



Underwater Localization and Mapping Based on Multi-Beam Forward Looking Sonar

Chensheng Cheng, Can Wang, Dianyu Yang, Weidong Liu and Feihu Zhang*

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China

SLAM (Simultaneous Localization And Mapping) plays a vital role in navigation tasks of AUV (Autonomous Underwater Vehicle). However, due to a vast amount of image sonar data and some acoustic equipment's inherent high latency, it is a considerable challenge to implement real-time underwater SLAM on a small AUV. This paper presents a filter based methodology for SLAM algorithms in underwater environments. First, a multi-beam forward looking sonar (MFLS) is utilized to extract environmental features. The acquired sonar image is then converted to sparse point cloud format through threshold segmentation and distance-constrained filtering to solve the calculation explosion issue caused by a large amount of original data. Second, based on the proposed method, the DVL, IMU, and sonar data are fused, the Rao-Blackwellized particle filter (RBPF)-based SLAM method is used to estimate AUV pose and generate an occupancy grid map. To verify the proposed algorithm, the underwater vehicle is equipped as an experimental platform to conduct field tasks in both the experimental pool and wild lake, respectively. Experiments illustrate that the proposed approach achieves better performance in both state estimation and suppressing divergence.

OPEN ACCESS

Edited by:

Chen Qiao,
Xi'an Jiaotong University, China

Reviewed by:

Zhe Zhang,
Taiyuan University of Technology,
China

Tian Wang,
Beihang University, China

*Correspondence:

Feihu Zhang
feihu.zhang@nwpu.edu.cn

Received: 26 October 2021

Accepted: 15 December 2021

Published: 07 January 2022

Citation:

Cheng C, Wang C, Yang D, Liu W and Zhang F (2022) Underwater Localization and Mapping Based on Multi-Beam Forward Looking Sonar. *Front. Neurobot.* 15:801956. doi: 10.3389/fnbot.2021.801956

Keywords: SLAM, multi-beam forward looking sonar, point cloud, grid map, underwater vehicle

1. INTRODUCTION

AUVs (Autonomous Underwater Vehicles) have been widely applied to perform various complex underwater tasks such as resource exploration (Ohta et al., 2016), environmental monitoring (Williams et al., 2012; Barrera et al., 2018), underwater rescue (Venkatesan, 2016), and military operations (Hagen et al., 2005), etc. To satisfy the safety and reliability, AUVs should acquire accurate localization in underwater unknown environments.

To achieve this goal, Doppler Velocity Logging (DVL) and Inertial Measurement Unit (IMU) are fused with acoustic long baseline (Matos et al., 1999), short baseline (Vickery, 1998), and ultra-short baseline (Hao et al., 2020) to calculate the position of AUVs. However, these traditional methods have shortcomings regarding to error divergence. DVL measures the speed by integrating the acceleration and meanwhile further calculates localization from dead-reckoning, the final result may, therefore, contain cumulative errors; The method based on the acoustic baseline needs to arrange the equipment in the environment in advance; Therefore, it is essential to use a more robust and reliable method to solve above problems. On the other hand, SLAM enables AUVs to fuse sensor data and build a map of an unknown environment, while localizing simultaneously. So far, sensors applied in underwater slam include cameras, side-scan sonar (SSS), single-beam mechanical scanning sonar (SMSS), and multi-beam forward-looking sonar (MFLS).

Camera-based underwater SLAM estimates the ego-motion by extracting and matching features from adjacent images and optimizing the pose at the back-end (Kim and Eustice, 2015; Hong et al., 2016). Jongdae Jung et al. proposed a vision-based SLAM, where artificial underwater landmarks help visualize camera poses (Jung et al., 2017). Suresh et al. proposed a novel method for underwater localization using natural landmarks (Suresh et al., 2019). Sparse features were obtained via an onboard upward-facing stereo camera through water for underwater localization. Although the cost of the camera is low, vision-based underwater SLAM has significant limitations. The camera's detection range is close and can only work in a clean environment with good light.

Compare to cameras, sonar emits sound waves in single or multiple directions and obtains information about the surrounding environment by analyzing each echo's strength and return time. Sonar-based method is, therefore, the development trend of underwater SLAM (Wang et al., 2017; Wang and Cheng, 2020). Chen et al. proposed an RBPF SLAM algorithm to tackle the issues of scan distortion and data sparseness caused by the slow-sampling mechanical scanning sonar, by carefully designed a sliding window-based scan module (Chen et al., 2020). The formed scans are then fed into the modified RBPF to build a consistent grid-based map. Siantidis et al. described a SLAM system with a dead reckoning system and a side-scan sonar (Siantidis, 2016), which can compensate for the position drifts. Aulinas et al. proposed a feature-based sub-mapping SLAM approach, which considered side-scan salient objects as landmarks (Aulinas et al., 2010). However, the long scanning period is quite challenging to meet underwater real-time performance, as the return of the side-scan sonar and mechanical scanning sonar image is delayed.

Meanwhile, MFLS is becoming more and more popular in underwater perception because of its solid real-time performance, small size, and easy installation (Hurtós et al., 2014; Wright and Baldauf, 2016). Wang et al. proposed a novel approach for underwater SLAM using an MFLS for 3D terrain mapping tasks (Wang et al., 2019). Instead of repeatedly projecting extracted features into Euclidean space, they applied optical flow within bearing-range images for tracking extracted features and assumed these features are sampled from a Gaussian Process terrain map. Neves et al. introduced a novel multi-object detection system, which outputs object position and rotation from MFLS images (Neves et al., 2020). Pyo et al. proposed a novel localization method in shallow water, where localization is based on passive-type acoustic landmarks. Through modeling, the distance from landmark to MFLS could be calculated (Pyo et al., 2017). However, a complete occupancy grid map using underwater vehicles with MFLS is still missing.

This paper presents a methodology for the SLAM algorithm based on MFLS, by building an accurate occupancy grid map and providing an accurate estimation of AUV poses. The occupancy grid graph can be used for subsequent global positioning and path planning. The main contributions of the proposed algorithm are in two aspects. (1) Aiming at the slow processing speed caused by a large amount of MFLS image data, a method is

proposed to convert the collected sonar image into sparse point cloud format data through threshold segmentation and distance-constrained filtering. (2) Based on the proposed method, the DVL, IMU, and MFLS data are fused, and then the RBPF-based SLAM method is used to generate an accurate occupancy grid map, and at the same time, the drift of the inertial navigation can be suppressed.

The structure of the proposed approach is as follows. Section 2 introduces the characteristics of the MFLS used in this article. The proposed SLAM method for underwater vehicles is detailed in section 3. The experimental results are shown in section 4. Section 5 presents a brief conclusion and section 6 is our future work.

2. PROBLEM DESCRIPTION OF MFLS SLAM

MFLS is an image sonar. It can emit multiple sound waves with a vertical width in the horizontal direction and detect the environment based on the echoes. The working principle is shown in **Figure 1**. However, it has no resolution in the vertical direction, so the result is a two-dimensional image. By measuring the flight time and intensity of the echo, images with different degrees of brightness will be obtained, as shown in **Figure 2**. The bright part indicates an obstacle with high echo intensity, and the dark part indicates that the echo intensity at that location is weak.

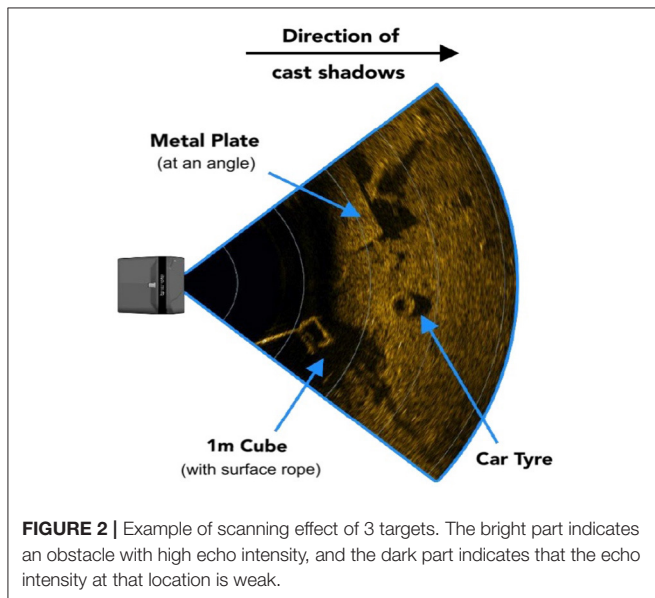
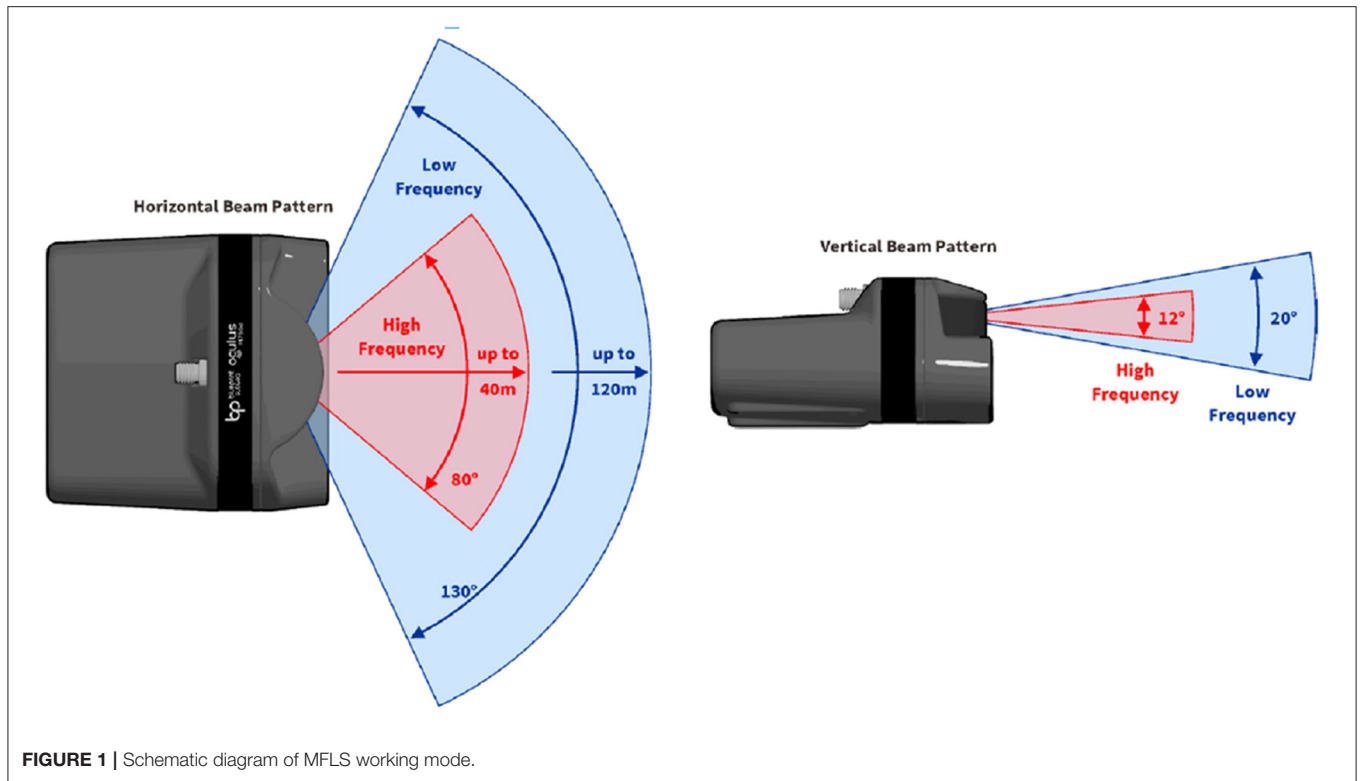
Generally, there are two MFLS data processing methods: image-level processing and echo intensity processing. However, due to the dense beams, high resolution, and the relatively large amount of image sonar data, processing directly from the image level will cause a large amount of calculation in the SLAM process, and it is difficult for the processor installed on the small AUV to process the data in real-time. In order to solve this problem, this paper converts the sonar image data into corresponding point cloud data and then uses distance-constrained filtering to extract the necessary information and reduce the amount of calculation. In the case of limited processor performance, the goal of real-time SLAM is achieved on a small AUV.

3. PROPOSED SLAM METHOD

Figure 3 is the overall framework of the SLAM algorithm proposed in this article. We first fuse DVL data and IMU data to obtain odometer data in the algorithm and use it for dead reckoning. At the same time, the MFLS sonar data is preprocessed. First, the sonar data is converted into point cloud data through threshold segmentation and data conversion. The obtained point cloud data is then subjected to sparse processing using distance constraints filtering. Finally, send the processed data into the RBPF-SLAM algorithm for positioning and composition.

3.1. Dead Reckoning

The function of the dead reckoning module is to apply the IMU and DVL to provide a rough estimate of the AUV pose. When the



sonar samples the data, the dead reckoning module applies the extended Kalman filter (EKF) to estimate the attitude by fusing the data from these sensors.

Suppose the AUV estimated state is $X_k = [p_k^T, \varphi_k^T]^T$ in the global coordinate system at time k , where p_k represents the position of the AUV, and φ_k represents the attitude of the AUV.

p_k and φ_k are, respectively, defined as

$$p_k = [x_k \ y_k \ z_k]^T, \quad \varphi_k = [\phi_k \ \theta_k \ \psi_k]^T$$

where x_k, y_k, z_k are the position coordinates in each axis in the global coordinate frame, and ϕ_k, θ_k, ψ_k are the Euler angles roll, pitch, and yaw in each corresponding axis.

Assuming that the linear velocity and angular velocity of the AUV are v_k and ω_k , they are jointly used as the control input $u_k = [v_k^T, \omega_k^T]^T$. Specifically, v_k and ω_k are expressed as

$$v_k = [l_k \ m_k \ n_k]^T, \quad \omega_k = [o_k \ q_k \ r_k]^T$$

where each element in the two vectors is the linear velocity and angular velocity on each axis in the AUV coordinate system. Then the kinematics model of AUV can be expressed as

$$X_{k+1} = f(X_k, u_k) = X_k + \Delta_T * J(X_k) * u_k \quad (1)$$

where $J(X_k)$ is the transformation matrix, and Δ_T is the sampling time interval. u_k can be represented by DVL measurement value and IMU measurement value with Gaussian noise $\omega_k \sim N(0, Q_u)$. Due to this kind of noise, there will be error accumulation in dead reckoning. Therefore, other sensor information is called to correct the error during the update phase of EKF. Through formula (1), the AUV state can be estimated as

$$\hat{X}_{k+1|k} = f(\hat{X}_{k|k}, \hat{u}_k) \quad (2)$$

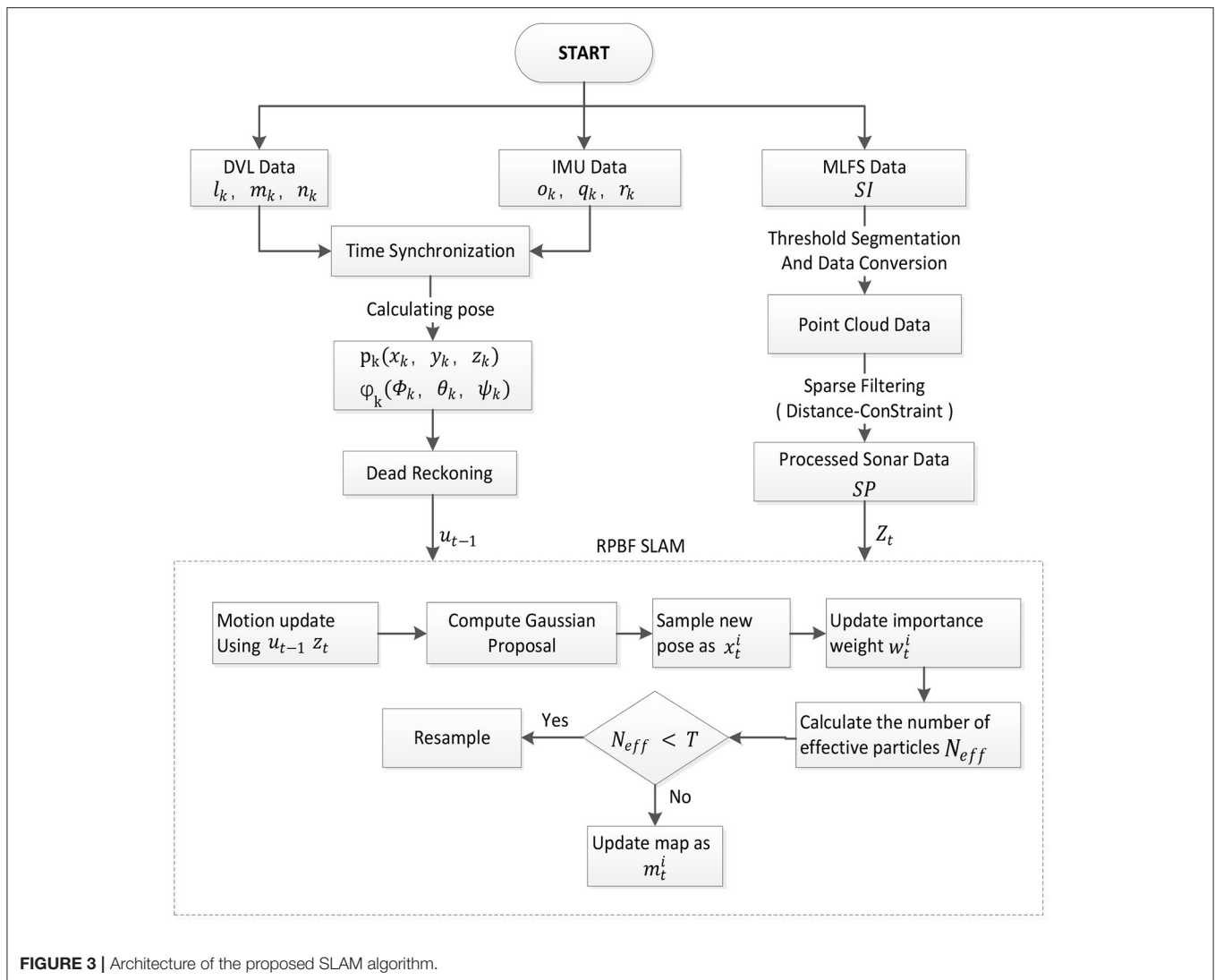


FIGURE 3 | Architecture of the proposed SLAM algorithm.

The covariance matrix used for the prediction error can be expressed as

$$P_{k+1|k} = F_{k+1} * P_{k|k} * F_{k+1}^T + G_{k+1} * Q_u * G_{k+1}^T \quad (3)$$

where F_{k+1} and G_{k+1} are the Jacobian matrices obtained by solving the partial derivative of the nonlinear model function f about the state X_k and the noise ω_k .

Finally, the model prediction is updated by applying the standard EKF update equation to generate the estimated pose of the AUV.

3.2. Threshold Segmentation and Data Conversion

The working principle of sonar is to generate an echo according to the sound wave encountering an object and then generate an image according to the time and intensity of the return of the echo. Due to water quality and acoustic interference, multi-beam sonar data will carry a lot of clutter and outliers in a natural

environment. Direct conversion into lidar data for mapping will distort the resulting map. Therefore, it is necessary to filter according to the environment so that the data can better reflect the characteristics of the environment.

In this experiment, the raw sonar data were processed in three steps: threshold segmentation, data conversion, and distance-constrained filtering. The flow chart of the proposed algorithm is presented in **Algorithm 1**.

SI is sonar's original data, an image generated with parameters with a scanning angle of 130° and a scanning distance of $40m$. **Figure 4** is an image generated by aiming at the corner of the pool with sonar, which contains a lot of clutter. $T1$ is the threshold for filtering selection. In general, we use the average pixel value as the threshold for filtering. At the same time, the threshold can also be manually set according to the water quality environment. Under normal circumstances, we put several targets in the water or look for an environment with apparent characteristics in the background. Then, we use sonar to scan in real-time and continuously adjust the threshold manually until the generated

Algorithm 1: Sonar data process.

```

Data: [SI] Sonar's raw image data and the average value of
pixels.
Result: [SP] Processed sonar data.
1 begin
2   SI = ReadSonarData();
3   //Threshold segmentation and data conversion
4   for  $Pixel_{(i*j)} \in SI$  do
5     if  $Pixel_{(i*j)} < T_1$  then
6       |  $Pixel_{(i*j)} = 0$ ;
7     else
8       |  $Pixel_{(i*j)} = Pixel_{(i*j)}$ ;
9     end
10  end
11  for  $k = 0; k < i; k++$  do
12    for  $Pixel_{(k*j_{th})} \in Pixel_{(k*j)}$  do
13      |  $Pixel_{(k*j_{th})} = FindMaxValue(Pixel_{(k*j)})$ ;
14    end
15     $range_i = j_{th} * range\_step$ ;
16     $angle_i = i * angle\_step$ ;
17    PointC.PushBack( $range_i, angle_i$ );
18  end
19  //Distance-constrained filtering
20   $num = 0, sum = 0$ ;
21  for  $p = 0; p < i; p++$  do
22     $num = num + 1$ ;
23     $sum = sum + range_p$ ;
24    if  $p \% T_2 == 0 \ \&\& \ p! = 0$  then
25      |  $average\_value = sum / num$ ;
26      | for  $n = p - num; n < num; n++$  do
27        |  $dif = range_p - average\_value$ 
28        | if  $dif > T_3$  then
29          |  $range_p = 0$ 
30        | else
31          |  $range_p = range_p$ 
32        | end
33      | end
34      |  $num = 0, sum = 0$ ;
35    else
36    end
37  end
38  SP.PushBack( $range_p$ )
39 end

```

point cloud data can better reflect the target profile. At the same time, when watching open waters, less noise is generated, and the current threshold is selected as the optimal threshold in the current environment.

The pixel value below T_1 is assigned a value of 0, and the pixel value above T_1 remains unchanged. Since only the features of the surface of the object are considered when constructing the map, we believe that the brightest point on the beam is formed by the sound wave hitting the surface of the object and

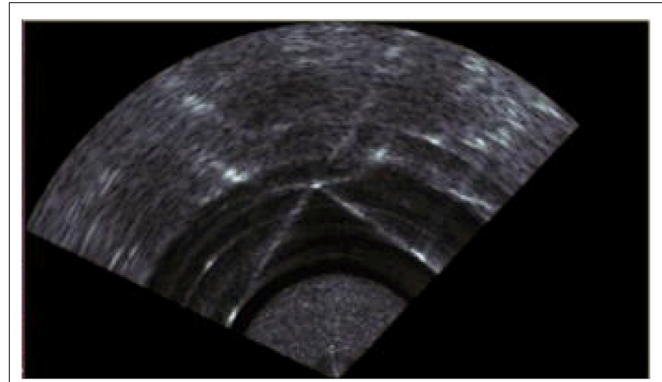


FIGURE 4 | Sonar raw data. This is the raw data picture of the sonar facing the corner of the experimental pool.

returning. Therefore, we only select a point with the highest intensity on each beam as the target of interest. After preliminary filtering, we calculate the position of the target pixel in the sonar coordinate system according to the angular resolution $angle_step$ and distance resolution $range_step$ of the sonar. The resolution value can be changed by setting the sonar parameters. Calculating all the pixels of the sonar image, we can get the point cloud data, as shown in **Figure 5A**.

3.3. Distance-Constrained Filtering

From **Figure 5A**, we can find that many bright spots are generated behind the wall of the pool. This is not the data we want, and it will affect the positioning accuracy and mapping effect of the AUV. Aiming at these clutter interference, this paper uses a distance constraint-based method to filter out clutter while reducing the amount of data. Doing so can improve the real-time performance of the algorithm while ensuring positioning accuracy and mapping quality. The flow chart of this algorithm is shown in the lower part of **Algorithm 1**.

Process the obtained point cloud data *PointC*:

- Set a beam threshold T_2 , and divide all range data $range_i$ into $A = \{[range_{(0)} - range_{(T_2-1)}], [range_{(T_2)} - range_{(2*T_2-1)}], \dots, [range_{(i-T_2-1)} - range_{(i)}]\}$ according to the angle order ($-65^\circ - 65^\circ$).
- Set the distance threshold T_3 , calculate the average of all $range_i$ in A_k , and then calculate the difference dif between each $range_i$ and the average. If dif is greater than T_3 , the data is judged to be noise deleted, if it is less than T_3 , it is judged to be valid data.
- Save the valid data into the *SP*, and the sonar data processing ends.

Figure 5B is the point cloud image obtained after distance constraint filtering. It can be found that the clutter behind the wall of the pool is successfully filtered, and data that can truly reflect the environmental characteristics are obtained.

3.4. RBPF SLAM With MFLS

The flow chart of the proposed algorithm is presented in **Algorithm 2**. *PS* is the particle set generated according to the initial state of the AUV. According to the theory that the joint

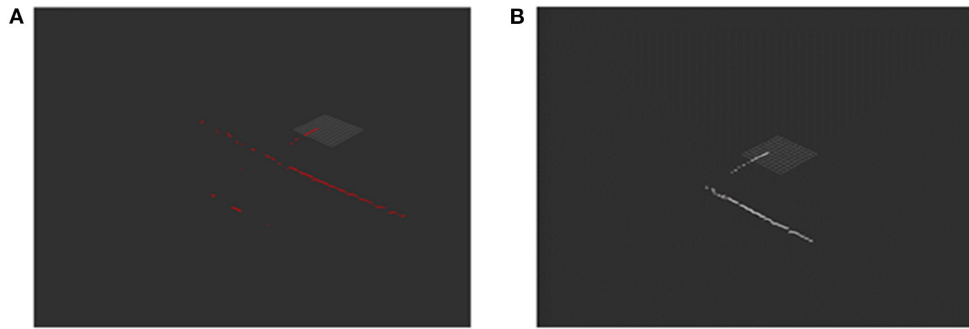


FIGURE 5 | Panel (A) shows the point cloud data of the sonar raw data after threshold segmentation and data conversion. Panel (B) is the result of filtering based on distance-constrained.

Algorithm 2: RBPF with multi-beam forward looking sonar.

Data: $[X_{t-1}, SP]$ The AUV's initial pose and processed sonar data.

Result: $[X_t]$ The pose of the most weighted particles.

```

1 begin
2    $PS_{t-1} = \text{Init}();$ 
3   for  $SP = \text{ProcessSonarData}()$  do
4      $X_k = \text{DeadReckoning}();$ 
5      $z_t = SP;$ 
6      $u_{t-1} = X_k - X_{k-1};$ 
7     for  $PS_{t-1}^{(i)} \in PS_{t-1}$  do
8        $\langle x_{t-1}^{(i)}, \omega_{t-1}^{(i)}, m_{t-1}^{(i)} \rangle = PS_{t-1}^{(i)};$ 
9        $\tilde{x}_t^{(i)} = \text{Sample}(x_{t-1}^{(i)}, u_{t-1});$  //sampling
10       $\hat{x}_t^{(i)} = \text{IW}(m_{t-1}^{(i)}, z_t, \tilde{x}_{t-1}^{(i)});$  //scan matching and
        importance weighting
11      if  $\hat{x}_t^{(i)} = \phi$  then
12         $x_t^{(i)} = \tilde{x}_t^{(i)};$ 
13         $\omega_t^{(i)} = \omega_{t-1}^{(i)} p(z_t | m_{t-1}^{(i)}, x_t^{(i)});$ 
14      else
15         $[x_t^{(i)}, \omega_t^{(i)}] = \text{Resample}(\hat{x}_t^{(i)}, \Upsilon, u_{t-1}, m_{t-1}^{(i)});$ 
16        //resampling
17      end
18       $m_t^{(i)} = \text{MapUpdate}(m_{t-1}^{(i)}, z_t, x_t^{(i)});$  //map
        updating
19       $PS_t = PS_t \cup (x_t^{(i)}, \omega_t^{(i)}, m_t^{(i)});$ 
20    end
21     $N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\bar{\omega}^{(i)})^2};$ 
22    if  $N_{\text{eff}} < T$  then
23       $PS_t = \text{resample}(PS_t);$ 
24    else
25       $PS_{t-1} = PS_t;$ 
26    end
27 end

```

probability can be converted into the product of conditional probabilities, the solution of RBPF SLAM is to decompose the original SLAM problem into separate positioning and mapping parts (Grisetti et al., 2007).

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(m | x_{1:t}, z_{1:t}) \cdot p(x_{1:t} | z_{1:t}, u_{1:t-1}) \quad (4)$$

where $p(x_{1:t} | z_{1:t}, u_{1:t-1})$ is the posterior of potential trajectories $x_{1:t}$ given observations $z_{1:t}$ and odometry measurements $u_{1:t}$ of the AUV, $p(m | x_{1:t}, z_{1:t})$ is the posterior of maps, and $p(x_{1:t}, m | z_{1:t}, u_{1:t-1})$ is the posterior of maps and trajectories. Given the values of $x_{1:t}$ and $z_{1:t}$, $p(m | x_{1:t}, z_{1:t})$ can be calculated analytically, so the key to the problem is to calculate $p(x_{1:t} | z_{1:t}, u_{1:t-1})$.

To estimate the posterior $p(x_{1:t}, m | z_{1:t}, u_{1:t-1})$, a group of particles is first introduced. Each particle is composed of the pose x of the AUV, the grid map m , and the weight ω . The particle filter algorithm incrementally uses dead reckoning values and sonar scan data to update the particle set. This process can be divided into four steps, *sampling*, *scanmatchingandimportanceweighting*, *resampling*, and *mapupdate*. Function $\hat{x}_t^{(i)} = \text{IW}(m_{t-1}^{(i)}, z_t, \tilde{x}_{t-1}^{(i)})$ is a scan matching and importance weighting module, and its function is to calculate the pose that best matches the current map $m_{t-1}^{(i)}$ based on the current observation z_t and all samples $\tilde{x}_{t-1}^{(i)}$.

Then each particle is assigned a separate importance weighting based on the importance sampling principle $w_t^{(i)}$.

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{\eta p(z_t | m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)}, u_{t-1})}{p(x_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1})} \quad (5)$$

$$\propto w_{t-1}^{(i)} \frac{p(z_t | m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)}, u_{t-1})}{\frac{p(z_t | m_{t-1}^{(i)}, x_t) p(x_t | x_{t-1}^{(i)}, u_{t-1})}{p(z_t | m_{t-1}^{(i)}, x_{t-1}, u_{t-1})}} \quad (6)$$

$$= w_{t-1}^{(i)} \cdot p(z_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}) \quad (7)$$

TABLE 1 | The algorithm parameters.

Parameter	Value
Map-update-interval	0.3
maxRange	25
maxUrange	24
Number of particles	150
Resample threshold	0.5
Number of iterations	7

$$= w_{t-1}^{(i)} \cdot \int p(z_t|x')p(x'|x_{t-1}^{(i)}, u_{t-1})dx' \quad (8)$$

Here, $\eta = 1/p(z_t|z_{1:t-1}, u_{1:t-1})$ is the normalization factor produced by Bayes' rule where all particles are equal.

In function $Resample(\hat{x}_t^{(i)}, \Upsilon, u_{t-1}, m_{t-1}^{(i)})$, the Gaussian approximation of the proposed distribution is calculated, and new particles are sampled for the next iteration based on the calculated result. Υ is the interval threshold for resampling in the vicinity area of $\hat{x}_t^{(i)}$.

Finally, the map $m_t^{(i)}$ is updated based on the estimated pose $x_t^{(i)}$ and the observed value z_t . Then, select the map and pose of the particle with the largest weight among all the particles as the final constructed map and estimated AUV pose.

4. EXPERIMENTAL RESULT

Both simulation and practical experiments are conducted to verify the effectiveness of the proposed SLAM algorithm. **Table 1** shows the main parameters of our algorithm for experiments.

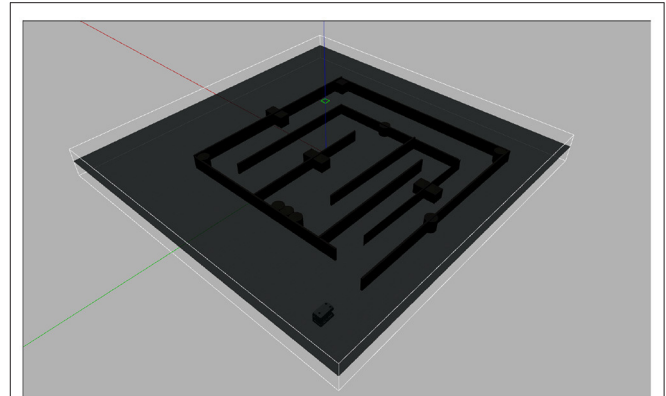
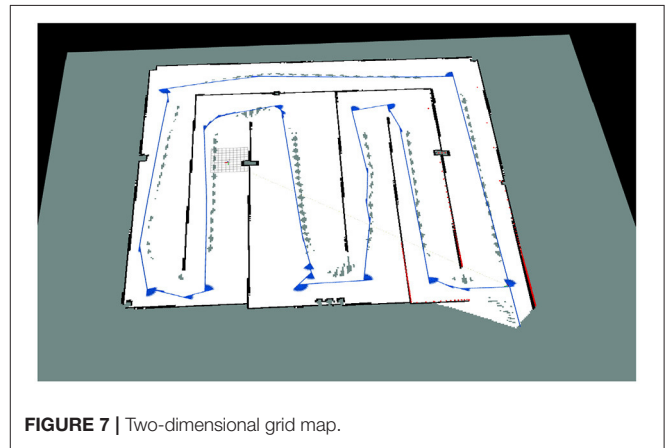
4.1. Simulation Experiments

In the simulation experiment, we used UUV-Simulator to create a Rexrov2 model, which is a full-propeller-driven ROV, and it is equipped with four cameras, four lights, and a wide range of sensors, including sonar, DVL, IMU, etc.

Figure 6 shows the Rexrov2 model and simulation environment, an underwater maze. First, build a 3-D model of the environment and load it into the 3-D simulator Gazebo. To facilitate interaction with Gazebo, we used Robot Operating System (ROS) in our simulation. The drivers of IMU, DVL, and MFLS are loaded as Gazebo plug-ins and used to publish ROS-compatible data, which are subscribed by the proposed SLAM algorithm. Our ROV is controlled to navigate the maze for one round during the simulation. The algorithm's outputs include an occupancy grid map and an estimated ROV trajectory.

Figure 7 shows the occupancy grid map generated by the proposed SLAM algorithm. Based on this map, AUV can make path planning to avoid obstacles and reach the designated position.

The comparison between the ROV position calculated by the proposed algorithm and the ground truth is shown in **Figure 8**. Their trajectory error is very small, and the simulation experiment proves the effectiveness of our proposed method.

**FIGURE 6** | Experimental environment and Rexrov2 model.**FIGURE 7** | Two-dimensional grid map.

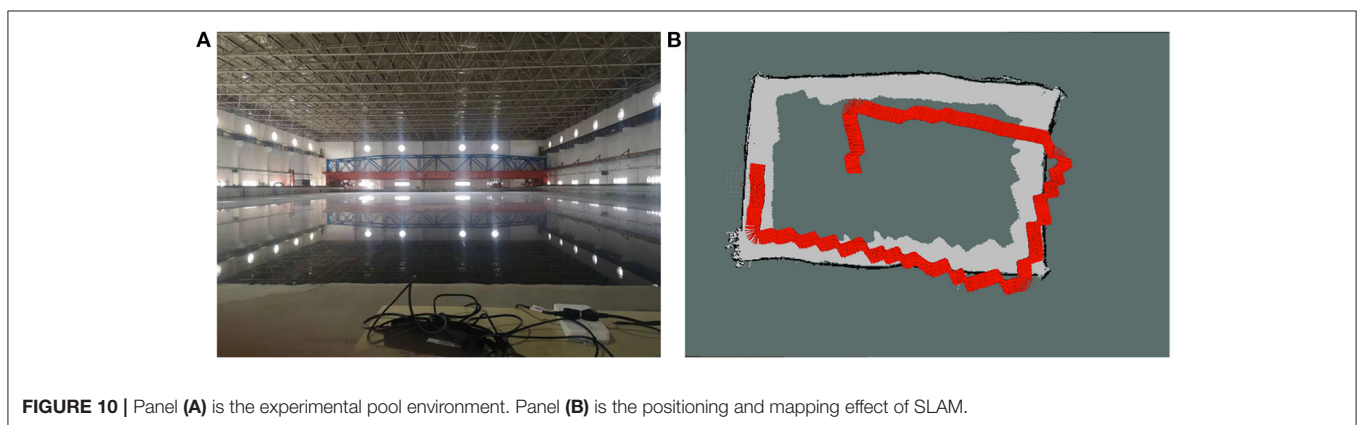
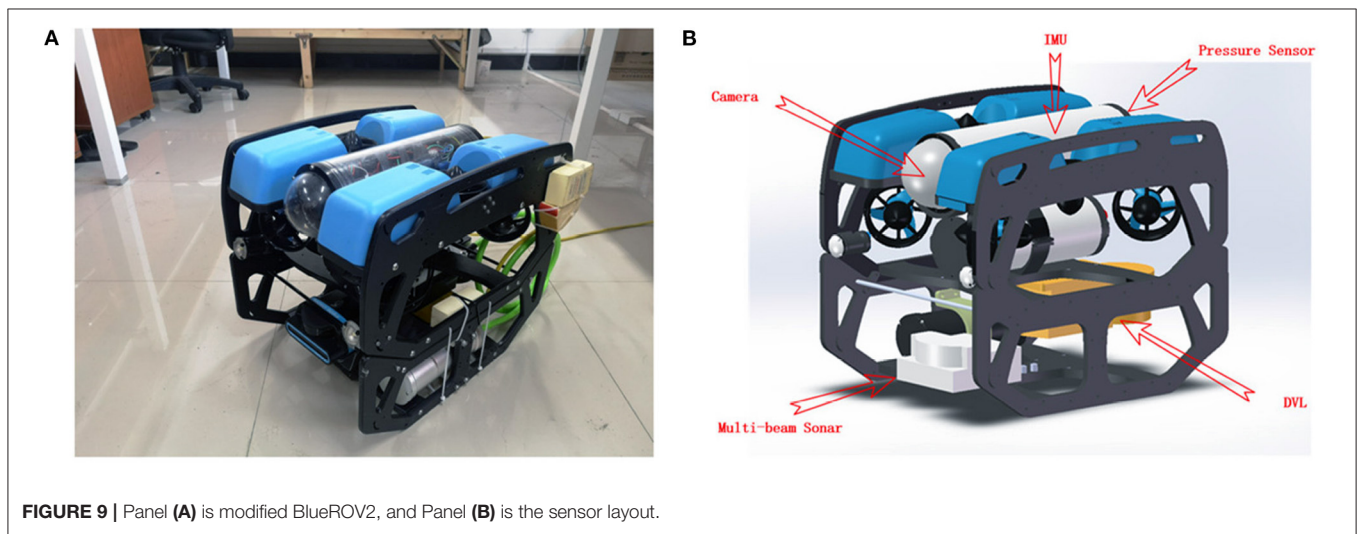
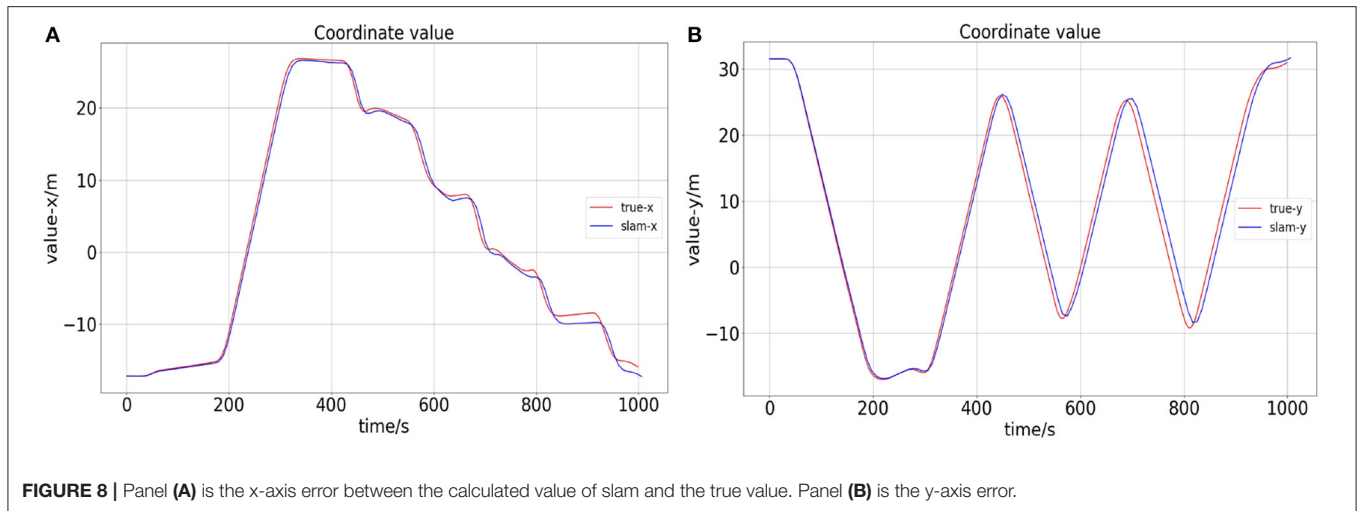
4.2. Experimental Pool Experiment

In the practical experimental, the open-source underwater robot platform BlueROV2 was used to complete the SLAM experiment of the underwater vehicle. In this paper, we only focus on the positioning and surveying indicators of the underwater vehicle, without considering the control. Therefore, the effect is the same regardless of whether the AUV or ROV is used as the experimental carrier to carry out the verification experiment. To make BlueROV2 meet our experimental requirements, we installed a multi-beam forward looking sonar and a DVL based on the original BlueROV2. Also, we installed a pair of power cat modules to achieve underwater BlueROV2 and the shore PC Long-distance data transmission. The modified BlueROV2 (as shown in **Figure 9**) can well meet the SLAM experimental requirements.

We first tested in the multi-purpose pool indoor. The size of the pool is 70 x 44 m. In the experiment, we controlled the BlueROV2 to face the wall for scanning and mapping.

The experimental pool environment is shown in **Figure 10A**, and the positioning and mapping effect of the proposed algorithm is shown in **Figure 10B**.

In **Figure 10B**, the red part represents the trajectory of the odometer, the white part represents the detectable travelable area,



the black part represents the obstacle, and the gray part represents the undetected area.

In this experiment, we used threshold segmentation and distance-constrained filtering to process the multi-beam sonar

data and then used it to build the map. It can be seen from the mapping results that good results have been achieved. At the same time, it can be seen that the odometer error is continuously accumulating. When the scan is completed, the odometer has

not completed closing, but the map established by the SLAM algorithm based on the multi-beam sonar has been closed, the odometer's position deviation is amended. Furthermore, the superiority of the SLAM algorithm based on multi-beam sonar used in this paper is proved.

4.3. Wild Lake Experiment

The field lake experiment was carried out in Liqian Lake, Xi'an, Shaanxi. During the experiment, we take a dinghy to approach the target environment, launch BlueROV2, and control it to scan the target environment for positioning and mapping

experiments. Also, there is a GPS positioning antenna on the dinghy. When the BlueROV2 scanning environment, the boat closely followed BlueROV2 to obtain GPS coordinates near it, providing actual data for the quantitative analysis of the positioning surveying experiment.

Figure 11 is a satellite image of the two experimental locations we choose.

Figures 12, 13 are the positioning and mapping results of the two experimental scenes with our proposed algorithm.

It can be seen from the experimental results (**Figures 12A, 13A**) that the mapping effect is mostly consistent with satellite



FIGURE 11 | Satellite picture of the place where the experiment was conducted.

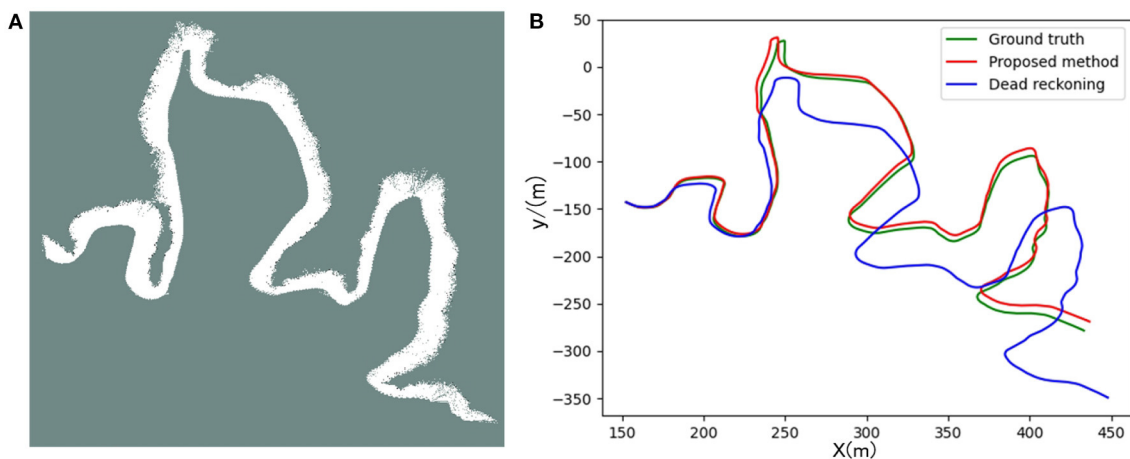
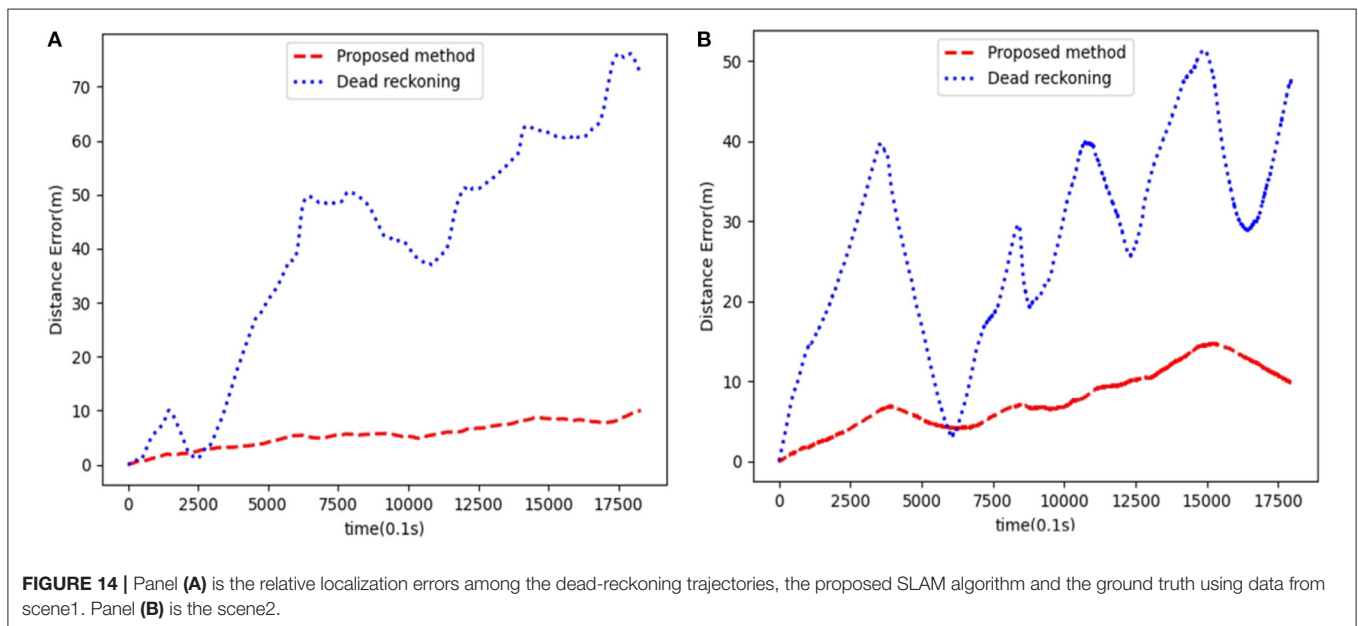
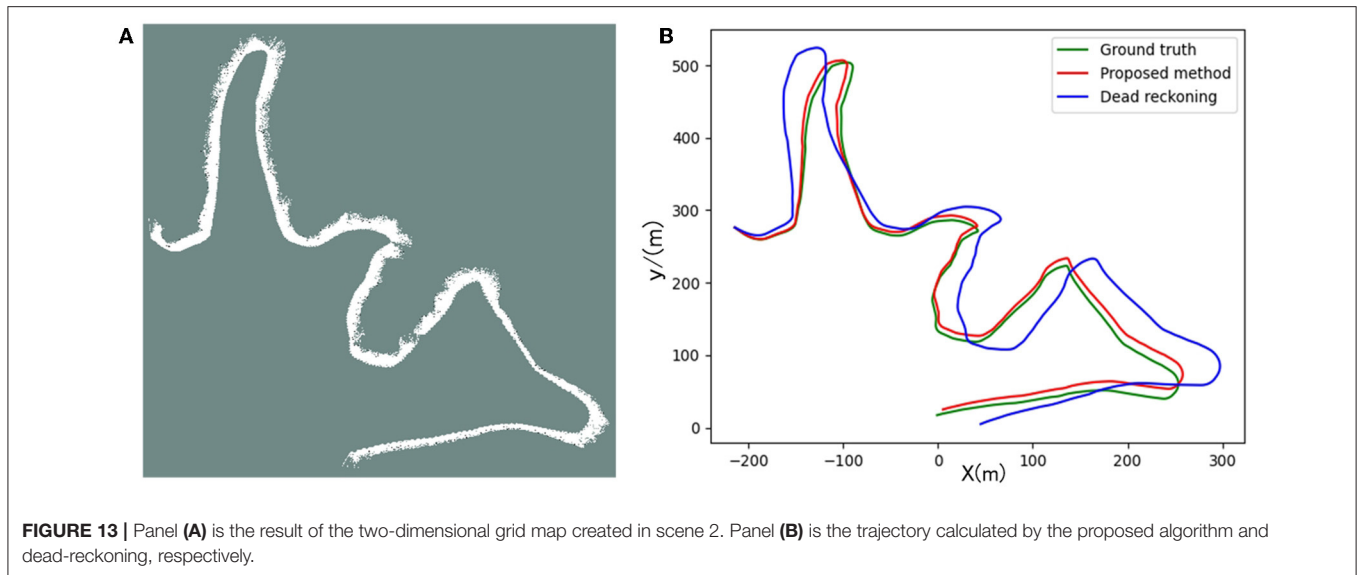


FIGURE 12 | Panel (A) is the result of the two-dimensional grid map created in scene 1. Panel (B) is the trajectory calculated by the proposed algorithm and dead-reckoning, respectively.



images. There are several reasons for the deviation between the satellite image and the map created by SLAM:

- Due to changes in water level, satellite maps may deviate from actual maps.
- When BlueRov2 dives into the water, it scans the underwater extension of the lake bank, including objects such as branches and rocks. Unfortunately, the satellite image of this part of the underwater terrain is invisible, so the constructed map will be different from the satellite image.
- BlueRov2 cannot reach some waters, so the factual environmental information has not been thoroughly scanned, which will cause differences between the constructed map and the actual environment.

To quantitatively analyze the map's location accuracy, we compare and analyze the ground truth (GPS measurement values), the estimated value of the proposed algorithm, and the dead reckoning value.

Since the GPS measured value is the latitude and longitude information, we used the geodesy package provided by ROS in the experiment and completed the conversion of latitude and longitude coordinates to two-dimensional coordinates. In this way, we unified the above three coordinate values under the same reference system to evaluate the SLAM algorithm's positioning accuracy.

The trajectory result graph is shown in **Figures 12B, 13B, 14** shows the relative localization errors. From **Figure 14** we can find that with the accumulation of time, the deviation of the

odometer will gradually increase. Using SLAM for simultaneous positioning and mapping will correct the odometer's deviation, correct the pose, and build a better map.

The maps based on the multi-beam sonar SLAM algorithm proposed in this paper can better represent the environment's characteristics. And positioning accuracy is also better than pure dead reckoning.

5. CONCLUSIONS

This paper proposed a SLAM algorithm using MFLS. Two problems are solved: Aiming at the slow processing speed caused by a large amount of MFLS image data, and a method is proposed to convert the collected sonar image into sparse point cloud format data through threshold segmentation and distance-constrained filtering; Based on the proposed method, the DVL, IMU, and MFLS data are fused, and then the RBPF-based SLAM method is used to suppress the accumulation of errors of the inertial unit and generate an accurate occupancy grid map. Finally, we used BlueROV2 as the experimental carrier, conducted field tests in the experimental pool and Liqian Lake, Xi'an, Shaanxi, and achieved good positioning and mapping results.

6. FUTURE WORK

This article mainly uses MFLS image data to be converted into sparse point cloud data format for SLAM experiments, which involves the problem of sonar filtering and data matching after filtering. There are still some problems currently, such as the inability to fully extract environmental features due to clutter

interference and the map drifting due to data matching failure. For these problems, future work will continue to study better filtering methods and data matching methods to improve the accuracy of positioning and mapping.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

CC and FZ presented the idea in this article and wrote the paper together. FZ and WL provided technical support and experimental equipment. DY and CW helped complete the experimental verification. All authors contributed to the article and approved the submitted version.

FUNDING

This study was supported by the National Natural Science Foundation of China (52171322), the National Key Research and Development Program (2020YFB1313200), and the Fundamental Research Funds for the Central Universities (D5000210944).

ACKNOWLEDGMENTS

The authors would like to thank Songxiang Wang, Xijun Zhou, and Liyuan Chen et al. for their help during the experiment.

REFERENCES

- Aulinas, J., Lladó, X., Salvi, J., and Petillot, Y. R. (2010). "Feature based slam using side-scan salient objects," in *OCEANS 2010 MTS/IEEE SEATTLE* (Seattle, WA), 1–8.
- Barrera, C., Real-Arce, D. A., Waldmann, C., Caldeira, R., Rueda, M. J., Hernández, J., et al. (2018). "Ocean-gliders contribution to improve monitoring capacity in the east-central north atlantic," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)* (Porto), 1–5.
- Chen, L., Yang, A., Hu, H., and Naeem, W. (2020). Rbpf-msis: toward rao-blackwellized particle filter slam for autonomous underwater vehicle with slow mechanical scanning imaging sonar. *IEEE Syst. J.* 14, 3301–3312. doi: 10.1109/JSYST.2019.2938599
- Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* 23, 34–46. doi: 10.1109/TRO.2006.889486
- Hagen, P. E., Storkersen, N., Marthinsen, B. E., Sten, G., and Vestgard, K. (2005). "Military operations with hugin auvs: lessons learned and the way ahead," in *Europe Oceans 2005*, vol. 2 (Brest), 810–813.
- Hao, R., Zhuo, C., Zhao, P., Nie, Q., and Li, B. (2020). "Research on optimized estimate arithmetic in ins/usbl integrated navigation system," in *2020 Chinese Automation Congress (CAC)* (Shanghai), 939–944.
- Hong, S., Kim, J., Pyo, J., and Yu, S.-C. (2016). A robust loop-closure method for visual slam in unstructured seafloor environments. *Auton. Robots* 40, 1095–1109. doi: 10.1007/s10514-015-9512-6
- Hurtós, N., Nagappa, S., Palomeras, N., and Salvi, J. (2014). "Real-time mosaicing with two-dimensional forward-looking sonar," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 601–606.
- Jung, J., Lee, Y., Kim, D., Lee, D., Myung, H., and Choi, H. (2017). "Auv slam using forward/downward looking cameras and artificial landmarks," in *2017 IEEE Underwater Technology (UT)* (Busan), 1–3.
- Kim, A., and Eustice, R. M. (2015). Active visual slam for robotic area coverage: theory and experiment. *The Int. J. Robot. Res.* 34, 457–475. doi: 10.1177/0278364914547893
- Matos, A., Cruz, N., Martins, A., and Lobo Pereira, F. (1999). "Development and implementation of a low-cost lbl navigation system for an auv," in *Oceans '99. MTS/IEEE. Riding the Crest into the 21st Century. Conference and Exhibition. Conference Proceedings (IEEE Cat. No.99CH37008)*, vol. 2 (Seattle, WA), 774–779.
- Neves, G., Ruiz, M., Fontinele, J., and Oliveira, L. (2020). Rotated object detection with forward-looking sonar in underwater applications. *Exp. Syst. Appl.* 140:112870. doi: 10.1016/j.eswa.2019.112870
- Ohta, Y., Yoshida, H., Ishibashi, S., Sugawara, M., Fan, F. H., and Tanaka, K. (2016). "Seabed resource exploration performed by auv 'yumeiruka,'" in *OCEANS 2016 MTS/IEEE Monterey* (Monterey, CA), 1–4.
- Pyo, J., Cho, H., and Yu, S. (2017). Beam slice-based recognition method for acoustic landmark with multi-beam forward looking sonar. *IEEE Sens. J.* 17, 7074–7085. doi: 10.1109/JSEN.2017.2755547
- Siantidis, K. (2016). "Side scan sonar based onboard slam system for autonomous underwater vehicles," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)* (Tokyo), 195–200.

- Suresh, S., Westman, E., and Kaess, M. (2019). Through-water stereo slam with refraction correction for auv localization. *IEEE Robot. Autom. Lett.* 4, 692–699. doi: 10.1109/LRA.2019.2891486
- Venkatesan, S. (2016). “Auv for search rescue at sea - an innovative approach,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)* (Tokyo), 1–9.
- Vickery, K. (1998). “Acoustic positioning systems. a practical overview of current systems,” in *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)* (Cambridge, MA), 5–17.
- Wang, J., Bai, S., and Englot, B. (2017). “Underwater localization and 3d mapping of submerged structures with a single-beam scanning sonar,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 4898–4905.
- Wang, J., Shan, T., and Englot, B. (2019). “Underwater terrain reconstruction from forward-looking sonar imagery,” in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC), 3471–3477.
- Wang, W., and Cheng, B. (2020). “Augmented ekf based slam system with a side scan sonar,” in *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1 (Hangzhou), 71–74.
- Williams, S. B., Pizarro, O. R., Jakuba, M. V., Johnson, C. R., Barrett, N. S., Babcock, R. C., et al. (2012). Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robot. Autom. Mag.* 19, 73–84. doi: 10.1109/MRA.2011.2181772
- Wright, R. G., and Baldauf, M. (2016). Hydrographic survey in remote regions: using vessels of opportunity equipped with 3-dimensional forward-looking sonar. *Mar. Geodesy.* 39, 439–457. doi: 10.1080/01490419.2016.1245226

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Cheng, Wang, Yang, Liu and Zhang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.