IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 8, 2021, accepted May 3, 2021, date of publication May 10, 2021, date of current version May 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078854

Bathymetric Particle Filter SLAM Based on Mean Trajectory Map Representation

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 Postdoctoral Science Foundation of China under Grant 2020M670887.

ABSTRACT To obtain independent navigation results for autonomous underwater vehicles (AUVs) and construct high-resolution consistent seabed maps, a particle filter-based bathymetric simultaneous localization and mapping (BSLAM) method with the mean trajectory map representation is proposed. To reduce the computational consumption, particles only keep the current estimated position of the vehicle, while all historical states of the vehicle are stored in the mean trajectory map. Using this set-up, only the weights of the particles which closed to the mean trajectory map are calculated with newly collected bathymetric data. A hierarchical clustering procedure is also discussed to identify invalid loop closures. The performance of the proposed method is validated using both the simulated data and the field data collected from sea trails. The results demonstrate that the proposed method is 50% more accurate and 50% faster than a state-of-the-art particle filter-based BSLAM method, and it has similar accuracy but 30% faster compared with a graph-based BSLAM method.

INDEX TERMS Autonomous underwater vehicle, bathymetric simultaneous localization and mapping, particle filter, hierarchical clustering.

I. INTRODUCTION

Autonomous underwater vehicles (AUVs) have been used for a variety of tasks, including oceanographic surveys, demining, and seabed mappings [1]. When an AUV operates close to the seabed, it can collect high-resolution bathymetric data [2]. But a consistent seabed map can only be constructed with accurate navigation results of the vehicle [3]. Water can greatly attenuate or even stop Global Positioning System (GPS) signals, making GPS incompetent for the underwater navigation of AUVs. Alternatively, long and ultra-short baseline acoustic positioning systems provide navigation results with bounded localization errors but restrict operations to a relatively confined area around acoustic units [4]. Dead reckoning (DR) systems are common navigation methods to locate AUVs with short-term accurate positioning results, but the resulting navigation error will gradually accumulate over time, thus rendering it unsuitable for long-term underwater positioning [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Rui-Jun Yan.

Simultaneous localization and mapping (SLAM) enables underwater vehicles to construct or update a map of an unknown environment while positioning themselves using the map built [6], [7]. Most SLAM methods estimate the difference between states of an AUV by calculating the similarity of associated features. However, underwater features such as peaks and valleys are sensitive to viewing angles and lack spatial density in underwater environments, making feature extraction or identification difficult. Thus, featureless SLAM methods are developed to solve bathymetric SLAM problems. Many breakthroughs have been made in featureless BSLAM methods using the particle filtering theory [8]–[11]. In the PF-BSLAM, if the particle does not store all the historical positions of the vehicle, a map representation is needed to store and update the maps. The grid map representation which based on the distributed particle mapping (DPM) used in [8], [11] that each grid stores estimated terrain depths and the corresponding particle IDs. The particle can be weighted by accessing the grids, and the particle's map can be constructed through the particle ancestry. However, the resolution of the grid influences the quality of the map, and

a substantial amount of memory resources for storage is still needed to store the grid map.

The trajectory map representation proposed by Barkby *et al.* [9] is also based on the DPM. Using this set-up, the trajectory map maintains each particle's map by storing the particle ancestry tree and a list of bathymetric observations shared by all particles. Each particle stores a trajectory since the last execution of the resampling. The entire trajectories of the particles can be reconstructed by accessing the corresponding particles and their parent IDs from the ancestry tree. Compared with the grid map representation, this method reduces the memory usage, but the computational efficiency limits its applications as the Gaussian process regression used in particle weighting needs a substantial amount of computational resources for terrain prediction.

Contribution: In the proposed PF-BSLAM method, the whole trajectory of the vehicle is estimated in two parts: particles only keep the vehicle's current estimate positions to reduce the storage consumption, and all historical positions of the vehicle are stored in a mean trajectory map. To detect the loop closure between a particle's current estimated positions and historical positions of the vehicle, a temporary map is constructed in particles closed to possible overlapping areas use the inverse distance weighting (IDW) interpolation algorithm. Loop closure consistency determination (LCCD) method is designed to identify invalid loop closures. Compared with the grid and trajectory map representations, the proposed method is more efficient and accurate because the use of submap matching allows more bathymetric observations to be matched in particle weighting. The performance of the newly proposed BSLAM method has been proved using both simulated and field data.

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

Many breakthroughs have been made in bathymetric SLAM. Roman and Singh [12] decompose a map into submaps, which are matched using the iterative closest point algorithm. Although experimental results confirm the higher accuracy compared with those of traditional navigation methods, this method cannot handle errors inside submaps. Palomer *et al.* [13] use the probabilistic iterative closest point algorithm for 3D underwater SLAM to obtain consistent maps, but only the relative positions of the patches are corrected, the internal patch error cannot be modified, and this method cannot be used online due to the high computational cost. Stuckey [14] achieves BSLAM using navigation cells, which are patches stored as grid submaps along the AUV trajectory. When the AUV passes through a previously explored patch, the new cell is compared to the original one, and the observations are updated accordingly. Although simulation results demonstrate the effectiveness of this method, a variety of topographic information is required, and the AUV should pass close to the navigation cells. Ma *et al.* [15] use graph-based BSLAM to identify invalid loop closures by a multi-window consistency method (MCM) that demonstrates robustness in simulations. However, the MCM needs plenty of computational resources, especially when dealing with a large number of loop closures. Another graph-based BSLAM proposed by [16] using sparse Gaussian processes as the bathymetry representation, but it has the accuracy problem if the smooth seafloor prior is not valid.

We proposed a BSLAM method using the PF theory, a non-parametric Bayesian filter that represents the state distribution by a set of samples [17]. The traditional PF framework with a particle set size (N) and a total mission time (T) is described in Algorithm 1. In particle filter, the prior estimation of the vehicle state is approximated by the propagation of all the positions of the particles $S_n^{(t)} \sim p(S_n^{(t)}|u_t, S_n^{(t-1)})$. After the particle propagation procedure, the particles are weighted based on how well the observed swaths matched with the previous bathymetric observations at position $S_n^{(t)}$, $w_n^{(t)} \sim p(z_t|S_n^{(t)})$. Normally, as the prior estimation $S_n^{(t)}$ is already calculated in particle propagation, z_t (the bathymetric observations) are only considered in particle weighting [8], [9], [18]. After particle weighting, the posterior estimation of the vehicle position is approximated by all the positions of the particles after resampling.

Algorithm 1 Particle Filter (PF) Framework

for time $t = 1$ to T **do**
  Input particle set at time $t-1$ ($P_{(t-1)}$), control at time t ($u_{(t)}$), bathymetric observation collected using a multibeam sonar at time t ($z_{(t)}$).
  for particle $n = 1$ to N **do**
    Propagate the position of particle n: $S_n^{(t)} \sim p(S_n^{(t)}|u_t, S_n^{(t-1)})$.
  end for
  for particle $n = 1$ to N **do**
    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

III. PARTICLE UPDATE

By using the high precision fiber optic gyrocompass and altimeter, the impact of the systematic errors of depth, heading, pitch, and roll on the vehicle state is virtually negligible, which means that the states of particles only include the estimated positions of the vehicle on the horizontal plane [8]–[11]. Thus, a particle set $P_{(t)}$ with N particles is defined as

$$P_{(t)} = \begin{bmatrix} S_1^{(t)} & \cdots & S_N^{(t)} \\ w_1^{(t)} & \cdots & w_N^{(t)} \end{bmatrix}, \tag{1}$$

where $S_n^{(t)}$ is the horizontal position of the particle n at time t, and $w_n^{(t)}$ represents the particle's weight. All the particles are initialized with the ground-truth location of the AUV obtained by the GPS or acoustic positioning methods.

A. MAP STRUCTURE

The central position of the particle set at time t ($S_{\mathrm{p}}^{(t)}$) is considered as the estimated position of the vehicle, and the mean trajectory map consists of all historical estimated positions. As illustrated in Fig. 1, the bathymetric map can be constructed by combining the mean trajectory and the bathymetric observations at corresponding moments. The storage requirements of the map representations are detailed in Table 1.

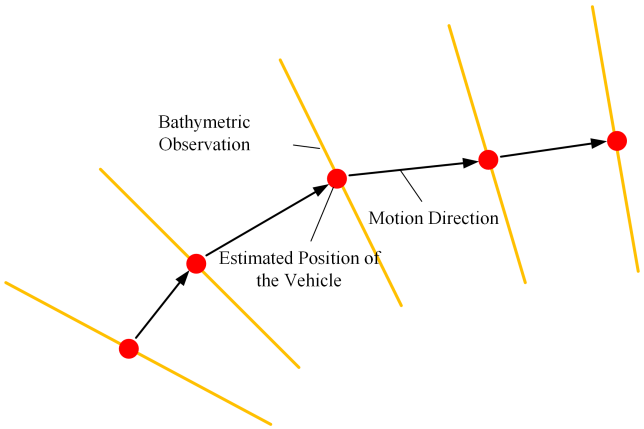


FIGURE 1. Mean trajectory map with five estimated positions of the vehicle and five bathymetric observations.

TABLE 1. Comparison of memory requirements. (a) Storage requirements of the mean trajectory map. (b) Storage requirements for trajectories stored in particles. The memory requirements of the current AUV position, complete trajectory, and bathymetric observations are denoted as *mi*, *mj*, and *mk*, respectively. The mean trajectory representation from the proposed method requires less storage because *mi* is much smaller than *mj*.

(a)

Parameter	Memory requirement
N particles with current vehicle position	$mi \times N$
Complete mean trajectory	mj
Bathymetric observations	mk
Total	$mi \times N + mj + mk$

(b)

Parameter	Memory requirement
N particles with complete trajectory	$mj \times N$
Bathymetric observations	mk
Total	$mj \times N + mk$

B. PROPAGATION OF THE PRATICLES

To prevent the particle depletion problem [19]–[22], the Jittered Bootstrap discrete-time model [20], [21] is applied to propagate the particle position at each moment:

$$S_n^{(t)} = S_n^{(t-1)} + u_{(t)} + \omega + \mathcal{N}(0,\sigma_m^2), \tag{2}$$

where $u_{(t)}$ is the control vector of the vehicle at time t, ω represents a process noise with Gaussian distribution, $\mathcal{N}(0,\sigma_m^2)$ is a Gaussian noise to recover the particle diversity in the state space after resampling, with σ_m being the standard deviation:

$$\sigma_m^2 = \sigma_c^2 + Ae^{\frac{-t_r}{\mathrm{C}}}, \tag{3}$$

where t_r is the time interval since the last execution of the particle resampling, A is an attenuation factor (when t_r is below a predefined threshold B, $A = 0$), σ_c^2 is a constant variance, and C is a constant. At the beginning of the mission, t_r is set as a large value, and hence, $\sigma_m^2 \approx \sigma_c^2$. After resampling, t_r restarts from 1. When $t_r \geq$ B, the particles are added a decaying variance to ensure their dispersion in the state space, as shown in Fig. 2.

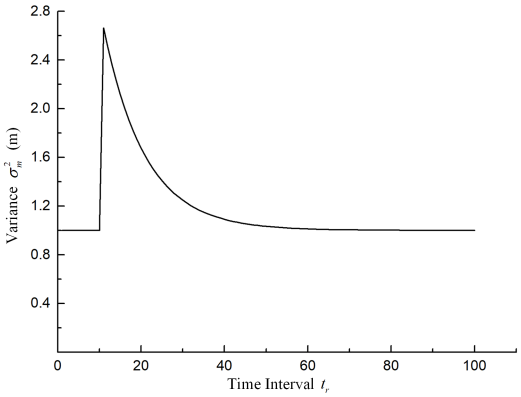


FIGURE 2. Variance σ_m^2 according to resampling period t_r, for $\sigma_c^2 = 1$, A = 5, B = 10, and C = 10.

C. PARTICLE WEIGHTING

The particle weighting process is detailed in Algorithm 2. As the mean position of the particles is considered as the posterior position of the vehicle, all the particles are included in resampling, which means all the particles are weighted in particle weighting. If the minimum horizontal distance between the position of particle n and the mean trajectory before time $t - t_l$ (t_l is a threshold) exceeds a radius R, $w_n^{(t)}$ is set as $1/N$, or the particle is considered as overlapped with the mean trajectory, and $w_n^{(t)}$ is calculated based on submap matching.

In the mean trajectory map, we denote a submap including m measurement points collected between the time interval $[t - T_{\mathrm{p}} + 1,\ t]$ as $PC_{m\times 3}$, and the corresponding submap associated with the particle n can be given by:

$$Pmap = PC_{m\times 3} + N_{m\times 1}\begin{bmatrix}\vec{V}_{ec}\\ 0\end{bmatrix}_{1\times 3}^{\mathrm{T}}, \tag{4}$$

where

$$\vec{V}_{ec} = S_n^{(t)} - S_{\mathrm{p}}^{(t)}, \tag{5}$$

$N_{m\times 1}$ is the matrix that all elements are 1. Similarly, the mean trajectory submap ($Mmap$) is the mean trajectory map

Algorithm 2 Particle Weighting

if minimum horizontal Euclidean distance of particle n to mean trajectory map is below R.

Generate a temporary submap (*Pmap*) and a submap of the mean trajectory map (*Mmap*).

Interpolate *Pmap* by using the inverse distance weighted interpolation.

Calculate the mean squared error (MSE, denoted as *mse*) of the overlapping area between submaps.

if $mse < threshold$

Weight particle n based on similarity of two submaps.

end

end

between $[t_{lp} - T_m/2, t_{lp} + T_m/2]$, where T_m is the number of swaths in *Mmap* and t_{lp} is the time that corresponds to the historical positions of the vehicle that closest to $S_n^{(t)}$, as illustrated in Fig. 3.

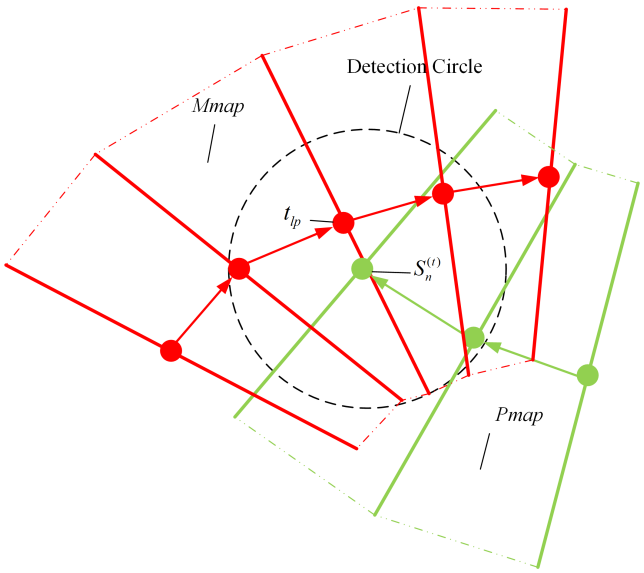


FIGURE 3. Loop closure detection in particle *n*. The *Mmap* and *Pmap* containing five and three swaths, respectively.

The similarity between *Pmap* and *Mmap* is quantified using the mean squared error (MSE) [23]:

$$mse = \frac{1}{I}\sum_{i=1}^{I}(Z_{pmap}^{(i)} - Z_{mmap}^{(i)})^2, \tag{6}$$

where I is the number of valid measurements, $Z_{mmap}^{(i)}$ is the terrain depth of *i-th* measurement point in the *Mmap*, and $Z_{pmap}^{(i)}$ is the depth obtained using IDW interpolation at the same position as $Z_{mmap}^{(i)}$ in *Pmap*, as shown in Fig. 4. If I is greater than a given threshold while *mse* is less than another given threshold, $S_n^{(t)}$ is considered as a possible current location of the vehicle and the particle n is weighted. Otherwise, the particle weight is set as $1/N$.

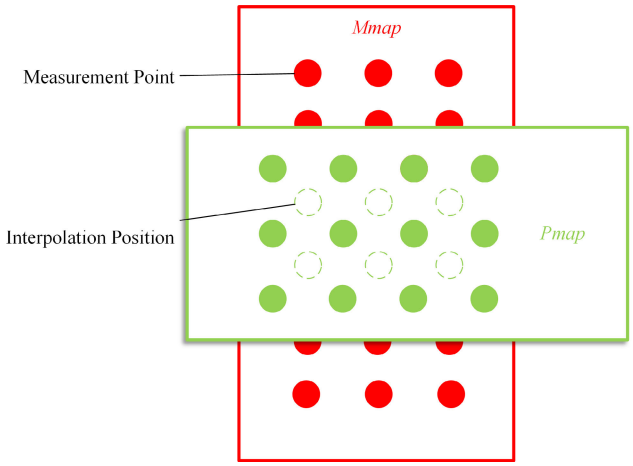


FIGURE 4. Submap interpolation based on IDW.

Assumed that the difference between the estimated bathymetric measurement at the same positions of *Pmap* and *Mmap* follows a Gaussian distribution [8], $w_n^{(t)}$ is calculated as:

$$w_n^{(t)} = \frac{1}{I}\sum_{i=1}^{I} p(Z_{pmap}^{(i)} = Z_{mmap}^{(i)}), \tag{7}$$

$$p(Z_{pmap}^{(i)} = Z_{mmap}^{(i)}) = \frac{\exp(\frac{-(Z_{pmap}^{(i)} - Z_{mmap}^{(i)})^2}{2\sigma_s^2})}{\sqrt{2\pi\sigma_s^2}}, \tag{8}$$

where σ_s^2 is the measurement variance of the sensor, I is the number of valid measurement points in *Pmap*, respectively.

As the flat terrain may lead to a biased estimate on the particle weights, the particle weight indexation is used to improve the robustness of the proposed BSLAM [17], [24]:

$$w_n^{(t)} = (w_n^{(t)})^{\alpha}, \tag{9}$$

where α is a function of *Pmap* or *Mmap* related to the terrain information

$$\alpha = \min(\alpha_{pmap}, \alpha_{mmap}). \tag{10}$$

Specifically,

$$\alpha_{pmap} = \frac{(\sigma_s^2 + \sigma_{pmap}^2)\hat{\delta}_{pmap}^2}{(\sigma_s^2 + \sigma_{pmap}^2)(\hat{\delta}_{pmap}^2 + \sigma_{pmap}^2) + \sigma_s^2\sigma_{pmap}^2}, \tag{11}$$

where σ_{pmap}^2 is the variance of bathymetric measurements in *Pmap*, $\hat{\delta}_{pmap}^2$ represents the terrain information expressed using the fluctuation variance of bathymetric measurements in *Pmap* [25]:

$$\hat{\delta}_{pmap}^2 = \frac{1}{I}\sum_{i=1}^{I}[Z_{pmap}^{(i)} - \frac{1}{I}\sum_{i=1}^{I} Z_{pmap}^{(i)}]^2 - \sigma_{pmap}^2. \tag{12}$$

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

$$N_{\text{eff}} = \frac{1}{\sum_{n=1}^{N}[\tilde{w}(n)]^2} \tag{13}$$

is less than two thirds the number of particles, the particles are resampled [18], where $\tilde{w}(n)$ is the normalized weight of particle n, and the n-th loop closure is defined as

$$L_{(n)} = \left\{ \vec{m}_{(n)}, [t_{loop}^{(n)}, t_{end}^{(n)}] \right\}, \tag{14}$$

where $\vec{m}_{(n)}$ is the vector from $S_{\mathrm{p}}^{(t)}$ (before resampling) to $S_{\mathrm{p}}^{(t)}$ (after resampling), $t_{loop}^{(n)} = t_{lp}$ and $t_{end}^{(n)} = t$ are the start and end times of the n-th loop closure, respectively.

IV. MEAN TRAJECTORY MAP

The validity of a loop closure detected at time t is judged by the loop closure consistency determination (LCCD) method, and the corresponding trajectory will only be updated if the loop closure is considered valid.

A. MAP UPDATE

To avoid the repeated update of the same part on the mean trajectory caused by various loop closures, the loop closure detected at time t ($L_{(n)}$) only updates the mean trajectory in the time period $U_{(n)}$

$$U_{(n)} = l_{(n)} - l_{(1:n-1)}^{h} \cap l_{(n)}, \tag{15}$$

in which,

$$l_{(1:n-1)}^{h} = l_{(1)} \cup l_{(2)} \ldots \cup l_{(n-1)}, \tag{16}$$

and

$$l_{(n)} = [t_{loop}^{(n)}, t_{end}^{(n)}]. \tag{17}$$

The corrections of the estimated position of the vehicle (denoted as $\Delta\vec{X}_{(t)}$) corresponding to $U_{(n)}$ are given by the terrain correlation correcting method [26]:

$$\Delta\vec{X}_{(t)} = \left(a_t / a_{t_b}\right) \cdot \vec{m}_{(n)} \quad t \in U_{(n)}, \tag{18}$$

$$a_i = \sum_{j=t_a}^{i} \frac{1}{p_{(S_{\mathrm{p}}^{(j)} | S_{\mathrm{p}}^{(j-1)})}}, \tag{19}$$

where t_a and t_b are defined as the minimum and maximum values in $U_{(n)}$, $p_{(S_{\mathrm{p}}^{(j)} | S_{\mathrm{p}}^{(j-1)})}$ is the interframe motion uncertainty of the AUV. $p_{(S_{\mathrm{p}}^{(j)} | S_{\mathrm{p}}^{(j-1)})}$ can be estimated using the similarity between the predicted and actual bathymetric observations. However, the terrain prediction methods such as Gaussian process regression require a substantial amount of computational resources. For an accurate inertial navigation system, we assume that the interframe motion uncertainty in $[t_a, t_b]$ is approximately constant with value p. Hence,

$$a_i = \sum_{j=t_a}^{i} \frac{1}{p_{(S_{\mathrm{p}}^{(j)} | S_{\mathrm{p}}^{(j-1)})}} \approx \sum_{j=t_a}^{i} \frac{1}{\mathrm{p}}, \tag{20}$$

and (18) can be simplified as

$$\Delta\vec{X}_{(t)} = \left(a_t / a_{t_b}\right) \cdot \vec{m}_{(n)}$$

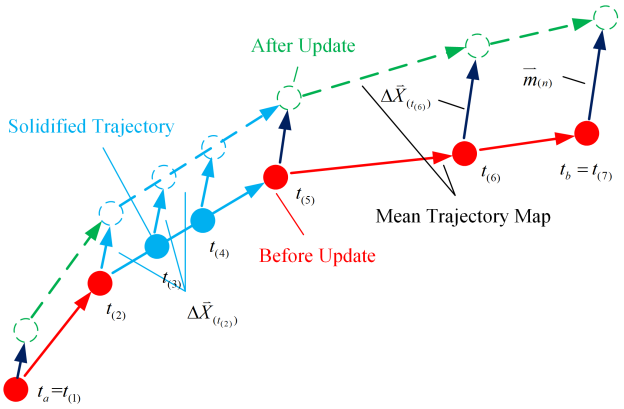


FIGURE 5. Trajectory update using $\vec{m}_{(n)}$ for $U_{(n)} = [t_{(1)}, t_{(2)}] \cup [t_{(5)}, t_{(7)}]$ and solidified period $(t_{(2)}, t_{(5)})$. As $t_{(2)} \in U_{(n)}$ is the time closest to $(t_{(2)}, t_{(5)})$, $\Delta\vec{X}_{(t_{(2)})} = \Delta\vec{X}_{(t_{(3)})} = \Delta\vec{X}_{(t_{(4)})}$.

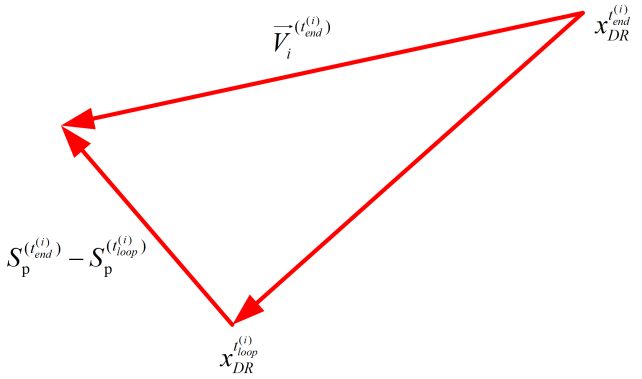


FIGURE 6. The calculation of $\tilde{V}_i^{(t)}$ for $t = t_{end}^{(i)}$.

$$\approx \left(\sum_{j=t_a}^{t} \frac{1}{\mathrm{p}} / \sum_{j=t_a}^{t_b} \frac{1}{\mathrm{p}} \right) \cdot \vec{m}_{(n)}$$
$$= [(t - t_a + 1)/(t_b - t_a + 1)] \cdot \vec{m}_{(n)} \quad t \in U_{(n)}. \tag{21}$$

If $U_{(n)}$ is not a continuous period, as the relative positions of the vehicle in each continuous period in $[t_a, t_b] - U_{(n)}$ are solidified, the positions in each continuous period in $[t_a, t_b] - U_{(n)}$ are updated by the same vector

$$\Delta\vec{X}_{(t)} = \Delta\vec{X}_{(t_{(i)})}, \tag{22}$$

where $t_{(i)} \in U_{(n)}$ is the earliest time closest to each continuous period in $[t_a, t_b] - U_{(n)}$, as illustrated in Fig. 5.

B. LCCD

The assumption of the loop closure consistency determination (LCCD) method is that the valid loop closures are consistent in the update of the trajectory, while the invalid loop closures have completely contradictory views on the map update. Therefore, the invalid loop closures can be detected by clustering.

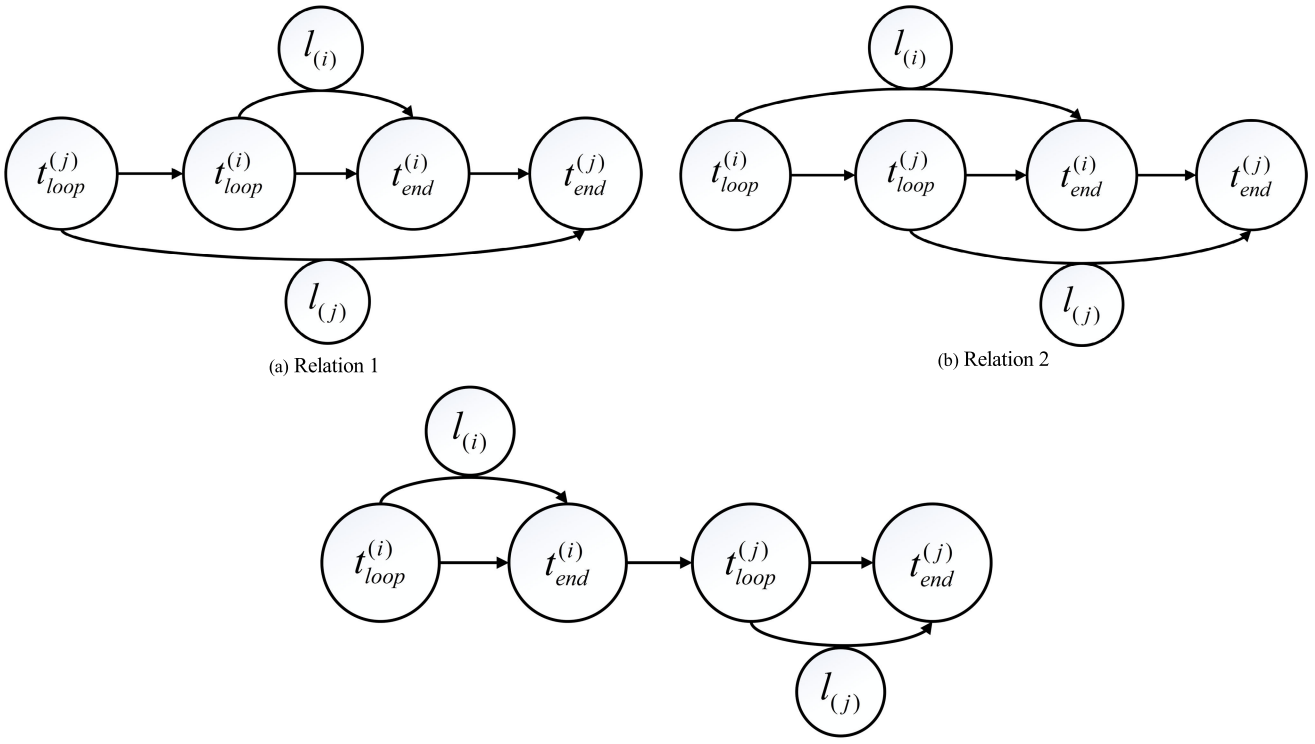


FIGURE 7. The relations between $L_{(i)}$ and $L_{(j)}$, for $i < j$.

1) EUCLIDEAN MATRIX

The $M \times M$ Euclidean matrix D defined as

$$D_{ij} = \begin{cases} d\,(i,j) & i \neq j \\ 0 & i = j \end{cases} \tag{23}$$

is used as the basis of clustering, where M is the number of loop closures input to the LCCD, and $d(i,j)$ is the distance between the DR correction vectors of loop closure i and loop closure j at the corresponding time, hence

$$d(i,j) = d(j,i). \tag{24}$$

The DR correction vector of loop closure i (i is ordinal) at time t (denoted as $\overrightarrow{V_i}^{(t)}$) is defined as

$$\overrightarrow{V_i}^{(t)} = \frac{t - t^{(i)}_{loop}}{t^{(i)}_{end} - t^{(i)}_{loop}}(x^{t^{(i)}_{loop}}_{DR} - x^{t^{(i)}_{end}}_{DR} + S_p^{(t^{(i)}_{end})} - S_p^{(t^{(i)}_{loop})}) \quad t \in l_i, \tag{25}$$

where x^t_{DR} is the position of the vehicle estimated by DR, as illustrated in Fig. 6.

Assuming $i < j$, there are three relations between $L_{(i)}$ and $L_{(j)}$, as illustrated in Fig. 7.

For relation 1 ($t^{(j)}_{loop} < t^{(i)}_{loop}$):

$$d(i,j) = \left\| (\overrightarrow{V_j}^{(t^{(i)}_{end})} - \overrightarrow{V_j}^{(t^{(i)}_{loop})}) - \overrightarrow{V_i}^{(t^{(i)}_{end})} \right\|_2. \tag{26}$$

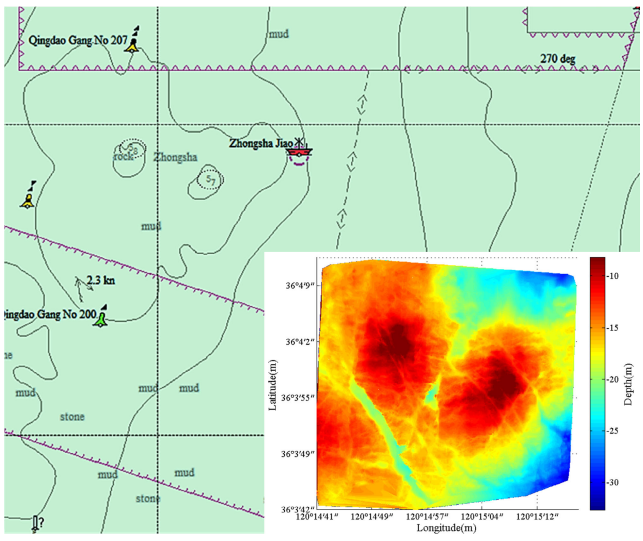


FIGURE 8. Sea trail region corresponding to experimental data.

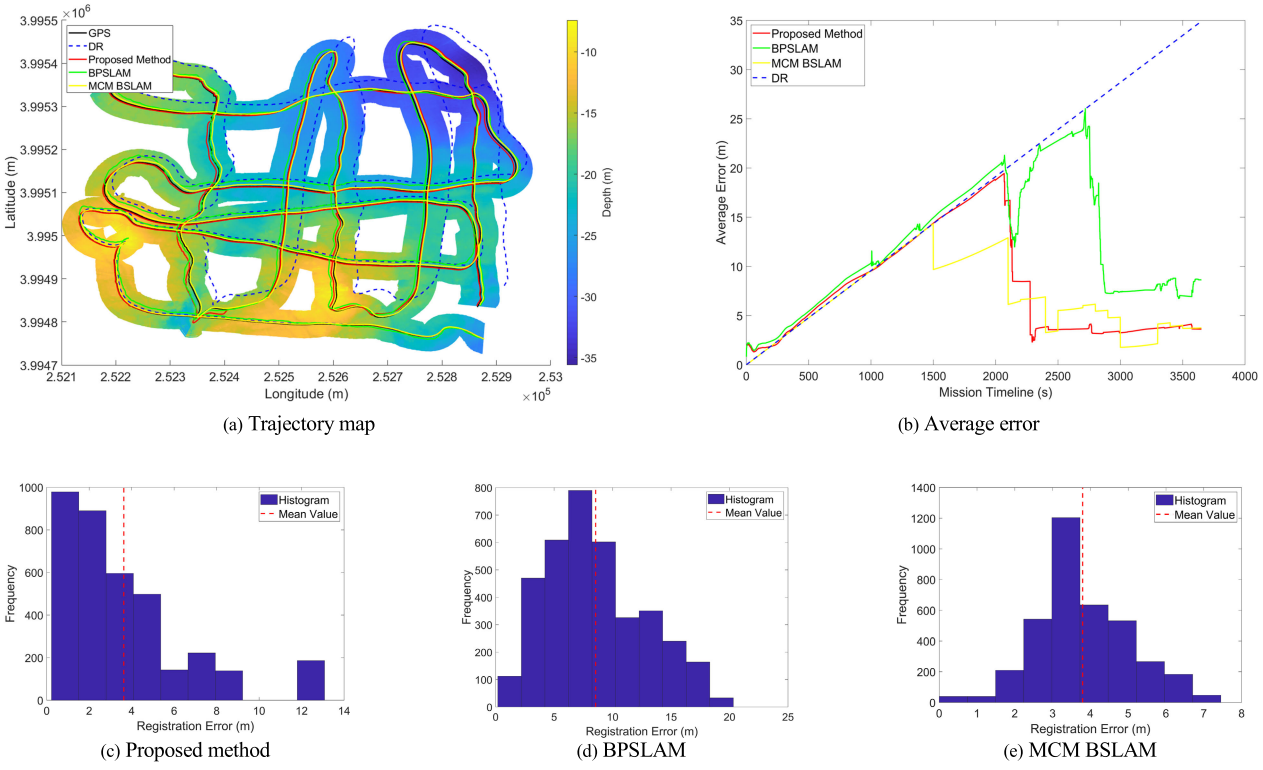
For relation 2 ($t^{(i)}_{loop} \leq t^{(j)}_{loop} \leq t^{(i)}_{end}$):

$$d(i,j) = \left\| (\overrightarrow{V_i}^{(t^{(i)}_{end})} - \overrightarrow{V_i}^{(t^{(j)}_{loop})}) - \overrightarrow{V_j}^{(t^{(i)}_{end})} \right\|_2. \tag{27}$$

For relation 3 ($t^{(j)}_{loop} > t^{(i)}_{end}$), $d(i,j)$ is set as zero because $l_{(i)} \cap l_{(j)} = \emptyset$.

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

FIGURE 9. Experimental equipment for data collection.



(a) Trajectory map (b) Average error

(c) Proposed method (d) BPSLAM (e) MCM BSLAM

FIGURE 10. SLAM results for 400 particles. The map in (a) was generated by the mean trajectory and represents the final result of the proposed BSLAM method. The error in (b) was calculated with respect to GPS data. The error distribution of the trajectory points shown in (c)–(e) have the mean values indicated by the red dashed lines.

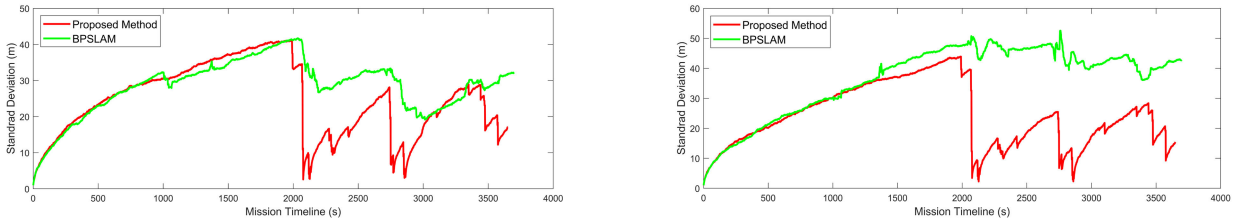
2) HIERARCHICAL CLUSTERING

After generating Euclidean matrix D, hierarchical clustering is applied to judge the validity of $L_{(n)}$. The hierarchical clustering method divides data into various layers according to the similarity of the data sets, it does not need a predefined number of clusters, rendering it suitable for LCCD. Detailed explanations of the hierarchical clustering can be found in [27].

If $L_{(n)}$ is a single cluster under a predefined threshold, $L_{(n)}$ is considered invalid. The mean trajectory is not updated, and all the particle positions are restored to the estimated position of the vehicle before resampling. If $L_{(n)}$ is not a single cluster under the threshold, $L_{(n)}$ is considered valid, and the map update method described in Section IV is applied to the mean trajectory map.

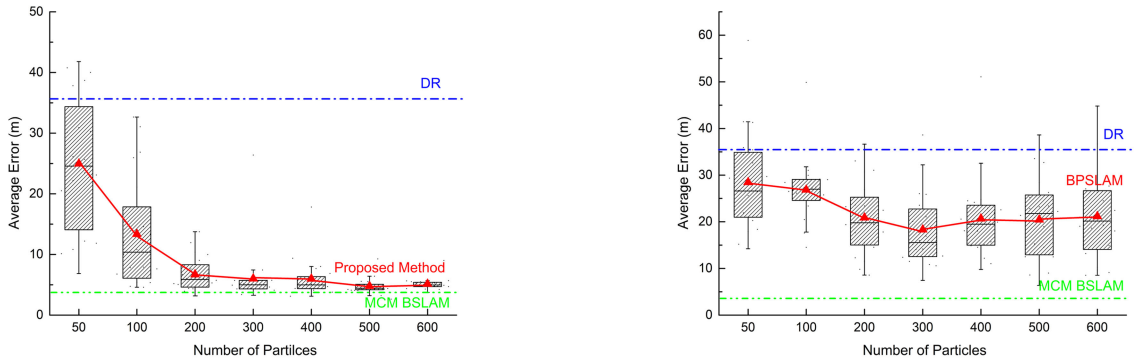
V. PLAYBACK EXPERIMENTS

Two playback experiments were conducted using simulated data and field data, in which the latter was collected from a sea trial around Zhongsha Reef, China (shown in Fig. 8). All datasets were corrected considering the influence of waves on the measurements before experiments.



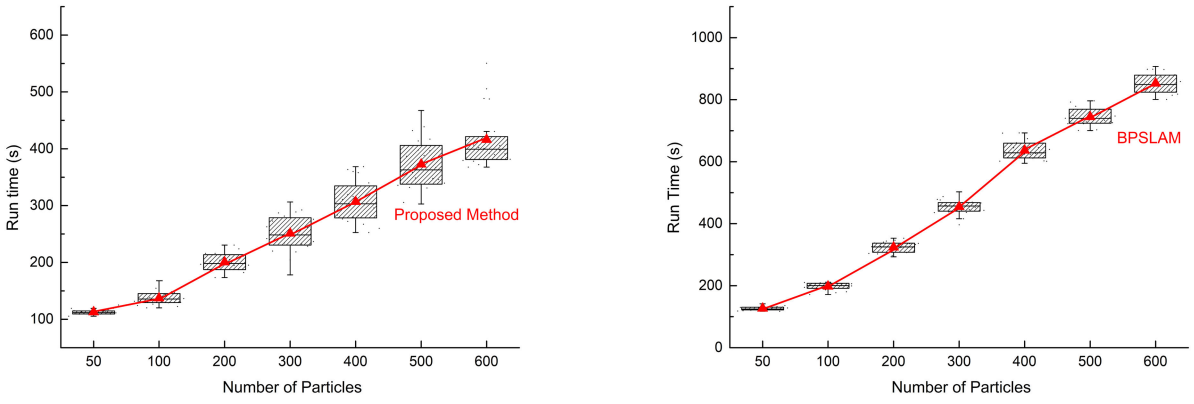
(a) X-direction (b) Y-direction

FIGURE 11. The particle standard deviation of BPSLAM and proposed BSLAM.



(a) Proposed method (b) BPSLAM

FIGURE 12. Effect of number of particles on SLAM. Each black point represents the result of one test, and the red triangles indicate the average error of each group.



(a) Proposed method (b) BPSLAM

FIGURE 13. Computation time of proposed BSLAM and BPSLAM. The red line represents the average computation time.

The graph-based MCM BSLAM [15] and a traditional PF BSLAM (denoted as BPSLAM) were used for comparisons. For BPSLAM, each particle stores its own trajectory, and the particles are weighted by submap matching. The simulation framework was implemented in MathWorks MATLAB, and the simulations were executed on a computer with Intel Core i7-8750H CPU and 16 GB RAM.

A. EXPERIMENTS ON SIMULATED DATA

The original bathymetric data were collected from a Kongsberg GeoSwath Plus multibeam bathymetry system installed in a vessel at a sampling frequency of 1 Hz, as shown in Fig. 9. The actual horizontal position of the vehicle was provided by a differential GPS 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

(a) CMBS200 multibeam sonar

(b) Monitoring system

FIGURE 14. Experimental equipment to collect field data.

was around 8 km in 3613 s. Each simulated swath contained 141 bathymetric measurement points.

As the navigation results provided by DR were not collected in experiments, they were simulated by adding Gaussian noise to the GPS data as follows:

$$x_{DR}^{t} = x_{DR}^{t-1} + x_{GPS}^{t} - x_{GPS}^{t-1} + \mathcal{N}(m_e, \sigma_e^2), \tag{28}$$

where x_{GPS}^{t} is the vehicle position provided by GPS at time t, m_e and σ_e are the mean, and the standard deviation of the Gaussian noise, respectively. We set $m_e = 0.012$m and $\sigma_e = 0.01$m.

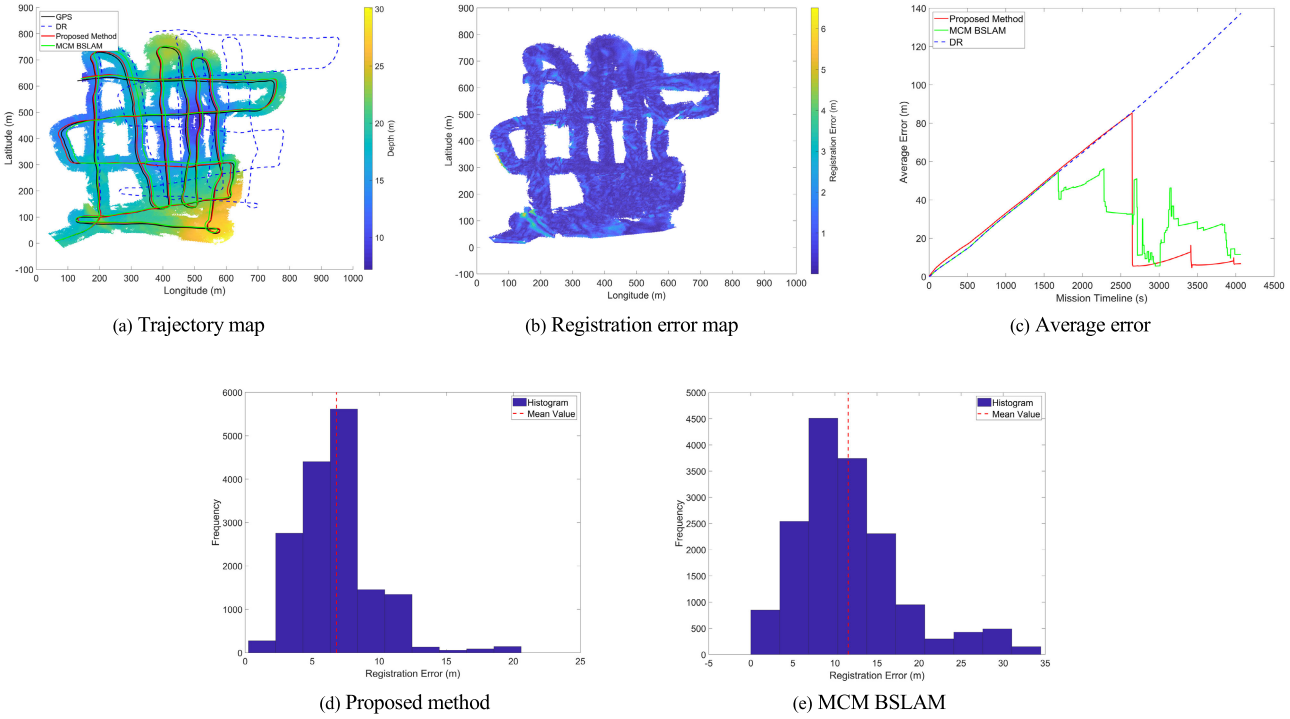
1) NAVIGATION AND MAPPING COMPARISON

The parameters of the proposed method were set as: $N = 400$, $R = 10$ m, $\sigma_C^2 = 1$ m^2, $A = 1$ m^2, MSE threshold was 0.5m^2, $T_{\mathrm{p}} = 20$, $T_{\mathrm{m}} = 20$. The parameters of BPSLAM were set as: $N = 400$, $R = 10$ m, $\sigma_C^2 = 1$ m^2, $T_{\mathrm{p}} = 20$, and $T_{\mathrm{m}} = 20$.

As shown in Fig. 10, the errors of the proposed method, BPSLAM, and MCM BSLAM being 3.63, 8.55, and 3.80 m at the end of the mission, and computation times being 300, 600, and 700 s, respectively. The main reason for the different accuracy of the two PF BSLAM methods is that σ_C^2 is set as a large value to ensure the particles are widely distributed in the state space. However, in BPSLAM, the larger the value of σ_C^2, the higher the inter-frame uncertainty given in the propagation of particles' states. Thus, the particles cannot be accurately weighted using unreliable trajectories.

As shown in Fig. 11, the standard deviation of the particles in the proposed method increased fast after each resampling to recover the particle diversity in the state space. In contrast, the standard deviation of the particles in BPSLAM decreased slowly after each resampling, the mean of the trajectories stored in particles could not effectively represent the estimated trajectory of the vehicle.

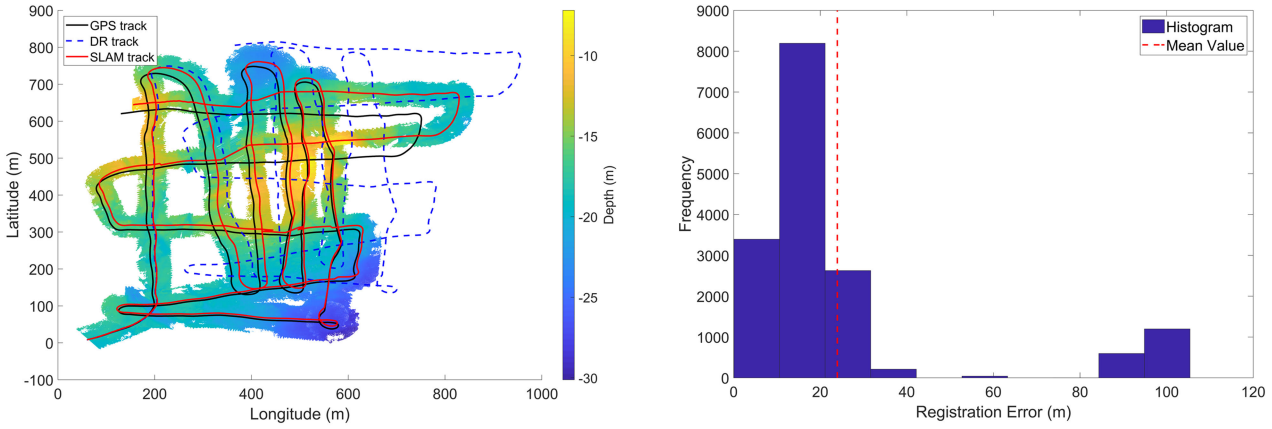
The main reason that the computation time of the BPSLAM is longer than that of the proposed BSLAM is the particles of BPSLAM do not converge during the mission, hence the particles are widely distributed in the state space. When the estimated state of the vehicle is away from the mean trajectory, BPSLAM needs extra computing resources to weigh the particles that close to the mean trajectory. In contrast, as the particles in the proposed method converge during the mission, most particles are out of the loop closure detection range when the estimated state of the vehicle is away from the mean trajectory, hence the time to weigh the particles that close to the mean trajectory is reduced.



(a) Trajectory map (b) Registration error map (c) Average error

(d) Proposed method (e) MCM BSLAM

FIGURE 15. SLAM results for 400 particles. The map in (a) was generated using the proposed BSLAM. The average error in (c) was calculated with respect to GPS data. The error distribution of the trajectory points shown in (c)–(e) have the mean values indicated by the red dashed lines.



(a) Trajectory map (b) Error distribution of the trajectory

FIGURE 16. Mapping results without LCCD.

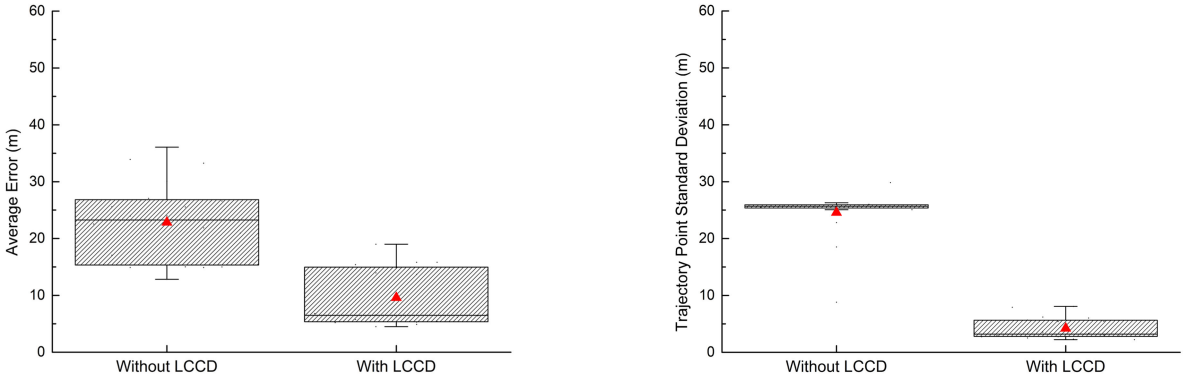
2) EFFECT OF NUMBER OF PARTICLES ON SLAM

We considered the number of particles $N = 50$, 100, 200, 300, 400, 500, and 600 and executed 20 Monte Carlo tests for every case. As shown in Fig. 12, as the number of particles increased, the probability of particles covering the correct state increased, and the accuracy of the proposed method was improved. When $N = 400$, the corresponding values were 5.97 m and 3.10 m. When the number of particles was larger than 400, since the number of particles was sufficient to cover the state space, the accuracy of the results was not increased by increasing the number of particles. In contrast, BPSLAM failed to provide stable and accurate results when σ_C^2 was a large value, which supports the analysis in Section V.

As shown in Fig. 13, when $N = 400$, the proposed method had a computation time of 300 s, being 50% faster than BPSLAM (600 s) and 57% faster than MCM BSLAM (700 s). The proposed BSLAM is more efficient than BPSLAM.

B. EXPERIMENTS ON FIELD DATA

We conducted experiments with field data to verify that the proposed BSLAM with LCCD can provide accurate navigational results for vehicles operating at sea.



(a) Average error

(b) Standard deviation of the trajectory

FIGURE 17. Effect of LCCD on proposed BSLAM. Each black point represents the result of one test, and the red triangles indicate the mean of each group.

The test vessel was a 16 m-long and 4 m-wide boat traveling at a speed of 4 kn. As shown in Fig. 14, the T-SEA CMBS200 200 kHz multibeam sonar emits 173 beams with a frequency of 4 Hz. The StarNeto XW-GI5651 inertial navigation system provides data with an angular accuracy of 0.1°. The NovAtel ProPak-LB differential GPS provides the position information with 20 cm accuracy.

As the accuracy of the BPSLAM degrades seriously under a large accumulated error, only the MCM BSLAM was used for comparisons. The DR trajectory was simulated by setting $m_e = 0.012$m and $\sigma_e = 0.01$m in (28). The parameters of the proposed method were set as: $N = 400$, $R = 10$ m, $\sigma_C^2 = 1.25$ m^2, $A = 3.12$ m^2, MSE threshold was 0.4 m^2, $T_\mathrm{p} = 50$, and $T_\mathrm{m} = 100$. As shown in Fig. 15, the mean registration error of the bathymetric map constructed by the proposed method was 0.59 m, with 98.46% of the measurement errors remaining below 2 m. The average error of the proposed BSLAM and MCM BSLAM at the end of the mission were 6.81 and 11.58 m, and their run times were 2610 and 3760 s, respectively. All the trajectory errors in the proposed BSLAM remained below 25 m.

We conducted 20 Monte Carlo tests on the proposed BSLAM, obtaining an average error of 9.59 m and a computation time of 2630 s. As MCM BSLAM took plenty of computational resources to identify invalid loop closures, the proposed BSLAM was 30% faster than MCM BSLAM. We also conducted 20 tests to demonstrate the performance of LCCD. As shown in Fig. 16 and Fig. 17, without LCCD, part of the trajectory was split due to the influence of invalid loop closures, and the errors of 1838 trajectory points were above 50 m. With LCCD, invalid loop closures were removed, the average error decreased from 22.84 to 9.59m, and the average standard deviation of the trajectory points decreased from 24.56 to 4.22 m.

The precision, recall, and F1 measure were used to evaluate the LCCD performance, detailed explanations of these measures can be found in [28]. The Single-Cluster Spectral Graph Partitioning (SCGP) proposed by [29] is a spectral clustering-based algorithm that can remove invalid loop closures and it was used for comparisons. As presented in Table 2, hierarchical clustering had a better performance compared with spectral clustering when the number of input loop closures is small.

TABLE 2. Average performance measures.

Method	$FP/(TP+FP)$	$FN/(TN+FN)$	*Precision*	*Recall*	F_1
LCCD	0.35/5.70	0.00/2.25	0.96	1.00	0.98
SCGP	0.45/5.10	0.80/3.10	0.93	0.89	0.90

VI. CONCLUSION

We proposed a bathymetric PF SLAM method to reduce the computational consumption and identify invalid loop closures. The following conclusions can be drawn:

1) The mean trajectory map reduces the algorithm complexity and computational consumption.
2) LCCD which is based on hierarchical clustering can remove the invalid loop closures and improve the global map consistency.
3) The playback experiments demonstrate that accurate mapping and navigation results can be obtained in different environments using the proposed BSLAM method.

The proposed method will fail if all the distances between each actual vehicle position above the radius R. The amount of the loop closures detected by the proposed method can be increased by increase the value of R, and a suitable trajectory may need predefined before the mission. The implementation of the proposed method on an AUV by using the DR data collected from experiments will also be our future works.

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] W. J. Kirkwood, "Development of the DORADO mapping vehicle for multibeam, subbottom, and sidescan science missions," *J. Field Robot.*, vol. 24, no. 6, pp. 487–495, Jun. 2007.

[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] J. J. Leonard and A. Bahr, "Autonomous underwater vehicle navigation," in *Springer Handbook of Ocean Engineering*. Cham, Switzerland: Springer, 2016, pp. 341–358.
[5] P. Chen, Y. Li, Y. Su, X. Chen, and Y. Jiang, "Review of AUV underwater terrain matching navigation," *J. Navigat.*, vol. 68, no. 6, pp. 1155–1172, Nov. 2015.
[6] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[8] 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.
[9] 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.
[10] 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. Autom. (ICRA)*, Orlando, FL, USA, May 2006, pp. 3575–3580.
[11] P. Norgren and R. Skjetne, "A multibeam-based SLAM algorithm for iceberg mapping using AUVs," *IEEE Access*, vol. 6, pp. 26318–26337, 2018.
[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] R. A. Stuckey, "Navigational error reduction of underwater vehicles with selective bathymetric SLAM," *IFAC Proc. Volumes*, vol. 45, no. 5, pp. 118–125, 2012.
[15] 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.
[16] N. Bore, I. Torroba, and J. Folkesson, "Sparse Gaussian process SLAM, storage and filtering for AUV multibeam bathymetry," in *Proc. IEEE/OES Auto. Underwater Vehicle Workshop (AUV)*, Nov. 2018, pp. 1–6.
[17] S. Thrun, "Particle filters in robotics," in *Proc. Uncertain. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 511–518.
[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[19] J. Cui, D. Feng, Y. Li, and Q. Tian, "Research on simultaneous localization and mapping for AUV by an improved method: Variance reduction FastSLAM with simulated annealing," *Defence Technol.*, vol. 16, no. 3, pp. 651–661, Jun. 2020.
[20] B. Claus and R. Bachmayer, "Terrain-aided navigation for an underwater glider," *J. Field Robot.*, vol. 32, no. 7, pp. 935–951, Oct. 2015.
[21] G. Salavasidis, C. Harris, S. McPhail, A. B. Phillips, and E. Rogers, "Terrain aided navigation for long range AUV operations at arctic latitudes," in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Nov. 2016, pp. 115–123.
[22] 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, 2007, pp. 200–205.
[23] O. Köksoy, "Multiresponse robust design: Mean square error (MSE) criterion," *Appl. Math. Comput.*, vol. 175, no. 2, pp. 1716–1729, Apr. 2006.
[24] S. Dektor and S. Rock, "Improving robustness of terrain-relative navigation for AUVs in regions with flat terrain," in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Southampton, U.K., Sep. 2012, pp. 1–7.
[25] D. Peng, T. Zhou, H. Li, and W. Zhang, "Terrain aided navigation for underwater vehicles using maximum likelihood method," in *Proc. IEEE/OES China Ocean Acoust. (COA)*, Harbin, China, Jan. 2016, pp. 1–6.
[26] 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.
[27] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview, II," *WIREs Data Mining Knowl. Discovery*, vol. 7, no. 6, p. e1219, Nov. 2017.
[28] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
[29] E. Olson, M. R. Walter, S. J. Teller, and J. J. Leonard, "Single-cluster spectral graph partitioning for robotics applications," in *Proc. Robot., Sci. Syst. (RSS)*, Cambridge, MA, USA, Jun. 2005, pp. 265–272.

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 in engineering from Harbin Engineering University, in 2001, with a focus on shipping and marine engineering, the master's degree from the Science and Technology on Underwater Vehicles Laboratory (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 Science and Technology on Underwater Vehicles Laboratory (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 Technology University, Qingdao, China, in 2015. She is currently pursuing the Ph.D. degree in marine engineering with the Science and Technology on Underwater Vehicles Laboratory (STUVL), Harbin Engineering University. Her general interests include seafloor terrain matching navigation and underwater SLAM.