research Fund for Science and Technology on Underwater Vehicles Laboratory under Grant 6142215180102; and in part by the Science and Technology Projects of Hainan Province under Grant 520LH006.

ABSTRACT A graph-based particle filter bathymetric simultaneous localization and mapping (BSLAM) method is proposed to solve the oscillation problem of the trajectories estimated by particles when using a low precise vehicle motion model and obtain accurate navigation results for autonomous underwater vehicles (AUVs). A graph-based trajectory update method is proposed to update the trajectories stored in particles before particle weighting to weaken the influence of the low precise odometer model on the particle trajectories. A particle weighting method based on submap matching is proposed to improve the robustness of the particle filter. Besides, a graph-based map generation method is proposed to solve the map selection problem of the particle filtering theory. The performance of the proposed method is demonstrated using a simulated dataset and a field dataset collected from a sea trial. The results show that the proposed method is more accurate and effective compared with a state-of-art particle filter BSLAM method.

INDEX TERMS Autonomous underwater vehicle, bathymetric simultaneous localization and mapping, particle filter, pose graph optimization.

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) are widely used in underwater topography mapping to obtain high-resolution bathymetric maps [1], [2]. Precise positioning and navigation are the prerequisites for accurate mapping [3]. Conventional underwater navigation methods, such as the long-baseline and ultrashort-baseline acoustic positioning methods require external equipment or support from the vessel [4]. Dead reckoning (DR) can provide short-term navigation results for AUVs without external supplements, but the navigation error accumulates over time, thus rendering it unsuitable for long-term underwater positioning [5].

Simultaneous localization and mapping (SLAM) enables AUVs to construct a map of unknown environments and positions AUVs in the same map simultaneously [6]. Many SLAM methods estimate the difference between the states of a vehicle by calculating the similarity of associated features, but due to the lack of identifiable features (such as valleys or mountains) on seabed topographies, the feature-based SLAM methods are not suitable for underwater environments [7]. Bathymetric SLAM (BSLAM) is a featureless SLAM method that directly uses the terrain information collected by the sensors (often a multibeam sonar) to measure the difference between the previous and current observations, hence it bypasses the feature extraction problem [8].

Particle filter BSLAM methods are normally based on grid or trajectory map representations. The bathymetric distributed particle SLAM (BPSLAM) presented by Barkby *et al.* [9] is a particle filter BSLAM method using the grid map representation. In this method, each grid stores estimated terrain depths and the corresponding particle IDs. The particles are weighted by the information stored in the grids, and their maps are constructed through the particle ancestry tree. This method was also applied in iceberg mapping [10]. However, as this representation requires huge memory resources, it is not suitable for large-scale seabed mapping. Besides, the resolution of grids influences the precision of the bathymetric map. Another particle filter BSLAM method proposed by Barkby *et al.* [11] is based on the trajectory map representation, which maintains each particle map by the particle ancestry tree and a list of bathymetric

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang.

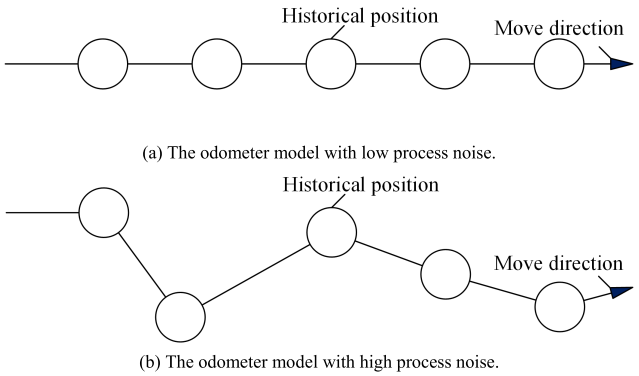


FIGURE 1. The process noise in the odometer model influences the trajectory stores in a particle.

observations shared by all particles. Each particle stores an individual trajectory since the last execution of the resampling, and entire trajectories of particles can be reconstructed by accessing the corresponding particles and their parent IDs from the ancestry tree. This method greatly reduces the memory consumption compared with BPSLAM, but the particle weighting based on Gaussian process regression requires a substantial amount of computational resources.

Besides, the particle filter BSLAM methods listed above have three weaknesses:

1) Only current bathymetric observations are considered in particle weighting. As the number of valid measurement points input to the particle weighting is limited, these methods may face perceived ambiguity problems when dealing with similar topography.
2) As shown in Fig. 1(b), if the process noise in the odometer model is large, which leads to the oscillation of the particle trajectories, the particle trajectories cannot effectively represent a posterior hypothesis of the vehicle. As the final map selection is used in these methods to choose a particle and its trajectory as the output at the end of the mission, the results of the BSLAM are influenced by the unreliable odometer model.
3) As an online SLAM method, the historical positions stored in each particle cannot be updated based on current bathymetric observations in particle filter BSLAM. The results provided by the particle filter BSLAM are only the optimal state at the current moment.

The contributions of the paper are combining the particle filter BSLAM framework with graph theory to solve the above problems, specifically:

1) A trajectory update method based on graph theory is designed to reduce the influence of the unreliable odometer model on the trajectories stored in particles.
2) A particle weighting method based on submap matching is proposed to allow more valid measurement points include in particle weighting.
3) A map generation method based on graph theory is used to solve the map selection problem of the particle filter.

The rest of this paper is organized as follows. Section II outlines related work. Sections III and IV detail the proposed algorithm, and Section V presents the results of the playback experiments. Finally, we draw the conclusions in Section VI.

II. RELATED WORK

BSLAM methods can be classified as filter-theory-based or graph-theory-based:

A. FILTER-THEORY-BASED BSLAM

Roman and Singh [12] proposed a BSLAM method based on the delayed state Kalman filter. The point cloud map is divided into groups of submaps, the trajectory is constrained by pairwise matching the overlapping submaps. Another approach reported by Palomer *et al.* [13] used a two-step Probabilistic Iterative Closest Point (ICP) algorithm in submap matching to reduce the computational time and the probability of falling into local minima during the registration. Although the above methods reduce the errors among submaps, they cannot handle the internal errors within the submaps Fairfield *et al.* [14] proposed a particle filter BSLAM using the occupancy grid-based volumetric map representation. The datasets collected from the closed cave and open marine environment demonstrated this method can generate a consistent three-dimensional bathymetric map.

B. GRAPH-THEORY-BASED BSLAM

Bichucher *et al.* [15] proposed a BSLAM method using Generalized ICP to provide a translation constraint between overlapping submaps. Another robust method proposed by Ma *et al.* [16] used the multi-window consistency method to avoid the influence of invalid loop closures on results, and its performance has been demonstrated in several experiments.

As featureless SLAM uses bathymetric measurements as observations, the noise of the sensors influences the performance of BSLAM. Particle filter (PF) is a non-parametric implementation of Bayes filter that can be used to approximate the probability distribution of a state by a set of samples (particles) from the distribution [17]. In PF BSLAM, as each particle contains its own map (trajectory), and the particles are evaluated with their previous observed topography to find the best fit, the PF framework is considered suitable for BSLAM due to its robustness. The PF BSLAM framework used in the proposed method is given in Algorithm 1.

III. PARTICLE PROPAGATION

By using the high precision fiber optic gyrocompass and altimeter, the impact of the systematic error of depth, heading, pitch, and roll on the vehicle state is virtually negligible, the particle filter only focuses on tracking the uncertainty of the AUV on horizontal positions [9], [11], [14], [16]. Thus, the particle set $P_{(t)}$ with N particles at time t is given by

$$P_{(t)} = \begin{Bmatrix} S_1^{(t)} & \cdots & S_n^{(t)} & \cdots & S_N^{(t)} \\ Trj_1^{(t)} & \cdots & Trj_n^{(t)} & \cdots & Trj_N^{(t)} \\ w_1^{(t)} & \cdots & w_n^{(t)} & \cdots & w_N^{(t)} \\ L_1 & \cdots & L_n & \cdots & L_N \end{Bmatrix}, \quad (1)$$

Algorithm 1 PF BSLAM framework

for time $t = 1$ to total mission time (T) **do**
  Input particle set at time $t-1$ ($P_{(t-1)}$), control at time t ($u_{(t)}$), bathymetric observations collected using a multibeam sonar at time t ($z_{(t)}$).
  for particle $n = 1$ to particle set size (N) **do**
    Propagate the position of particle n : $S_n^{(t)} \sim p(S_n^{(t)}|u_t, S_n^{(t-1)})$.
    Propagate the map stored in particle n : $Trj_n^{(t)} \sim p(Trj_n^{(t)}|S_n^{(t)}, Trj_n^{(t-1)})$.
    Update the weight of particle n: $w_n^{(t)} \sim p(z_t|S_n^{(t)})$.
  end for
  Resample the particle set.
  Update the map based on surviving particles.
  Output $P_{(t)}$.
end for

where $S_n^{(t)}$ is the horizontal position of the particle n at time t, $Trj_n^{(t)}$ is the trajectory (historical positions) of the particle n from the beginning of the mission to the time t, $w_n^{(t)}$ is the weight of the particle n, and L_n is the loop closures detected by the particle n. To reduce memory consumptions, each particle only stores its own trajectory, and a point cloud map can be constructed by the trajectory and corresponding bathymetric observations.

A. PARTICLE STATE PROPAGATION

At the beginning of the mission, all the particles are initialized with the ground-truth location of the AUV obtained by GPS or acoustic positioning methods. At time t, an odometer model is applied to update the state of the particle n:

$$S_n^{(t)} = S_n^{(t-1)} + u_{(t)} + \omega, \tag{2}$$

where $u_{(t)}$ is the control vector at time t and ω is a process noise with Gaussian distribution.

B. PARTICLE MAP PROPAGATION

As an online BSLAM method, the conventional PF BSLAM not attempts to update historical positions of the particles, and the posterior distribution approximated by particles only represents the optimal approximation of the vehicle at the current time. It leads to a problem that $S_n^{(t)}$ is regarded as the high likelihood estimated positions of the vehicle but $Trj_n^{(t)}$ is probably not the high likelihood estimated trajectory. Besides, the trajectories stored in particles are influenced by the process noise ω. These trajectories used in particle weighting and output are unreliable when the value of ω is large. To prevent the issues, a pose-graph-based particle map propagation is conducted after the particle state propagation.

The pose graph theory is commonly used in BSLAM methods [18]–[20]. As a full SLAM method, the graph BSLAM updates the map based on all historical bathymetric measurements. However, it is sensitive to invalid loop closures, which means even a single invalid closure may lead the method to fail. The combination of the PF and pose graph can solve the problem that PF faces. Meanwhile, PF uses samples to approximate real distribution, which avoids the robust problems of the pose graph.

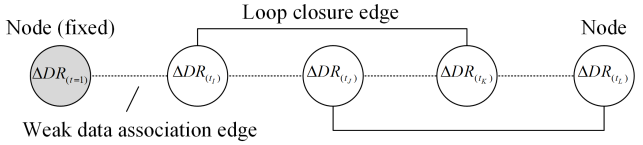


FIGURE 2. Pose graph structure with two loop closure edges.

The pose graph consists of nodes and edges. In the proposed BSLAM, after a particle n is updated by the odometer model, when the minimum horizontal distance between $S_n^{(t)}$ and $Trj_n^{(t-t_d)}$ is less than a predefined distance R, a loop closure l_m is considered detected

$$l_m = \left\{ t_i, t_j, \vec{V}_m \right\}, \tag{3}$$

where t_d is a predefined time, m is the index of the loop closure, t_i and t_j are the current time and the time corresponding to the historical position that closest to $S_n^{(t_i)}$, respectively, and $\vec{V}_m$ is the relative positions between $S_n^{(t_i)}$ and $S_n^{(t_j)}$

$$\vec{V}_m = S_n^{(t_i)} - S_n^{(t_j)}. \tag{4}$$

Then L_n with m loop closures is defined as

$$L_n = \{l_1 \dots l_m\}. \tag{5}$$

The pose graph is then constructed according to Ma *et al.* [16], as shown in Fig. 2. A pose graph with m loop closures has $2m+1$ nodes, m loop closure edges and $2m$-1 weak data association edges. The nodes in the pose graph are the offsets of the positions provided by DR at the times in L_n (denoted as $\Delta DR_{(t)}$). For l_m

$$Nodei : \Delta DR_{(t_i)} = S_n^{(t_i)} - DR_{(t_i)}, \tag{6}$$

$$Nodej : \Delta DR_{(t_j)} = S_n^{(t_j)} - DR_{(t_j)}, \tag{7}$$

where $DR_{(t)}$ is the position provided by the DR system at time t. The loop closure edge connects Node i and Node j is defined as

Loop closure edge i-j:

$$DR_{(t_i)} + \Delta DR_{(t_i)} - (DR_{(t_j)} + \Delta DR_{(t_j)}) - \vec{V}_m, \tag{8}$$

as shown in Fig. 3.

After sorting nodes in chronological order, for every three nodes that adjacent in time, a weak data association edge is constructed based on the terrain correlation correcting method [21]:

Weak data association edge I-J-K:

$$a_{t_J} \Delta DR_{(t_I)} + (1 - a_{t_J}) \Delta DR_{(t_K)} - \Delta DR_{(t_J)} \; t_I < t_J < t_K, \tag{9}$$

where a_{t_J} is calculated as

$$a_{t_J} = \left(\sum_{t=t_I}^{t_J} 1/p(S^{(t)}|S^{(t-1)})\right)/\left(\sum_{t=t_I}^{t_K} 1/p(S^{(t)}|S^{(t-1)})\right), \tag{10}$$

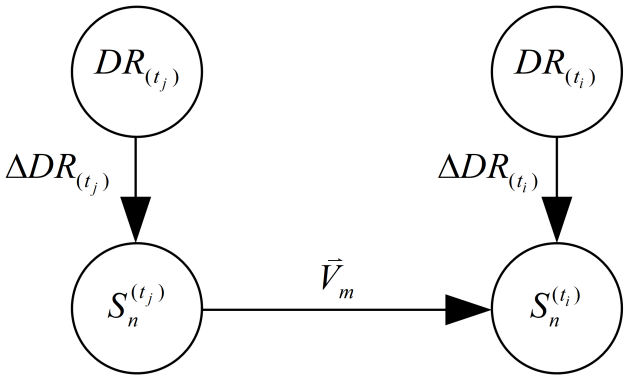


FIGURE 3. The vectors used in loop closure edge generation.

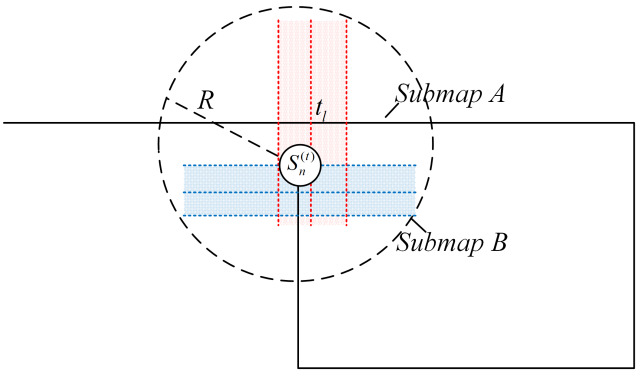


FIGURE 4. Submaps used in particle weighting, for $t_c = 2$.

and I, J, K are the node indexes that adjacent in time, respectively. $p(S^{(t)}|S^{(t-1)})$ is the interframe motion uncertainty of the vehicle, where $S^{(t)}$ is the vehicle position at time t. $p(S^{(t)}|S^{(t-1)})$ can be estimated using the similarity between the predicted and actual bathymetric observations. The terrain prediction methods such as Gaussian process regression are usually applied to calculated $p(S^{(t)}|S^{(t-1)})$. However, as terrain prediction methods require a substantial amount of computational resources. For an accurate inertial navigation system, we assume that interframe motion uncertainty of the vehicle is approximately constant with value p, hence (10) is approximated as

$$\begin{aligned} a_{t_J} &= (\sum_{t=t_I}^{t_J} 1/p(S^{(t)}|S^{(t-1)}))/(\sum_{t=t_I}^{t_K} 1/p(S^{(t)}|S^{(t-1)})) \\ &\approx (\sum_{t=t_I}^{t_J} 1/\mathrm{p})/(\sum_{t=t_I}^{t_K} 1/\mathrm{p}) = (t_J - t_I)/(t_K - t_I). \end{aligned} \quad (11)$$

Besides all the nodes of loop closures, an initial fixed node is added into the graph to make the graph computable, which defined as *Node 0*: $\Delta DR_{(t=1)}$. As the position provided by the DR is corrected by GPS at time $t = 1$, $\Delta DR_{(t=1)}$ is considered as a zero vector, and the value of $\Delta DR_{(t=1)}$ is not changed during the graph calculation.

According to the pose graph, a least squares problem

$$\begin{aligned} &\Delta DR^* \\ &= \arg\min \\ &\left\{ \begin{array}{l} \sum \left\| a_{t_J} \Delta DR_{(t_I)} + (1 - a_{t_J}) \Delta DR_{(t_K)} - \Delta DR_{(t_J)} \right\|^2 \\ + \\ \sum \left\| DR_{(t_i)} + \Delta DR_{(t_i)} - (DR_{(t_j)} + \Delta DR_{(t_j)}) - \vec{V}_m \right\|^2 \end{array} \right\} \end{aligned} \quad (12)$$

is constructed, and this graph is optimized by general graph optimization (G2o), detailed explanations of the G2o can be found in [22].

After the graph is optimized, the corrections of the DR navigation results between two adjacent nodes are calculated based on the same assumption of (11):

$$\Delta DR_{(t)} = a_t \Delta DR_{(t_I)} + (1 - a_t) \Delta DR_{(t_J)}. \quad (13)$$

Then the historical positions stored in a particle n at time i ($S_n^{(i)}$) are updated by

$$S_n^{(i)} = DR_{(i)} + \Delta DR_{(i)} i \in [1, t], \quad (14)$$

$$Trj_n^{(t)} = S_n^{(1:t)}. \quad (15)$$

IV. PARTICLE WEIGHTING AND MAP SELECTION

After the particle map propagation, if the particle n is close to its previous trajectory, this particle is weighted by submap matching. Compared with the weighting method that only includes current bathymetric observations, submap matching allows more valid measurement points in particle weighting, hence improves the robustness of the particle filter.

A. PARTICLE WEIGHTING

The same as the loop closure detection method described in Section III, if the minimum distance between the position of the particle n at current time t and its trajectory from $t = 1$ to $t - t_d$ is below R (the corresponding time of the closest historical position is denoted as t_l), two temporary point clouds are generated:

Submap A (from time $t - t_c$ to t), and

Submap B (from time $t_l - [t_c/2]$ to $t_l + [t_c/2]$), where $t_c \in N^*$ is a predefined value relative to the number of bathymetric swaths in submaps. As the horizontal positions of the measurement points are relative to the estimated positions of the vehicle, the submaps can be constructed by combining the trajectory and the bathymetric observations at the corresponding time, as shown in Fig. 4.

To estimate the similarity between submaps, the inverse distance weighting (IDW) method [23] is applied to interpolate *Submap B* to obtain the corresponding terrain depth at the same position of *Submap A*. Assumed that the difference between the bathymetric measurements at the same positions between submaps follows a Gaussian distribution [11], the weight of particle n is calculated as

$$w_n^{(t)} = \frac{1}{I} \sum_{i=1}^{I} p(z_A^{(i)} = z_B^{(i)}), \quad (16)$$

$$p(z_A^{(i)} = z_B^{(i)}) = \exp(\frac{-(z_A^{(i)} - z_B^{(i)})^2}{2\sigma_s^2})/\sqrt{2\pi\sigma_s^2}. \quad (17)$$

where $z_A^{(i)}$ and $z_B^{(i)}$ are the terrain depths of the *Submap A* and *Submap B* at the same horizontal position i, I is the number of the same horizontal positions between submaps, and σ_s^2 is the measurement variance of the sensor, respectively.

If the closest distance between particle n and its previous trajectory is out of R, the bathymetric measurements associated with the estimated current position of the vehicle stored in this particle are not considered overlapped with the historical bathymetric measurements, hence the particle n is not included in particle resampling.

After the particle weighting, if the effective particle size N_{eff}

$$N_{\text{eff}} = \frac{1}{\sum_{n=1}^{N} [\tilde{w}(n)]^2} \quad (18)$$

is less than a defined value N_{th}, the particles are resampled [24], where $\tilde{w}(n)$ is the normalized weight of the particle n.

B. MAP GENERATION

As each particle maintains a trajectory, it is important to develop an online method to generate the optimal trajectory of the vehicle. The commonly used map selection methods [9], [12] are offline, but online map generation is essential in the tasks like path planning. The online map selection method proposed by Barkby *et al.* [11] only compares the registration error between two submaps, one is generated by the currently observed swath and the other is a fusion of all previous overlapping swaths, the accuracy of this method is worse than offline map selection methods.

To generate the estimated trajectory of the PF BSLAM in real-time, a graph-based map generation method is proposed. The estimated positions of the vehicle between each time interval are considered as nodes, and the i-th node is defined as: *Node i*: $V_s^{(it_s)}$, where t_s is a predefined time interval. The same as Section III B, the fixed node $V_s^{(t=1)}$ is added in the graph, as well as the estimated position of the vehicle at time t.

The edges of the graph consist of the particle position edges and the week data association edges detailed in Section III B. The particle position edge represents the difference between the estimated position of the vehicle and the position stored in the particle

Particle position edge at time t:

$$V_s^{(t)} - S_n^{(t)}. \quad (19)$$

According to the graph shown in Fig. 5, a least squares problem

$$\Delta V_s^* = \arg\min \left\{ \begin{array}{l} \sum \left\| a_{t_J} V_s^{(t_I)} + (1 - a_{t_J}) V_s^{(t_K)} - V_s^{(t_J)} \right\|^2 \\ + \\ \sum \left\| V_s^{(it_s)} - S_n^{(it_s)} \right\|^2 + \left\| V_s^{(t)} - S_n^{(t)} \right\|^2 \end{array} \right\} \quad (20)$$

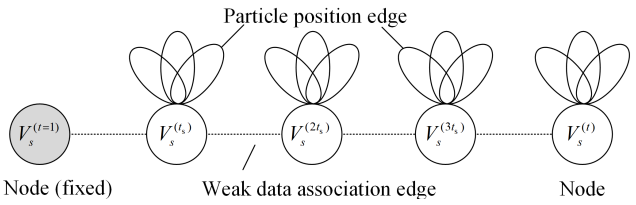


FIGURE 5. Graph structure with five nodes, and each node has three particle position edges, for *N*=3.

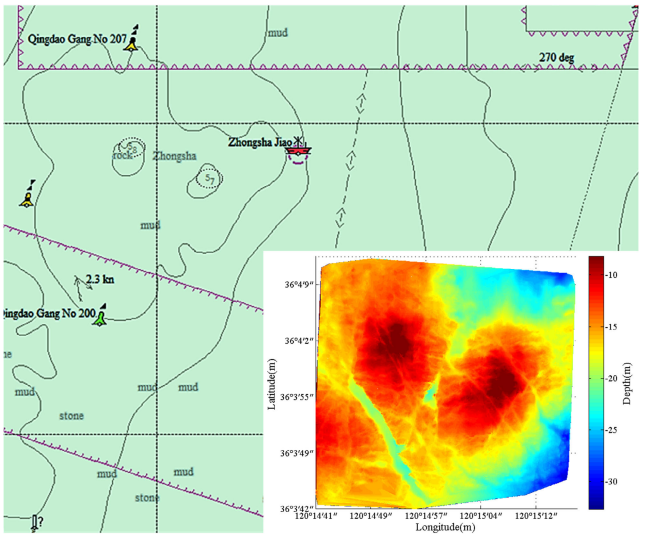


FIGURE 6. Sea trail region corresponding to experimental data.

is constructed, and the graph is optimized by G2o. The offset between the positions of $DR_{(it_s)}$ and $V_s^{(it_s)}$ is calculated as

$$\Delta DR_{(it_s)} = V_s^{(it_s)} - DR_{(it_s)}. \quad (21)$$

Then the offsets of the positions provided by DR at each time are generated using (13), and the estimated positions of the vehicle are calculated as

$$V_s^{(j)} = DR_{(j)} + \Delta DR_{(j)} \; j \in [1, t]. \quad (22)$$

V. PLAYBACK EXPERIMENTS

Two sets of playback experiments were conducted by using a simulated dataset and a field dataset. The original bathymetric data were collected from the sea trials around Zhongsha Reef, China, as shown in Fig. 6.

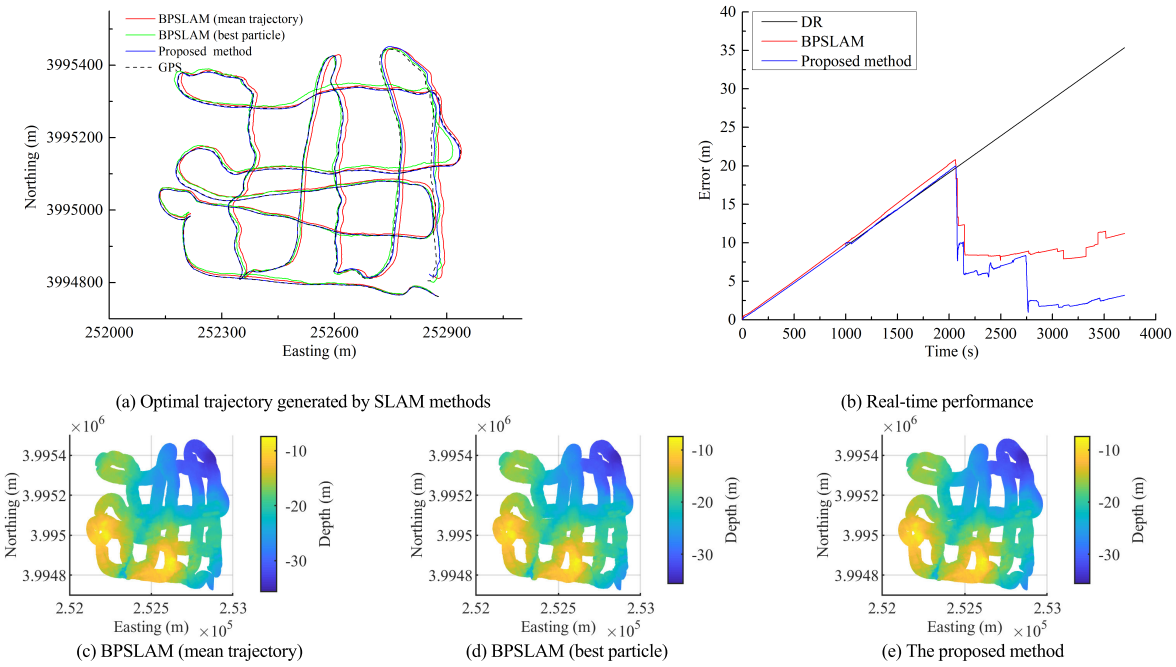
The BPSLAM detailed in Section I was used for comparisons. As DR data were not collected in experiments, the navigation results provided by DR were simulated by adding Gaussian Noise to the GPS data as follows:

$$DR_t = DR_{t-1} + GPS_t - GPS_{t-1} + \mathcal{N}(m_e, \sigma_e^2), \quad (23)$$

where GPS_t is the position of the vehicle provided by GPS at time t, m_e and σ_e are the mean, and the standard deviation of the Gaussian noise, respectively. The simulations were executed on a computer with Intel Core i5-6300HQ CPU and 16 GB RAM.

(a) Multibeam sonar (b) Data processing equipment (c) Fiber-optic gyroscope

FIGURE 7. Experimental equipment for data collection.



(a) Optimal trajectory generated by SLAM methods (b) Real-time performance

(c) BPSLAM (mean trajectory) (d) BPSLAM (best particle) (e) The proposed method

FIGURE 8. SLAM results for 500 particles. (c)-(e) are the maps generated by different trajectories.

A. EXPERIMENTS ON SIMULATED DATA

The original bathymetric data were collected by a Kongsberg GeoSwath Plus multibeam bathymetry system installed in a vessel with a sampling frequency of 1 Hz, as shown in Fig. 7. The differential GPS provided the actual horizontal position of the vehicle with 20 cm accuracy. The heading, pitch, and roll observations were yielded by a Fiber-optic gyroscope with an accuracy of 0.1°. The length of the vessel's trajectory is around 8 km in 3613 s. Each simulated swath contains 141 bathymetric measurement points. The parameters used in the simulations were: $m_e = 0.012$m, $\sigma_e = 0.01$m, $\sigma_s = 0.2$m, $N = 400$, $\omega \sim N(0, 0.25\ \mathrm{m}^2)$. For the proposed method: $R = 2$m, $t_\mathrm{d} = 500$, $t_\mathrm{c} = 20$, $t_\mathrm{s} = 500$. The resolution of the grid in BPSLAM was set as 1 m.

A group of results are demonstrated in Fig. 8. As the map selection method in BPSLAM is an offline method, the real-time error of the BPSLAM was calculated according to the mean of the trajectories stored in particles (mean trajectory). At the end of the mission, we choose the particle trajectory with the smallest error (best particle) as the output of the BPSLAM. The errors of BPSLAM (mean trajectory), BPSLAM (best particle), and the proposed method were 11.21, 7.46, and 3.19 m at the end of the mission, respectively. The computation times of the BPSLAM and the proposed method were 712.55 s and 69.45 s, respectively. After several particle resampling, the proposed method provided accurate results after 2750 s. The proposed method was 57.2% more accurate than BPSLAM at the end of the mission. Besides, because the proposed method saves time for grid map updates, it was also 90.2% faster than BPSLAM.

As shown in Fig. 9, the standard deviations of the particle set in both methods fluctuated after each resampling.

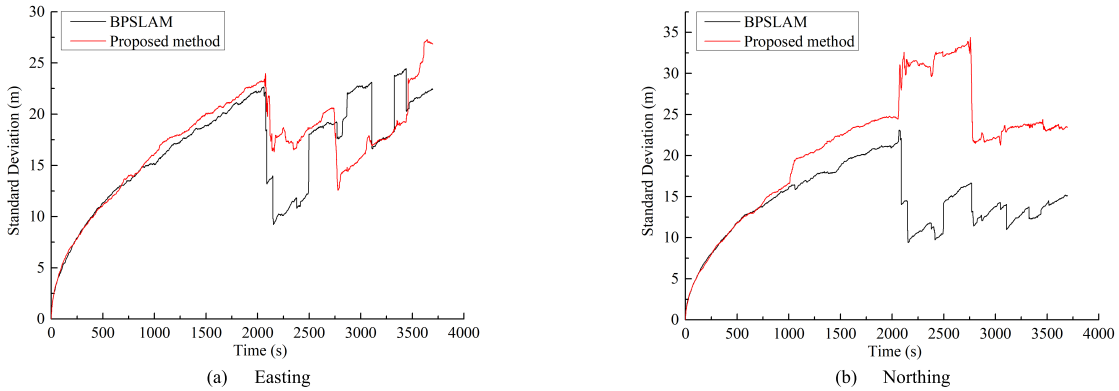


FIGURE 9. The standard deviation of the positions of the particles.

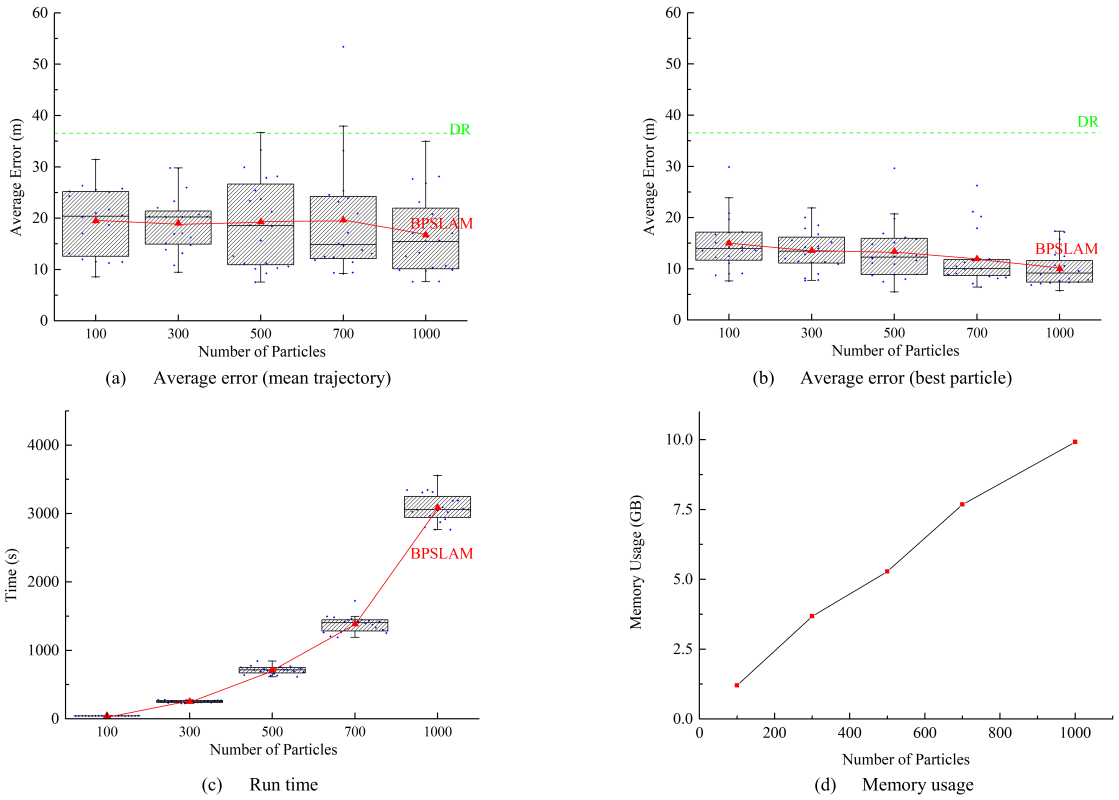


FIGURE 10. Effect of number of particles on BPSLAM. Each blue point represents the result of one test, and the red triangles indicate the mean of each group.

Normally the resampling reduces the standard deviation, but as not all the particles are included in particle resampling, the difference between particle positions may increase after resampling.

To test the stability of the methods, we considered the number of particles $N = 100, 300, 500, 700, 1000$, and executed 20 Monte Carlo tests for each case. As shown in Fig. 10 and 11, the accuracy and stability of both methods were improved by increasing the particle number. However, the proposed method had better accuracy and stability compared with BPSLAM when using the same particle set size. When $N = 1000$, the average error of the proposed method (5.27 m) was 47.2% lower than BPSLAM (9.99 m).

The run times of the methods are shown in Fig. 10(c) and Fig. 11(b), with the increment of the particle set size, the BPSLAM took more time in grid map update. The proposed method also took a substantial amount of computational resources in particle weighting as the particle increased, but

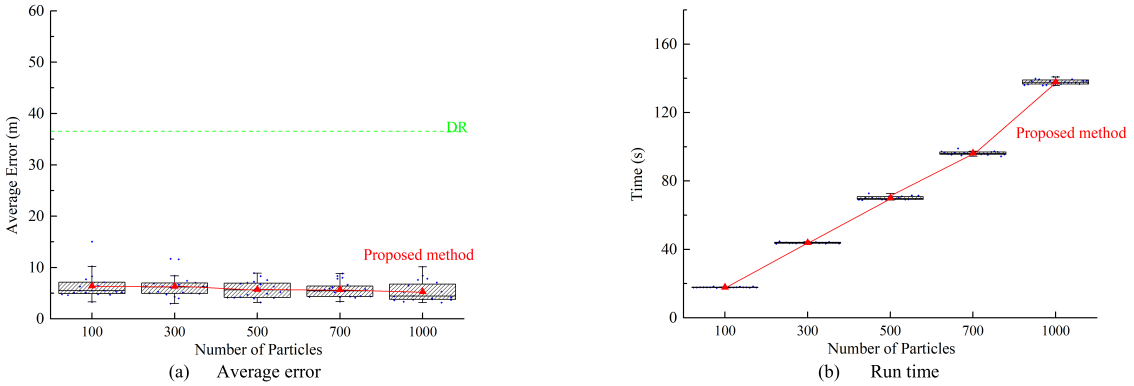


FIGURE 11. Effect of number of particles on the proposed method. Each blue point represents the result of one test, and the red triangles indicate the mean of each group.

FIGURE 12. CMBS200 multibeam sonar.

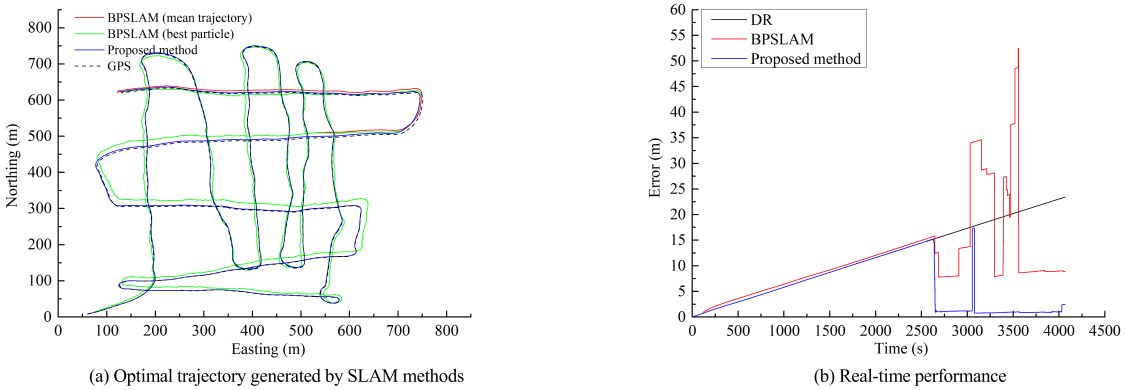


FIGURE 13. SLAM results for 300 particles.

it only took 137 s even in 1000 particles, which is 3.7% of the mission duration. By contrast, the average run time of BPSLAM was 3084 s when using 1000 particles.

The memory usage of the BPSLAM is shown in Fig. 10 (d). Compared with the proposed method that the memory usages were all 0.16 GB in different particle numbers, the BPSLAM took a substantial amount of memories during the mission even with the distributed map representation.

B. EXPERIMENTS ON FIELD DATA

The experiments using both accurate and unreliable odometer models were conducted. As shown in Fig. 12, the bathymetric data used in the experiments were collected by a shipborne T-SEA CMBS200 200 kHz multi-beam sonar with 173 beams spread equally over a sector of 120 degrees with a sampling frequency of 4 Hz. Navigation data including heading, pitch, roll angles were provided by the StarNeto XW-GI5651 inertial navigation system with an angular accuracy of 0.1°. The NovAtel ProPak-LB differential GPS provided the 20 cm positioning accuracy.

1) EXPERIMENTS USING ACCURATE ODOMETER

The parameters used in the simulations were: $m_e = 0.002$m, $\sigma_e = 0.005$m, $\sigma_s = 0.2$m, $N = 300$, $\omega \sim N(0, 0.04\ \text{m}^2)$.

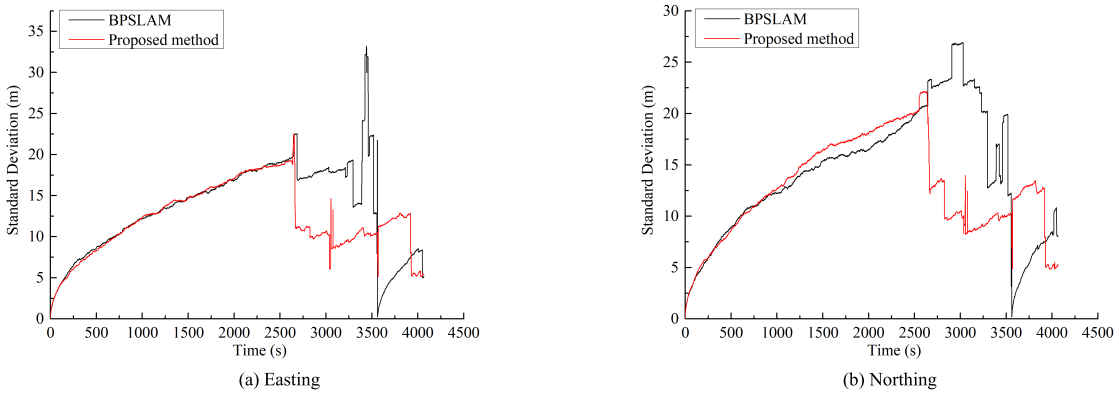


FIGURE 14. The standard deviation of the positions of the particles.

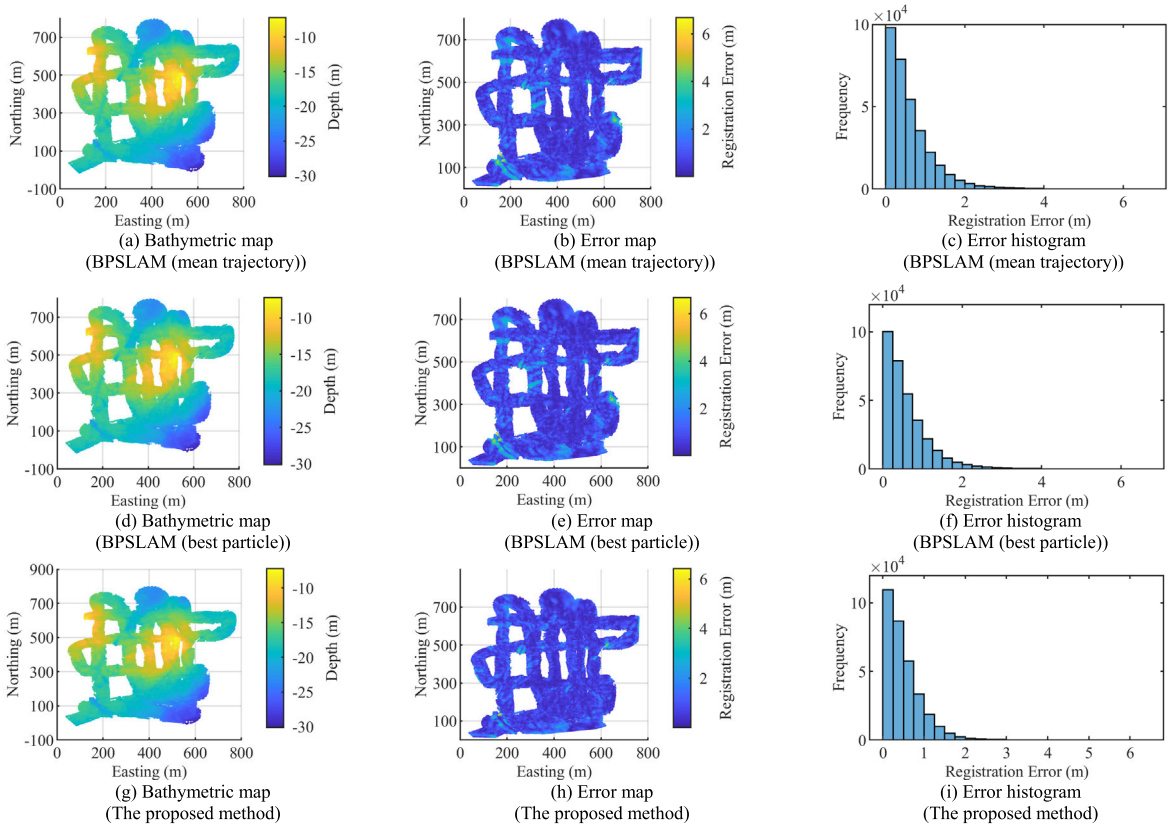
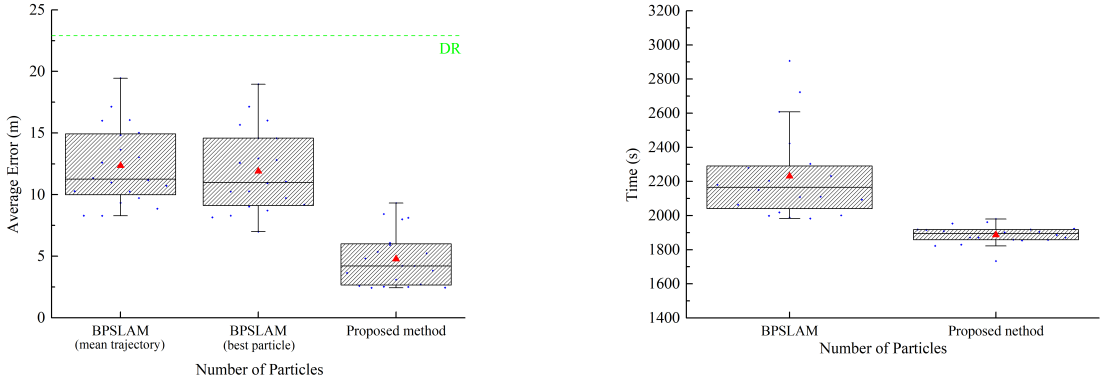


FIGURE 15. Bathymetric maps generated using different navigation solutions.

For the proposed method: $R = 20$m, $t_d = 500$, $t_c = 50$, $t_s = 500$. The resolution of the grid in BPSLAM was set as 1 m. In the proposed method, after submaps are generated, if the variance of the submap terrain depth is below 0.5 m^2, the submap is considered flat. As the particle weights calculated based on flat terrain are unreliable, the particles which at flat terrain location are not included in the resampling.

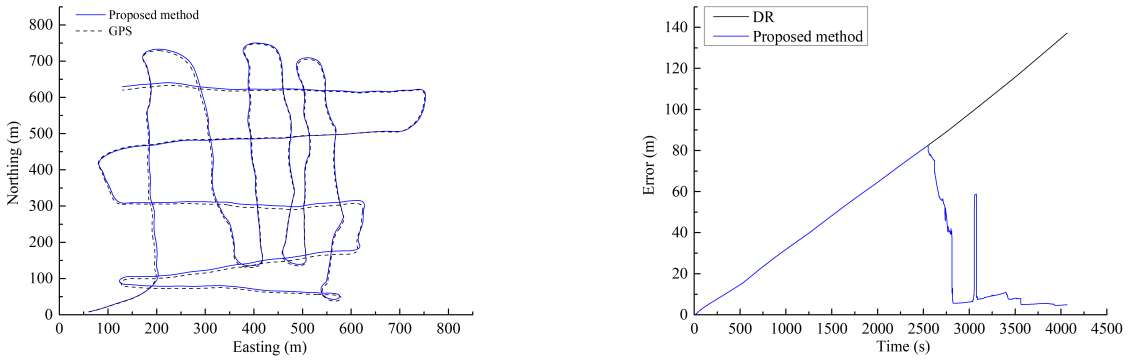
A group of results are demonstrated in Fig. 13. As the particles at the correct positions but with wrong trajectories were given high weights, the real-time average navigation errors of BPSLAM fluctuated during the mission, which even larger than the DR errors at some periods. The proposed method provided accurate and stable navigation results after 3250 s. The only fluctuation occurred in 3100 s. Because the graphs constructed by particles were not stable enough when the number of loop closures stored in particles was small.

The errors of BPSLAM (mean trajectory), BPSLAM (best particle), and the proposed method were 8.86, 8.16, and 2.43 m at the end of the mission, respectively. The proposed method was 70.22% more accurate than BPSLAM.



(a) Average error (b) Run time

FIGURE 16. The performance comparisons between BPSLAM and the proposed method.



(a) Optimal trajectory generated by SLAM methods (b) Real-time performance

FIGURE 17. SLAM results for 2000 particles.

The computation times of the BPSLAM and the proposed method were 2179.74 s and 1918.27 s, respectively. As the value of R was set as 20 m in the simulations to detect more loop closures, the proposed method spent more time in particle weighting. Hence the proposed method only 12% faster than BPSLAM.

As shown in Fig. 14, the BPSLAM faced the particle depletion problem after 3500 s, most of the particles were replaced by high weight particles. The standard deviation of the particle positions was greatly decreased. If no survival particle reflects the correct state, it will be hard for the BPSLAM to recover [25]. The proposed method has better robustness compared with BPSLAM.

As shown in Fig. 15, the mean trajectory and the best particle trajectory provided by BPSLAM had a similar accuracy, which was 0.63 m and 0.62 m, respectively. The mean registration error of the proposed method was 0.53 m, with 60.28% of the measurement errors remaining below 0.5 m. The proposed method has a better mapping performance. As the registration error of the map is sensitive to small position errors in areas with high terrain gradient, the registration errors of these areas are larger.

The results of 20 Monte Carlo tests on both methods are shown in Fig. 16, the average error of the proposed method was 4.77 m, which was 61.35% more accurate than BPSLAM using the mean trajectory (12.35 m) and 47.88% more accurate than BPSLAM using the best particle trajectory (9.15 m). The proposed method (1887 s) was also 15.4% faster than BPSLAM (2230 s).

2) EXPERIMENTS USING INACCURATE ODOMETER

To test the performance of the proposed method using an unreliable odometer, the odometer noise ω was set a big value: $\omega \sim U(-1.5\text{m}, 1.5\text{ m})$, the parameters of DR were set as: $m_e = 0.012\text{m}$, $\sigma_e = 0.01\text{m}$. As the BPSLAM is difficult to get available results under this condition, only the results of the proposed method are shown. The parameters of the proposed method were the same as the experiments using accurate odometer, except the number of particles was set as: $N = 2000$ to ensure the distribution of particles at the correct state. The Kullback-Leibler distance (KLD) sampling was used in the experiments to decrease the particle number after resampling and reduce the computational costs, more details of the KLD sampling can be found in [26].

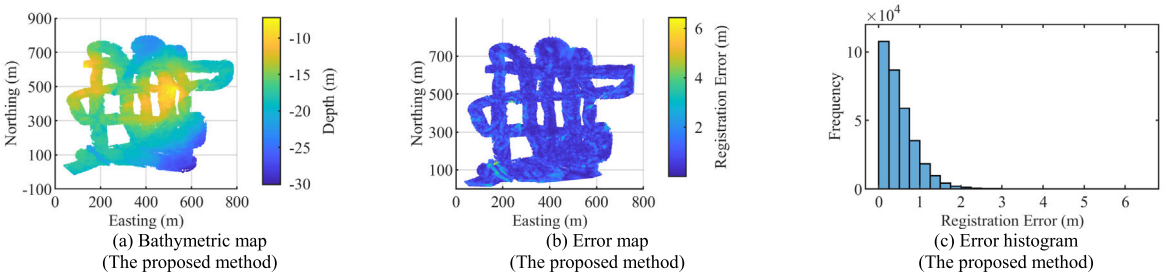


FIGURE 18. Bathymetric maps generated by the proposed method.

20 Monte Carlo tests were conducted, and a group of results is demonstrated in Fig. 17. Although the mean error of the proposed method (9.15 m) was 47.87% larger than the proposed method under small odometer error (4.77 m), it was still 25.91% more accurate than the BPSLAM under small odometer error (12.34 m). As the benefit of the KLD sampling, the run-time of the proposed method was 1930 s, which was similar to the run-time of the proposed method using 300 particles.

The bathymetric map generated by the proposed method is shown in Fig. 18, the mean registration error of the proposed method was 0.53 m, and 59.60% of the errors were below 0.5 m. The proposed method can still generate an accurate bathymetric map when the value of the odometer noise is large.

VI. CONCLUSION

We proposed two graph-based methods to improve the PF BSLAM performance when using an unreliable odometer model. The following conclusions can be drawn:

1) The graph-based particle map propagation method reduces the oscillation of the particle trajectories and improves the accuracy and stability of the particle filter.
2) The graph-based final map generation method can output the accurate vehicle trajectory online.
3) The playback experiments show that the proposed BSLAM method can provide accurate navigation and mapping results in different environments.

The proposed method needs sufficient particles to cover the correct states when the odometer model is unreliable. A large number of particles influence the real-time performance of the PF BSLAM. The use of GPU programming can improve the real-time performance of the proposed method. The implementation of the proposed method on an AUV will also be our future work.

REFERENCES

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.
[2] P. J. B. Sánchez, M. Papaelias, and F. P. G. Márquez, "Autonomous underwater vehicles: Instrumentation and measurements," *IEEE Instrum. Meas. Mag.*, vol. 23, no. 2, pp. 105–114, Apr. 2020.
[3] J. Melo and A. Matos, "Survey on advances on terrain based navigation for autonomous underwater vehicles," *Ocean Eng.*, vol. 139, pp. 250–264, Jul. 2017.
[4] Y. R. Petillot, G. Antonelli, G. Casalino, and F. Ferreira, "Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles," *IEEE Robot. Autom. Mag.*, vol. 26, no. 2, pp. 94–101, Jun. 2019.
[5] M. Rahmati, S. Karten, and D. Pompili, "SLAM-based underwater adaptive sampling using autonomous vehicles," in *Proc. MTS/IEEE OCEANS*, Charleston, SC, USA, Oct. 2018, pp. 1–7.
[6] F. Hidalgo and T. Bräunl, "Review of underwater SLAM techniques," in *Proc. 6th Int. Conf. Automat., Robot. Appl. (ICARA)*, Feb. 2015, pp. 306–311.
[7] G. Vallicrosa and P. Ridao, "H-SLAM: Rao-blackwellized particle filter SLAM using Hilbert maps," *Sensors*, vol. 18, no. 5, p. 19, May 2018.
[8] M. Massot-Campos, G. Oliver-Codina, and B. Thornton, "Laser stripe bathymetry using particle filter SLAM," in *Proc. MTS/IEEE OCEANS*, Marseille, France, Jun. 2019, pp. 1–7.
[9] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "A featureless approach to efficient bathymetric SLAM using distributed particle mapping," *J. Field Robot.*, vol. 28, no. 1, pp. 19–39, Jan. 2011.
[10] P. Norgren and R. Skjetne, "A multibeam-based SLAM algorithm for iceberg mapping using AUVs," *IEEE Access*, vol. 6, pp. 26318–26337, 2018.
[11] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric particle filter SLAM using trajectory maps," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1409–1430, Oct. 2012.
[12] C. Roman and H. Singh, "Improved vehicle based multibeam bathymetry using sub-maps and SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 3662–3669.
[13] A. Palomer, P. Ridao, and D. Ribas, "Multibeam 3D underwater SLAM with probabilistic registration," *Sensors*, vol. 16, no. 4, p. 23, Apr. 2016.
[14] N. Fairfield, G. Kantor, and D. Wettergreen, "Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Orlando, FL, USA, May 2006, pp. 3575–3580.
[15] V. Bichucher, J. M. Walls, P. Ozog, K. A. Skinner, and R. M. Eustice, "Bathymetric factor graph SLAM with sparse point cloud alignment," in *Proc. MTS/IEEE OCEANS*, Washington, DC, USA, Oct. 2015, pp. 1–7.
[16] T. Ma, Y. Li, R. Wang, Z. Cong, and Y. Gong, "AUV robust bathymetric simultaneous localization and mapping," *Ocean Eng.*, vol. 166, pp. 336–349, Oct. 2018.
[17] S. Thrun, "Particle filters in robotics," in *Proc. Uncertain. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 511–518.
[18] M. Massot-Campos, G. Oliver, A. Bodenmann, and B. Thornton, "Submap bathymetric SLAM using structured light in underwater environments," in *Proc. IEEE/OES Auton. Underwater Vehicles (AUV)*, Tokyo, Japan, Nov. 2016, pp. 181–188.
[19] T. Ma, Y. Li, Y. Zhao, Y. Jiang, Q. Zhang, and A. M. Pascoal, "Efficient bathymetric SLAM with invalid loop closure identification," *IEEE/ASME Trans. Mechatronics*, early access, Dec. 8, 2021, doi: 10.1109/TMECH.2020.3043136.
[20] N. Bore, I. Torroba, and J. Folkesson, "Sparse Gaussian process SLAM, storage and filtering for AUV multibeam bathymetry," in *Proc. IEEE/OES AUV*, Porto, Portugal, Nov. 2018, pp. 1–6.
[21] Y. Li, T. Ma, R. Wang, P. Chen, and Q. Zhang, "Terrain correlation correction method for AUV seabed terrain mapping," *J. Navigat.*, vol. 70, no. 5, pp. 1062–1078, Sep. 2017.

[22] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2O: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2011, pp. 3607–3613.
[23] G. Mei, N. Xu, and L. Xu, "Improving GPU-accelerated adaptive IDW interpolation algorithm using fast kNN search," *SpringerPlus*, vol. 5, no. 1, p. 1389, Dec. 2016.
[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[25] N. Kwak, I.-K. Kim, H.-C. Lee, and B.-H. Lee, "Analysis of resampling process for the particle depletion problem in FastSLAM," in *Proc. 16th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Jeju, South Korea, Aug. 2007, pp. 200–205.
[26] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *Int. J. Robot. Res.*, vol. 22, no. 12, pp. 985–1003, Dec. 2003.

QIANYI ZHANG received the bachelor's degree in shipping and marine engineering from Harbin Engineering University, in 2019, where he is currently pursuing the master's degree with the Science and Technology on Underwater Vehicles Laboratory (STUVL). His research interest includes bathymetric particle filter simultaneous localization and mapping (SLAM).

YE LI received the bachelor's degree (Eng.) in shipping and marine engineering from Harbin Engineering University, in 2001, the master's degree from the STUVL, Harbin Engineering University, in 2005, and the Ph.D. degree, in 2007. His main research interest includes navigation and control of autonomous underwater vehicle.

TENG MA received the B.S. and Ph.D. degrees in shipping and marine engineering from Harbin Engineering University, in 2015 and 2019, respectively. He is currently a Postdoctoral Researcher with Harbin Engineering University. His research interests include underwater SLAM, terrain-aided navigation, and path planning.

ZHENG CONG received the bachelor's degree in shipping and marine engineering from Harbin Engineering University, in 2016, and the master's degree from the STUVL, Harbin Engineering University, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include underwater SLAM, underwater multiple-AUV SLAM, terrain-aided navigation, and path planning.

WENJUN ZHANG received the bachelor's degree from Henan Polytechnic University, Jiaozuo, China, in 2012, and the M.Sc. degree in cartography and geographic information engineering from Shandong University of Science and Technology, Qingdao, China, in 2015. She is currently pursuing the Ph.D. degree in marine engineering with the STUVL, Harbin Engineering University. Her general research interests include seafloor terrain matching navigation and underwater SLAM.