THE JOURNAL OF NAVIGATION (2019), **72**, 1602–1622.
doi:10.1017/S0373463319000286

AUV Bathymetric Simultaneous Localisation and Mapping Using Graph Method

Teng Ma, Ye Li, Yusen Gong, Rupeng Wang, Mingwei Sheng and Qiang Zhang

(*Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin 150001, China*)
(E-mail: liyeheu103@163.com)

Although topographic mapping missions and geological surveys carried out by Autonomous Underwater Vehicles (AUVs) are becoming increasingly prevalent, the lack of precise navigation in these scenarios still limits their application. This paper deals with the problems of long-term underwater navigation for AUVs and provides new mapping techniques by developing a Bathymetric Simultaneous Localisation And Mapping (BSLAM) method based on graph SLAM technology. To considerably reduce the calculation cost, the trajectory of the AUV is divided into various submaps based on Differences of Normals (DoN). Loop closures between submaps are obtained by terrain matching; meanwhile, maximum likelihood terrain estimation is also introduced to build weak data association within the submap. Assisted by one weight voting method for loop closures, the global and local trajectory corrections work together to provide an accurate navigation solution for AUVs with weak data association and inaccurate loop closures. The viability, accuracy and real-time performance of the proposed algorithm are verified with data collected onboard, including an 8 km planned track recorded at a speed of 4 knots in Qingdao, China.

KEYWORDS

1. AUVs. 2. Navigation. 3. Graph SLAM. 4. Terrain estimation. 5. Weight voting method.

Submitted: 13 July 2017. Accepted: 2 April 2019. First published online: 5 July 2019.

1. INTRODUCTION. With the increasing demand of marine exploration and the development of seabed topographic mapping methods (such as bathymetry systems), a significant amount of work has been done to produce high-resolution maps of the seabed (Doble et al., 2009; Wang et al., 2017a). However, because the map resolution for shipborne bathymetry systems is proportional to the water depths, shipborne bathymetry systems cannot not meet the requirements of high-resolution mapping in deep waters. Due to their ability to operate close to the seafloor regardless of depth, Autonomous Underwater Vehicles (AUVs) have become important mapping platforms for topography missions

and geological surveys such as searching for manganese nodules, hydrothermal vents and remains of vessels.

The accuracy of a bathymetric map depends on the accuracy of navigation of the AUV. Global Navigation Satellite Systems (GNSSs), Dead Reckoning (DR), Ultrashort-Baseline (USBL), and Long-Baseline (LBL) acoustic positioning systems all provide options for improving navigational accuracy, with various levels of attainable precision (Paull et al., 2014). GNSS observations can provide accurate locations, but this requires the AUV to surface. DR navigation error is related to the total distance travelled, which means it is unsuitable for a long-term underwater navigation scenario without other precise auxiliary means of positioning. Although USBL and LBL could provide a bounded navigation error, support vessels or acoustic arrays are necessary, and this would limit the operating range of the AUV.

As a more recently developed approach to provide accurate navigation solution for AUVs without any aids, the Simultaneous Localisation And Mapping (SLAM) technique has been introduced. The SLAM technique makes it possible for a vehicle to incrementally build a map of an unknown environment with consistency while simultaneously determining its location within this map (Durrant-Whyte and Bailey, 2006). Many breakthroughs in the development of the SLAM technique have been made in Unmanned Ground Vehicles (UGVs) (Ila et al., 2017; Dellaert and Kaess, 2006) and Unmanned Aerial Vehicles (UAVs) (Kownacki, 2016). Underwater SLAM algorithms for AUVs have also been developed, and most of them are applied using a camera (Kim and Eustice, 2013), a mechanically scanned imaging sonar (Mallios et al., 2014; Ribas et al., 2008) and Forward-Looking Sonar (FLS) (Hurtós et al., 2015; Johannsson et al., 2010). However, the use of a camera is limited to applications in which the vehicle navigates in clear water and very near to the seafloor (Ribas et al., 2008), and SLAM techniques with mechanically scanned imaging sonars cannot be applied in unstructured environments. As imaging the same scene from two different vantage points can cause significant alterations in the measurement, the forward-looking sonar SLAM can only register images that differ in small orientations, and this makes it difficult to obtain loop closures (Hurtós et al., 2015). Compared with the techniques listed above, Bathymetric SLAM (BSLAM) is more flexible and has little restriction in terms of water clarity, environment and vantage points. The major limitation of BSLAM is that the best navigation results can only be provided when the depth profile of the seabed varies significantly (Barkby et al., 2009). However, with a prior bathymetric map with a much lower resolution (such as a map resolution of 500 metres), it is possible to calculate the corresponding terrain information such as the terrain entropy and choose some areas which satisfied the entropy condition (for one area including 20×20 points, its entropy should be no less than two according to Feng (2004)) to conduct the BSLAM process as a benchmark, while the remaining areas could be mapped via topographic expansion surveys. Accordingly, building a bathymetric map and then simultaneously locating using it is more reliable than the above techniques using vision, mechanically scanned imaging sonar, or forward-looking sensors in most cases.

BSLAM has received great attention. Barkby et al. (2012) proposed a Bathymetric Particle filter SLAM (BPSLAM) to help locate the position of the AUV in real time without loop detection. This algorithm applied a Gaussian process and a Rao–Blackwellized particle filter to calculate SLAM solutions, and an ancestry tree was used to reduced computation. Regrettably, the computational efficiency of the BPSLAM still limits its real-time performance. Palomer et al. (2013) improved the Extended Kalman Filter (EKF) SLAM using

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

an octree-established point-cloud sampling model and applied an Iterative Closest Point (ICP) algorithm for terrain matching. However, this method can only correct the overall position of each submap and the location errors within the submap were neglected. Roman and Singh (2010) proposed a SLAM method including map registration and pose filtering, and experiment results have shown its significant improvements in map quality. In addition, within the loop closures detection technique in BSLAM, a terrain matching technique has also been developed over a long period and has now been applied successfully in AUVs (Ånonsen et al., 2013). For example, based on a Particle Filter (PF), Donovan (2012) proposed a Three-Dimensional (3D) terrain matching method considering tide correction and presented a novel PF-resampling technique to consistently recover a position fix over a search area of several kilometres without significantly increasing the number of particles used, and also proved the algorithm's robustness.

Until recently, some difficulties have not been fully solved in BSLAM. They include the obvious overlap of the bathymetric data that could not be obtained for adjacent time periods, and this makes it difficult to measure the inter-frame motion. Also, loop closures obtained by terrain matching have a great impact on navigation results. Navigation results for vehicles are inaccurate if inaccurate loop closures cannot be identified.

To overcome these difficulties, the BSLAM algorithm based on the graph method is proposed. To estimate the inter-frame motion, the BSLAM algorithm constructs weak data association by comparing prior estimation and measurement, and estimation is obtained by a maximum likelihood terrain estimation approach. A weighted voting method for the loop closures has also been proposed to identify inaccurate loop closures. In contrast to the filter BSLAM methods proposed by Barkby et al. (2012) and Palomer et al. (2013), our proposed algorithm does not need to make strong assumptions of state transfer and measurement functions, and this means the modelling errors caused by hydrodynamic force coefficients and complex measurement noise would have little impact on the BSLAM solutions. Meanwhile, the BSLAM algorithm calculates a posterior probability over the entire path along with the map instead of just the current pose-like filter methods (Thrun et al., 2005).

An experiment using a shipborne bathymetry system was conducted. On board data was collected, including an 8 km planned track recorded at 4 knots, and a high-resolution bathymetric map of around $0{\cdot}56\,km^2$ was built. Using this data, a playback experiment was conducted to verify the validity and real-time performance of the BSLAM algorithm, and the experiment results verified the viability, accuracy and real-time performance of the BSLAM.

The rest of paper is organised as follows. An overview of the BSLAM algorithm is introduced in Section 2. A description of the graph construction algorithm is given in Section 3. The graph optimisation algorithm is proposed in Section 4 and the playback experiment and its results are discussed in Section 5. Finally, the conclusions are given in Section 6.

2. OVERVIEW OF THE BSLAM ALGORITHM.

Similar to graph SLAM, the BSLAM algorithm also consists of both graph construction and optimisation. In the graph construction, the trajectory of the AUV is divided into various submaps, and loop closures between submaps are detected. AUV motion between adjacent frames is estimated via weak data association. Meanwhile, all loop closures are identified via the weighted voting method, and then the pose graph is built using AUV states, odometry constraints, weak

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

data associations and loop closures. In the graph optimisation process, all frames associated with loop closures are named as key frames. AUV states on key frames are corrected via global trajectory correction and all of the rest states are modified in the local trajectory correction process. The main BSLAM framework is shown as Algorithm 1.

Algorithm 1 BSLAM FRAMEWORK

```
1  Do
2    Import DR data, bathymetric data and Store them to submap i
3    Build weak data association
4    If the Difference of Normals (DoN) value or length of the submap i satisfies the
     conditions
5      Store submap i, and build submap i + 1
6      Detect loop closures and Judge all of them via weighted voting method
7      If new loop closure is detected
8        Do global trajectory correction
9        Do local trajectory correction
10     end if
11   end if
12 Until mission complete
```

As shown in Algorithm 1, the main steps of the BSLAM are:

(1) *Graph Construction* (Line 2 to 6 in Algorithm 1):
   (a) The navigation and bathymetric data is obtained via the DR and bathymetry systems, respectively, and they are divided into various submaps; (Section 3.1)
   (b) Weak data association construction; (Section 3.2)
   (c) Loop closures between the current submap and all historical submaps are detected by terrain matching technique, and inaccurate loop closures can be identified via a weighted voting method; (Section 3.3)

(2) *Graph Optimisation* (Line 8 to 9 in Algorithm 1):
   (a) Combined with the loop closures and weak data association, the purpose of the global trajectory correction is to find the solution of the SLAM least squares problems. (Section 4.2.1)
   (b) After obtaining the global trajectory correction results, local trajectory correction modifies the local trajectory with the aid of weak data association. (Section 4.2.2)

Moreover, in the BSLAM process, the vehicle track is designed to achieve the best results from both vehicle navigation and topographic mapping. As shown in Figure 1, our planned track for the AUV looks like a Chinese knot to obtain more loop closures. Later, we adopted this track pattern during the experiment, and results show the BSLAM algorithm performs well when using this track.

3. GRAPH CONSTRUCTION.

The purpose of graph construction is to build the pose graph. As shown in Figure 2, the pose graph of the BSLAM consists of the vehicle states

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

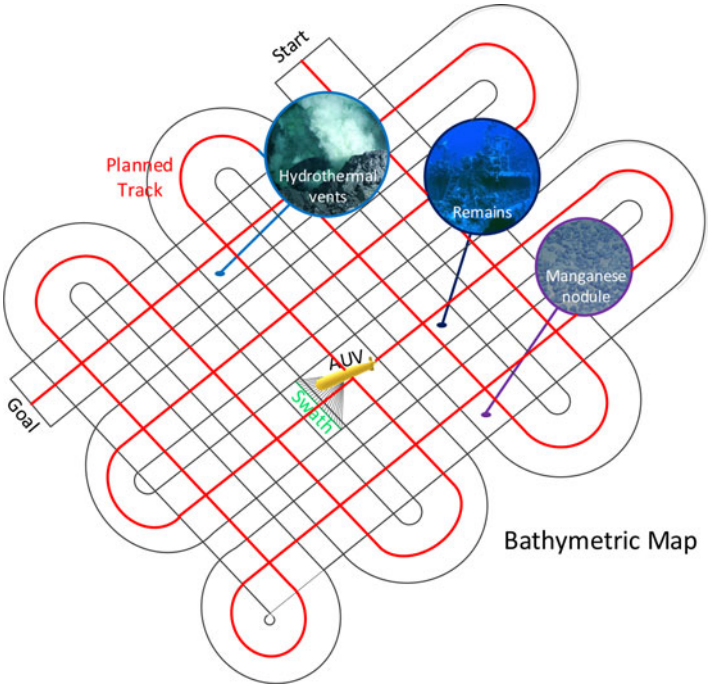

Figure 1. Planned track of the BSLAM process.

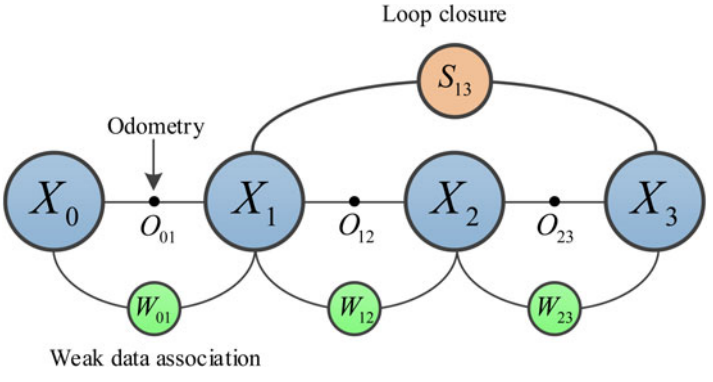

Figure 2. Schematic diagram of pose graph construction.

(○), odometer constraints (●), weak data association (○) and loop closures (○). The odometer constraint ● can be obtained from the navigation system, and the weak data association ○ is obtained using maximum likelihood terrain estimation. The loop closure ○ is detected by terrain matching techniques, and this closure is also called a loop closing constraint (Rosen et al., 2012).

3.1. *Generation of submaps.* In BSLAM, the first and most basic step is to log bathymetric data. The bathymetric data can be obtained by combining the Two-Dimensional (2D) swaths from a bathymetry system and vehicle trajectory from a DR system.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

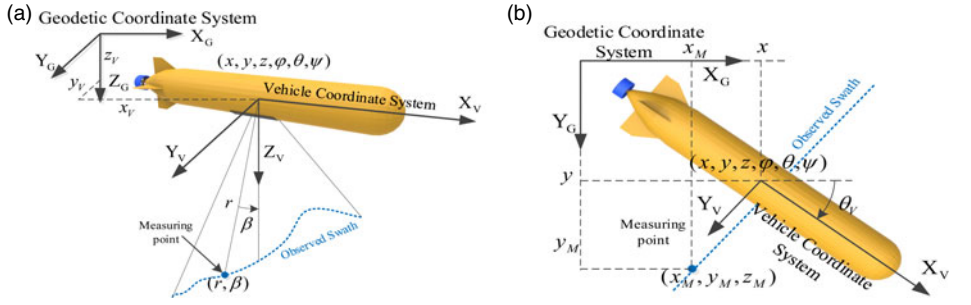

Figure 3. The measurement model of the AUV (a) Measurement model (b) 2D measurement model.

As shown in Figure 3, for one measurement point in the 2D swath, the bathymetry system would return the beam length r and the echo angle β. The corresponding trajectory point is $X = (x, y, z, \varphi, \theta, \psi)$, where x, y, z are the position in X_G (east), Y_G (north) and S_G (vertical) -axes in a geodetic coordinate system, and φ, θ, ψ denote the rotation around X_G, Y_G and Z_G -axes, respectively. Combined with the trajectory data, the corresponding bathymetric data $X_M = (x_M, y_M, z_M)$ in the geodetic coordinate system can be obtained via Equation (1):

$$\begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} = R \begin{bmatrix} 0 \\ r\sin\beta \\ r\cos\beta \\ 1 \end{bmatrix}$$

where

$$R = \begin{bmatrix} \cos\theta\cos\psi + \sin\varphi\sin\theta\sin\psi & \cos\varphi\sin\psi & -\sin\theta\cos\psi + \sin\varphi\cos\theta\sin\psi & x \\ -\cos\theta\cos\psi + \sin\varphi\sin\theta\sin\psi & \cos\varphi\cos\psi & -\sin\theta\sin\psi + \sin\varphi\cos\theta\cos\psi & y \\ \sin\theta\cos\psi & -\sin\varphi & \cos\theta\cos\psi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

It must be noted that, due to the measurement errors of a bathymetry system, noise would affect the bathymetric data and some outliers would be produced. To deal with this problem, the automatic processing algorithm proposed by Chen (2016) is applied to detect the outliers of bathymetric data in the process of bathymetric data acquisition. This algorithm uses the Alpha-Shapes model to detect the outliers in 2D swaths efficiently, and its viability and accuracy have been proved with both flat and complex terrain. In addition, although systematic errors of attitude and heading would also influence the bathymetric data, these impacts are negligible because the measurement error of the Fibre Optic Gyrocompass (FOG) used in this paper is less than 0·1°.

During the topographic mapping mission, the trajectory and the corresponding bathymetric data is constantly divided into various submaps considering the time-consumption problem (Palomer et al., 2016). Submaps are created on the basis of the trajectory length and the amount of information in the bathymetric data. The minimum and maximum lengths of the submap are set according to the actual situation. Considering the amount of information in the bathymetric data, Palomer et al. (2015) proposed a method using

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

DoN to identify the object when the measurement noise is not only considerably high, but also more scattered. DoN provides a multi-scale approach to process large unorganised Three-Dimensional (3D) point clouds for detecting the areas where the point cloud has significant changes, and it has been proven to be a computationally efficient method. Thus, in this study, DoN is introduced to represent the amount of information in bathymetric data, and the DoN value in measurement point i can be calculated as:

$$\Delta\hat{n}_i(p, r_1, r_2) = \frac{\hat{n}_i(p, r_1) - \hat{n}_i(p, r_2)}{2} \tag{2}$$

where, $\Delta\hat{n}_i(p, r_1, r_2)$ represents the DoN value, p denotes the point cloud of bathymetric data. $\hat{n}_i(p, r_1)$ and $\hat{n}_i(p, r_2)$ are the normal with support radius r_1 and $r_2(r_1 < r_2)$ at time I, and they are estimated by finding the tangent plane using the principal components of the points lying within a sphere of a fixed support radius. In this paper, we define the DoN value of one submap includes n measurement points as:

$$DoN = 2 \cdot \sum_{i=1}^{n} don_i \tag{3}$$

where:

$$don_i = \begin{cases} 1 & \Delta\hat{n}_i(p, r_1, r_2) > 0{\cdot}5 \\ 1 & \Delta\hat{n}_i(p, r_1, r_2) \leq 0{\cdot}5 \end{cases} \tag{4}$$

An increase in the DoN value means a clear change in the terrain is observed. When the DoN value of current submap is higher than the threshold and the minimum patch size has been reached, the current submap is stored and a new submap is started (Roman and Singh, 2010).

3.2. *Weak data association.* While obtaining and storing the bathymetric data, weak data association at adjacent time periods is constantly built to estimate the inter-frame motion. The weak data association is the likelihood that the measurement matches the prior estimation, and this can be calculated by a comparison of the measurement and a prior estimation of it.

As shown in Figure 3, the measurement model of bathymetry system at time i is:

$$Z_i^+ = h(X_i) + e_i \tag{5}$$

where Z_i^+ is measured terrain elevation, and e_i denotes measured noise of bathymetric data with a distribution of $N(0, 0{\cdot}4)$ according to the parameter of GS+ wide-swath bathymetry system used in this paper.

The prior estimation can be obtained with a maximum likelihood terrain estimation algorithm we have previously proposed, called the Terrain Elevation Measurement Extrapolation Estimation (TEMEE) (Li et al., 2017). By combining the historical data including $X_{i-n}, Z_{i-n}^+, \ldots, Z_{i-1}^+$, and the current vehicle state X_i, the estimated measurement Z_i^- can be obtained via the TEMEE algorithm. The details of the prior estimation are long and therefore omitted here for brevity.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

Then the weak data association between time $i-1$ and i can be expressed as:

$$\begin{aligned} likelihood &= p(X_i|X_{i-1}) \propto p(Z_i^+ - Z_i^-) \\ &= \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{C}_e)}} \exp\left(-\frac{1}{2}(Z_i^+ - Z_i^-)^T \mathbf{C}_e^{-1}(Z_i^+ - Z_i^-)\right) \\ &= \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\sum_{k=1}^{N}(Z_i^+[k] - Z_i^-[k])^2\right) \end{aligned} \tag{6}$$

where $\mathbf{C}_e = diag(\sigma_1^2, \sigma_2^2, \sigma_1^3, \ldots, \sigma_k^2)$ is a diagonal matrix consisting of the variance components of each sampling point, N represents the number of measurement points and $Z[k]$ is the elevation of the k-th measurement point. As the measurements of all the sampling points are considered independent of the others, assuming that measurement variance is the same in all beams, we can simply make $\mathbf{C}_e = \sigma_e^2\mathbf{I}$ and $\sigma_e^2 = 0{\cdot}4$ according to e_i in Equation (5) (Nygren, 2005).

3.3. *Loop closure detection.*

3.3.1. *Terrain matching.* Different from weak data association, loop closures have a high confidence level and they are detected by terrain matching techniques between submaps. Whenever a new submap is stored, the terrain matching procedure runs between the current submap and all the historical submaps in its search circle. The radius of the circle is the maximum navigation error in the current submap.

As shown in Figure 4, it is assumed that submap 1 is in the search circle of submap 2, then submap 2 would be divided into match units as shown in Figure 4, and each match unit looks for the most similar area in submap 1. For match unit *unit_k* in submap 2 and all M match units in submap 1, the likelihood function L_k is given as follows:

$$L_k = \max\left(\frac{1}{(2\pi\sigma_g^2)^N}\exp\left(-\sum_{n=1}^{N}(unit\_k_n - unit\_l_n)^2/2\sigma_g^2\right)\right) \quad (l = 1, 2, 3, \ldots, M) \tag{7}$$

where σ_g represents the covariance matrix of the measurement noise in the grid map (Zhou et al., 2017), but here it could be calculated as the average residual of the match unit, as well as N is the same as the corresponding values in Equation (6). If L_k is more than the *matching-threshold*, the terrain information would judge the correctness of the matching result. Here the terrain entropy is applied to represent the terrain information because it is insensitive to measurement noise. In the map with elevation value $h(i,j)$ at (i,j), terrain entropy H_T can be described as:

$$\begin{cases} P(i,j) = \dfrac{h(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} h(i,j)} \\ H_T = \dfrac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} P(i,j)\ln[P(i,j)] \end{cases} \tag{8}$$

Making m and n equal to 20, if the current terrain information H_T is no less than 2, this matching result is valid and we denote the corresponding loop closure as l_k (Feng, 2004). Finally, to reduce the computation, only one loop closure l_1 with maximum likelihood function is retained in one matching procedure. The number of other loop closures which have the same value or slight difference with l_1 is used to express the reliability of this retained loop closure.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

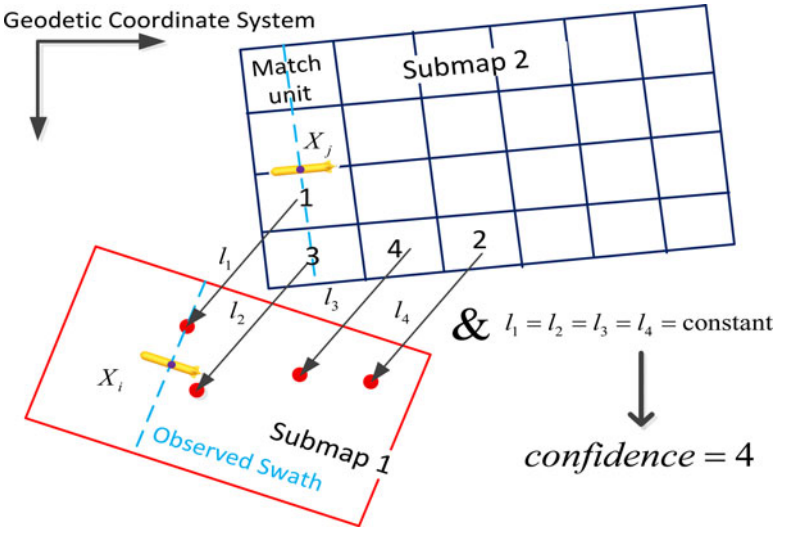

Figure 4. Calculation of confidence.

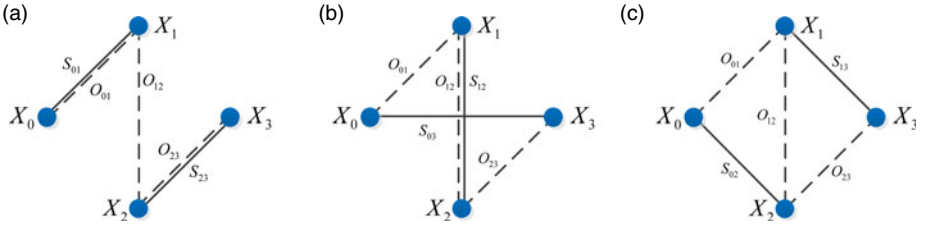

Figure 5. Network of combination of two loop closures (a) Network 1 (b) Network 2 (c) Network 3.

3.3.2. *Weighted voting method.* In the marine environment, the bathymetric data is influenced by the vehicle pose, reverberation and other environmental characteristics, and it can result in inaccurate loop closures (Nygren and Jansson, 2004; Wang et al., 2017b). If the inaccurate loop closures cannot be identified for a long time, the effectiveness of the BSLAM algorithm will be significantly affected. Thus, a weighted voting method is proposed to judge whether the loop closures are accurate.

In this method, each loop closure provides scores for each of the remaining closures in the survey process according to the corresponding combination of these two closures.

As shown in Figure 5, there are three kinds of possible networks for the combinations. ● is the vehicle state, and the black line connecting state X_i and X_j describes the loop closure with value D_{ij}^S and information matrix S_{ij}^{-1}, and the black dotted line connecting the same states denotes the odometer constraint with value D_{ij}^o and information matrix O_{ij}^{-1}.

For Network 1, the derived estimates of X_3 and its information matrix C_3^{-1} are:

$$\begin{aligned} X_3 &= (S_{01}^{-1} + O_{01}^{-1})^{-1}(S_{01}^{-1}D_{01}^s + O_{01}^{-1}D_{01}^o) + D_{12}^o + (S_{23}^{-1} + O_{23}^{-1})^{-1}(S_{23}^{-1}D_{23}^s + O_{23}^{-1}D_{23}^o) \\ C_3^{-1} &= [(S_{01}^{-1} + O_{01}^{-1})^{-1} + O_{12} + (S_{23}^{-1} + O_{23}^{-1})^{-1} \end{aligned} \tag{9}$$

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

For Network 2, the corresponding values are:

$$\begin{aligned} C'_{0123} &= O_{01} + (S_{12}^{-1} + O_{12}^{-1})^{-1} + O_{23} \\ D'_{0123} &= D^o_{01} + (S_{12}^{-1} + O_{12}^{-1})^{-1}(S_{12}^{-1} D^s_{12} + O_{12}^{-1} D^o_{12}) + D^o_{23} \\ X_3 &= (C'^{-1}_{0123} + S_{03}^{-1})^{-1}(C'^{-1}_{0123} D'_{0123} + S_{03}^{-1} D^s_{0123}) \\ C_3^{-1} &= C'^{-1}_{0123} + S_{03}^{-1}. \end{aligned} \tag{10}$$

Network 3 is in the form of a Wheatstone bridge, the variables X_3 can be solved from the linear system GX=B where X = $\{X_1, X_2, X_3\}$:

$$\begin{aligned} \mathrm{G} &= \begin{bmatrix} O_{01}^{-1} + O_{12}^{-1} + O_{23}^{-1} & -O_{12}^{-1} & -S_{13}^{-1} \\ -O_{12}^{-1} & S_{02}^{-1} + -O_{12}^{-1} + -O_{23}^{-1} & -O_{23}^{-1} \\ -S_{13}^{-1} & -O_{23}^{-1} & -S_{13}^{-1} + -O_{23}^{-1} \end{bmatrix} \\ \mathrm{B} &= \begin{bmatrix} O_{01}^{-1} D^o_{01} + O_{12}^{-1} D^o_{12} + S_{13}^{-1} D^s_{13} \\ S_{02}^{-1} D^s_{02} + O_{12}^{-1} D^o_{12} + O_{23}^{-1} D^o_{23} \\ S_{13}^{-1} D^s_{13} - O_{23}^{-1} D^o_{23} \end{bmatrix} \end{aligned} \tag{11}$$

The corresponding information matrix C_3^{-1} is:

$$C_3^{-1} = [O_{12}^{-1} \quad S_{13}^{-1}] \begin{bmatrix} O_{12}^{-1} + O_{23}^{-1} + S_{24}^{-1} & -O_{23}^{-1} \\ -O_{23}^{-1} & O_{23}^{-1} + O_{34}^{-1} + S_{13}^{-1} \end{bmatrix} \begin{bmatrix} S_{24}^{-1} \\ O_{34}^{-1} \end{bmatrix} \tag{12}$$

As S^{-1} is far greater than W^{-1}, information matrices of these networks can be simplified as:

$$\begin{aligned} &\text{Network 1: } C_3^{-1} = (S_{01} + O_{12} + S_{23})^{-1} \\ &\text{Network 2: } C_3^{-1} = (O_{01} + S_{12} + O_{23})^{-1} + S_{03}^{-1} \\ &\text{Network 3: } C_3^{-1} = [O_{12}^{-1} \quad S_{13}^{-1}] \begin{bmatrix} S_{24}^{-1} & O_{23}^{-1} \\ -O_{34}^{-1} & S_{13}^{-1} \end{bmatrix} \begin{bmatrix} S_{24}^{-1} \\ O_{34}^{-1} \end{bmatrix} \\ &\qquad\qquad = O_{12}^{-1} S_{24}^{-2} + O_{34}^{-1} S_{13}^{-2} - O_{23}^{-1} S_{13}^{-1} S_{24}^{-1} - O_{12}^{-1} O_{23}^{-1} O_{34}^{-1} \end{aligned} \tag{13}$$

The assumptions of the odometer constraint and loop closures have to be made in order to calculate C_3^{-1}. Referring to EKF theory, for a linear system without considering heading error, the observation update is omitted because there is no observation. The time update function of odometry is:

$$O_k = F_{k-1} O_{k-1} F_{k-1}^T + Q_{k-1} \tag{14}$$

in which F_{k-1} is the Jacobian matrix of the vehicle state transfer function. Hence, the increase of odometer covariance matrix O is achieved by a development of time t, and this means the odometer information matrix O^{-1} decreases as time goes on. O_{ij}^{-1} can be represented as $1/(t_j - t_i)$. For S_{ij}^{-1}, let S_{ij}^{-1} = 1,000 simply due to S_{ij}^{-1} being much bigger than the odometer information matrix. Finally, the corresponding C_3^{-1} value of network 1, 2 and 3 can be obtained according to Equation (13).

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

Now, combined with the DR location X_3^{dr}, the likelihood of X_3 can be calculated as:

$$likelihood = P(X_3) = \frac{C_3^{-1}}{\sqrt{2\pi}} \exp(-C_3^{-1} \|X_3 - X_3^{dr}\|^2) \tag{15}$$

Considering *confidence*, the score from D_1^s to D_2^s (or from D_2^s to D_1^s) is:

$$score_{1,2} = (confidence(1) + confidence(2)) - \frac{C_3^{-1}}{\sqrt{2\pi}} \exp(-C_3^{-1} \|X_3 - X_3^{dr}\|^2) \tag{16}$$

Algorithm 2 alg2

repeat
  Calculate $Score_t = \sum_{i=1, i \neq \ \ t}^{n} score_{t,i}, \quad t = 1, 2, \ldots, n$
  $t\min = find(Score_{t\min} == \min(Score_t))$
  if $Score_{t\min} < thre\text{-}2$
    Delete $Score_{t\min,i}, \quad i = 1, 2, \ldots, n$
    Delete $Score_{i,t\min}, \quad i = 1, 2, \ldots, n$
  end if
until $Score_{t\min} \leq thre\text{-}2$

As shown in Algorithm 2, the threshold *thre*-2 of the weighted voting method is the additive combination of the average value of $Score_t (t \in [1, n])$ and the corresponding standard deviation. If the smallest score is smaller than *thre*-2, this closure with the smallest score is judged as an inaccurate one. This method can delete all scores related to the closure with smallest score immediately, and in the next loop, voting results will not be influenced by the inaccurate one which has been identified.

4. GRAPH OPTIMISATION.

Once the new loop closure is detected, the graph optimisation process optimises the pose graph using global and local trajectory corrections.

4.1. *State transfer and measurement functions.*

The precondition of the optimisation process is to select the appropriate state transfer and measurement functions. There are many ways to generate the state transfer function of the AUV. Roman and Singh (2010) proposed the vehicle state model with six degrees of freedom. Stuckey (2012) proposed a state transfer function based on hydrodynamic coefficients. A simplified motion function has also been established to derive the weighted voting method.

The traditional SLAM algorithms, such as the EKF SLAM, predict X_i through the calculation of $f_i(X_{i-1}, u_i)$ (Durrant-Whyte and Bailey, 2006). However, the state transfer function $f_i(X_{i-1}, u_i)$ is not accurate whether we use the simplified state model or nonlinear state model based on the hydrodynamic coefficient. In fact, it is very difficult to build an accurate model of the vehicle state because of the influence of the coupled hydrodynamic coefficients. Thus, the state transfer function is replaced by the offset continuity function in BSLAM.

In the offset continuity function, the AUV states with loop closures are extracted according to the time order, and these states are call *key states*. DR errors of *key states* are defined

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

as $\Delta X^{key} = \{\Delta X_0^{key}, \Delta X_1^{key}, \ldots, \Delta X_n^{key}, \Delta X_{n+1}^{key}\}$, in which ΔX_i^{key} $(i = 1, 2, \ldots, n)$ is the error of all *n key states* during the mission. Meanwhile ΔX_0^{key} and ΔX_{n+1}^{key} are the error at the start and end points, respectively. It is important to note that vehicle states mentioned in Section 4.1 are all *key states*.

The offset of *key states* is accumulated over time because the DR error is accumulated. Therefore, we represent the current DR offset using the previous and subsequent ones. Assuming that the DR error at time i is expressed by DR errors at times $i-1$ and $i+1$, so $f_i(X_{i-1}^{key}, u_i) - X_i^{key}$ is represented by $a_i \Delta X_{i-1}^{key} + (1-a_i)\Delta X_{i-1}^{key} - \Delta X_1^{key}$. a_i is calculated using all vehicle states instead of only *key states*. If X_{i-1}^{key}, X_i^{key} and X_{i-1}^{key} are denoted as X_k, X_{k+m} and X_{k+m+n}, combined with weak data association $p(X_j | X_{j-1})$:

$$a_i = \frac{\sum_{j=k+1}^{k+m} 1/p(X_j | X_{j-1})}{\sum_{j=k+1}^{k+m+n} 1/p(X_j | X_{j-1})} \tag{17}$$

Please note that we assume the initial position of the AUV is accurate.

$$\Delta X_0^{key} = 0 \tag{18}$$

The offset continuity function of *key state i* is:

$$\Delta X_i^{key} = \begin{bmatrix} a_i & 0 \\ 0 & 1-a_i \end{bmatrix} \begin{bmatrix} \Delta X_{i-1}^{key} \\ \Delta X_{i-1}^{key} \end{bmatrix} \tag{19}$$

We can express the offset continuity function in a matrix form as:

$$\Delta \mathrm{X}^{key} = \mathrm{H} \Delta \mathrm{X}^{k} ey \tag{20}$$

where ΔX^{key} is a vector which is the concatenation of $\Delta X_0^{key}, \Delta X_1^{key}, \ldots, \Delta X_{n+1}^{key}$ and H denotes the incidence matrix with all entries being a, $1-a$, or 0. In addition, the relationship between DR offsets ΔX_{n+1}^{key} with other offsets need to be determined to fill the n+1 line in matrix H. When the time interval between ΔX_n^{key} and ΔX_{n+1}^{key} is quite short (less than 120 seconds), just make ΔX_{n+1}^{key} equal to ΔX_n^{key}, otherwise, make no assumption of ΔX_{n+1}^{key} and set all the elements of the n+1 line in H to 0.

On the other hand, the measurement function of loop closures is:

$$D_{ij}^{S} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta X_i^{key} \\ \Delta X_j^{key} \end{bmatrix} \tag{21}$$

With loop closure l_{ij} obtained in the terrain matching process, an observation of D_{ij}^{S} is modelled as:

$$\bar{D}_{ij}^{S} = D_{ij}^{S} + \hat{D}_{ij}^{S} = \Delta X_i^{key} \Delta X_j^{key} = l_{ij} + \hat{D}_{ij}^{S} \tag{22}$$

where $\hat{D}_{ij}^{S}$ is a random Gaussian error with zero mean and known covariance matrix S_{ij}. Given a set of measurements $\bar{D}_{ij}^{S}$, our goal is to derive the optimal estimate of all *key states*.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

We can express the measurement function in a matrix form as:

$$D^S = \mathrm{H}^S \Delta \mathrm{X}^{key} \tag{23}$$

where D^S represents the concatenation of all the position differences of D^S_{ij}, ΔX^{key} denotes the concatenation of $\Delta X_1^{key}, \Delta X_2^{key}, \Delta X_3^{key}, \ldots$, and H^S is the incidence matrix with all entries being 1, -1, or 0.

4.2. *Optimisation algorithm.*

4.2.1. *Global trajectory correction.* Combining the state transfer function (Equation (20)) and measurement function (Equation (23)), the BSLAM problem is transformed into a least squares problem:

$$\Delta X^{key*} = \arg\min \times \left\{ \sum_{i=1}^{M} \left\| a_i \Delta X_{i-1}^{key} + (1 - a_i) \Delta X_{i+1}^{key} - \Delta X_{i-1}^{key} \right\|^2 + \sum_{j=1}^{N} \left\| \Delta X_{j,1}^{key} - \Delta X_{j,2}^{key} - l_j \right\|_{c_j}^2 \right\} \tag{24}$$

where ΔX^{key*} are the modified AUV *key states*, $\Delta X^{key} \in \mathrm{R}^m$ denotes the DR offset of vehicle *key states* ($m = Md_x$ and d_x represents the dimensions of the vehicle state), as well as $\Delta X_{j,1}^{key}$ and $\Delta X_{j,2}^{key}$ are two *key states* associated with the loop closure l_j. We then collect the coefficients of ΔX^{key} into a large but sparse measurement matrix $A \in \mathrm{R}^{m \times n}$ where $m = (M + N)d_X$, and collecting the vectors l into the bottom of $d \in \mathrm{R}^n$. Equation (24) can be represented as:

$$\Delta X^{key*} = \arg\min \|A \Delta \mathrm{X}^{key} - b\|^2 \tag{25}$$

Therefore, the least squares problem can be simplified as follows:

$$A \Delta \mathrm{X}^{key} = b \tag{26}$$

Kaess et al. (2008) applied the standard QR decomposition to matrix A to solve Equation (26); however, the state model is linear in this study. Hence, we use the normal equation:

$$A^T A \Delta \mathrm{X}^{key} = A^T b \tag{27}$$

to solve this problem. Accordingly:

$$\Delta \mathrm{X}^{key} = (A^T A)^{-1} A^T b \tag{28}$$

4.2.2. *Local trajectory correction.* After obtaining the global correction results, all vehicle states between two adjacent key states are represented as a local trajectory. In the local trajectory with offsets of key points ΔX_i and ΔX_{i+end}, the DR offset ΔX_{i+j} at time $i + j$ can be calculated as follows:

$$\Delta \mathrm{X}_{i+j} = \Delta X_i + \left(\sum_{n=1}^{j} a_n / \sum_{n=1}^{end} a_n \right) .(\Delta X_{i+end} - \Delta X_i). \tag{29}$$

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

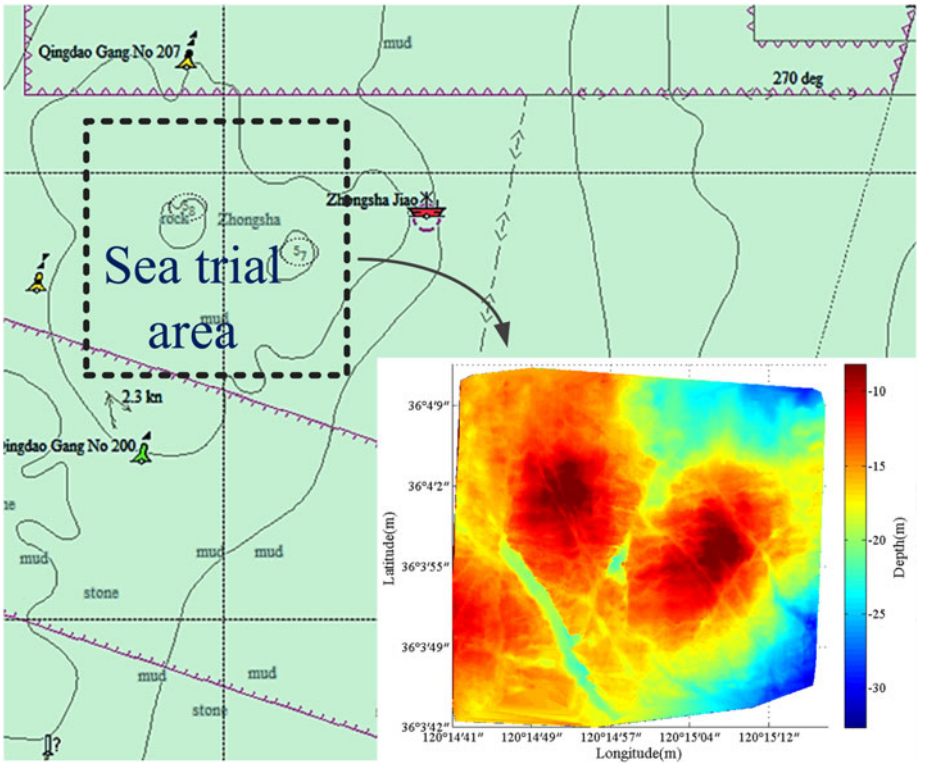

Figure 6. Sea Trial area around Zhongsha Jiao.

5. PLAYBACK EXPERIMENT. To test the algorithm, a simulation system was built in MATLAB to read time-stamped historical vehicle and sensor data log files and generate sequences of input data messages for the BSLAM algorithm.

5.1. *Acquisition of experimental data.* As shown in Figure 6, to obtain the bathymetric and navigation data, a sea trial experiment was conducted around Zhongsha Jiao in October 2016. In this experiment, a GS+ wide-swath bathymetry system was installed beneath the ship. This system was aided by a sonar altimeter (to provide depth gauge data) and a Mini-Sound Velocity Sensor (SVS) (to provide acoustic velocity data) from Valeport Company, and finally the bathymetry system provided the bathymetric data. The vehicle states were provided by a Global Positioning System (GPS) receiver (longitude and latitude) and a FOG (heading, pitch and roll). The data collected on board includes an 8 km planned track recorded at a speed of 4 knots. (Figure 7(a) shows the V-plate with transducers and Mini-SVS, and Figures 7(b) and (c) show the main module and FOG, respectively.)

The DR data X_{DR} was simulated using GPS data X_{GPS} along with the white noise, that is:

$$X_{DR} = X_{GPS} + Q, \tag{30}$$

where Q is the random drift with a distribution of $p(Q) = \mathrm{N}(0, 2)$. It is believed that the FOG is always accurate in the experiment to decrease computation so make $\theta_{DR} = \theta_{GPS}$.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

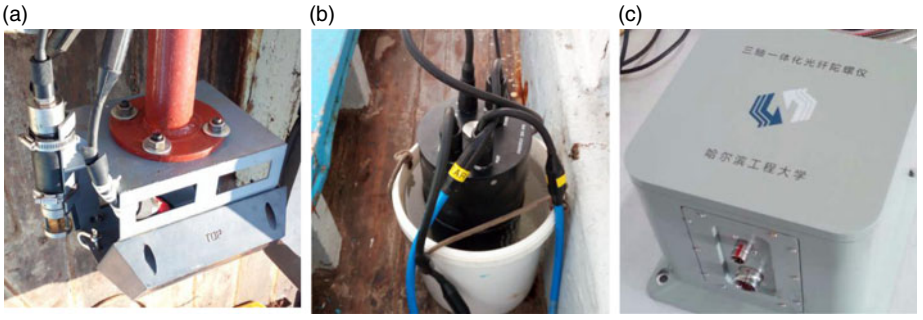

Figure 7. Devices used in the sea trial experiment (a) V-plate with transducers and Mini-SVS (b) Main module of GS+ (c) Fibre optic gyrocompass.

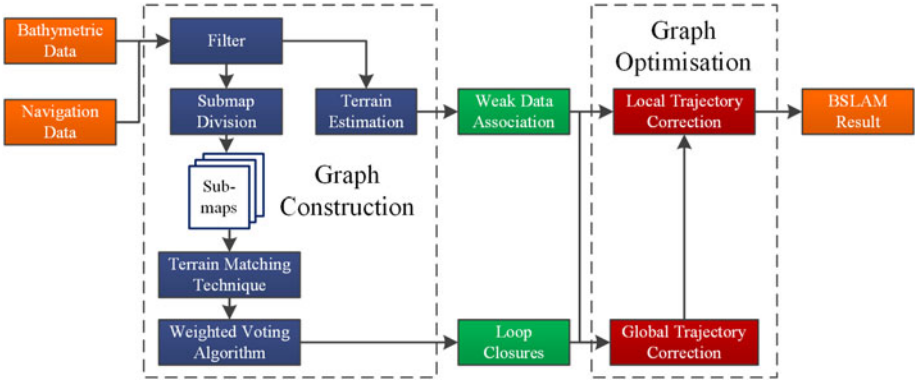

Figure 8. The architecture of the BSLAM simulation system.

5.2. *BSLAM experiment.* The simulation experiments were conducted using the simulation system based on MATLAB and running on a computer with an Intel I5 3210M CPU and 4 GB of memory.

As shown in Figure 8, the data collected on board was used to update the navigation bathymetric data at a frequency of 1 Hz in the simulation system, and the BSLAM result was exported at the same frequency.

The submap division results are shown in Figure 9(a). The X-axis is marked as the starting time for each new submap, and the Y-axis shows the swath length. The minimum and maximum lengths of the submap are set to 40 and 102 seconds and we set the DoN threshold to 300. It is obvious that the lengths of submaps are relatively short in the region with complex terrain changes in Figure 9(a).

In the loop closure detection procedure, the *matching-threshold* was set to 0·06 according to the measurement noise of the GS+. As shown in Figure 9(b), ○ denotes the vehicle *key point* connected by the blue line in the time order, and the black line represents the connection from the *key point* to its matching point. The green and red dashed lines indicate the correct and wrong loop closures identified by the weighted voting method, respectively, where their *confidences* are represented by the width. The number i on the red line indicates that the corresponding loop closure was identified as invalid in the i-th iteration in the weighted voting algorithm.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

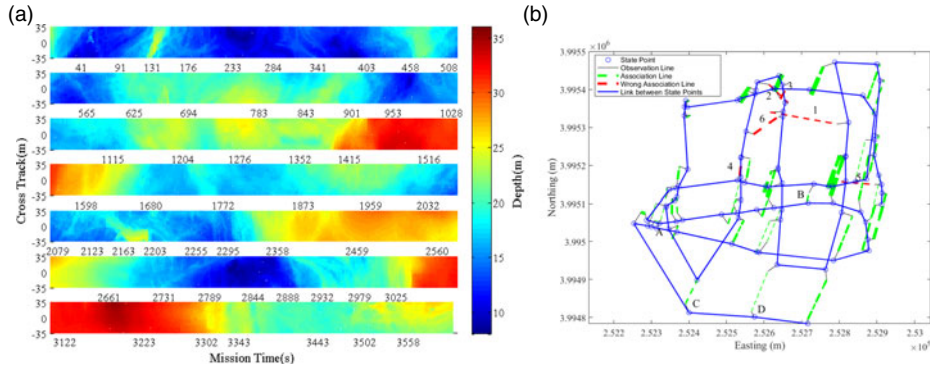

Figure 9. Graph construction results (a) Results of submap division (b) Results of weighted voting method.

It can be seen that all the inaccurate loop closures have been identified and the validity of the weighted voting method has been proved. However, there are still some problems, for example, a few accurate loop closures such as red line 3 have also been identified by mistake if they are close to some inaccurate ones, because these inaccurate loop closures could not be identified in time. However, in general, the weighted voting algorithm helps achieve the desired results, and the accurate loop closures can be used to build the pose graph.

Figures 10(a), (c) and (e) show the bathymetric maps built by DR, incomplete BSLAM (BSLAM without local trajectory correction, and this means all of the weak data associations $p(X_i|X_{i-1})$ are set to 1 in Equation (17)) and the BSLAM, respectively. The corresponding registration error histograms are shown in Figures 10(b), (d) and (f). The error histograms are drawn by counting all measurement point registration errors, and for each individual measured point, the registration error can be calculated as:

$$E_i = \sqrt{(x^i_{SLAM} - x^i_{GPS})^2 + (y^i_{SLAM} - y^i_{GPS})^2} \tag{31}$$

where x^i_{GPS} is the truth location for measured point I, and x^i_{SLAM} defines the point position corrected using the BSLAM result in the easting-northing surface.

As shown in Figure 10, the average of the registration errors in the maps produced by DR, incomplete BSLAM and the BSLAM were calculated as 72·62, 13·74 and 12·25 m, respectively. The corresponding median values were 65·98, 12·68 and 11·27 m. The experiment results prove that both the global and local trajectory corrections in the BSLAM have a clear effect on the mapping results. Compared with incomplete BSLAM, the local trajectory correction helps reduce the mean error by 10·82% and median error by 8·87%. The BSLAM algorithm helps reduce the mean error by 83·13% and median error by 82·92% compared to the DR results.

To verify the real-time performance of the BSLAM algorithm, a playback experiment with the input of the bathymetric and navigation data was conducted online, and this means all parts of the BSLAM including graph construction and optimisation were performed in real time.

Figure 11 shows the real-time registration errors of DR and BSLAM. Although the first loop closure was detected at A, the error did not decrease because of insufficient association and lack of information. Due to the same reason, the error decreased at B but the decrease

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

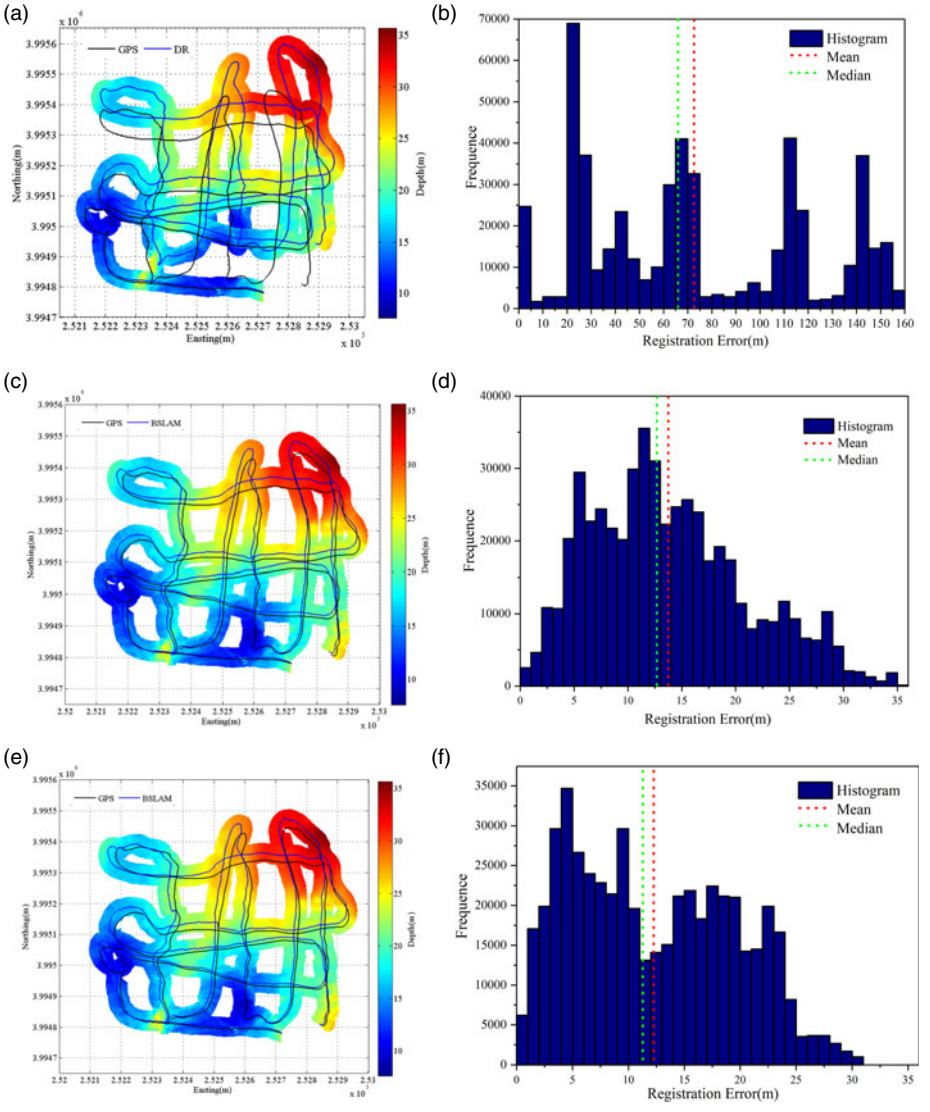

Figure 10. Graph optimisation results in the BSLAM (a) DR bathymetric map (b) DR error history (c) Incomplete BSLAM (without weak data association) bathymetric map (d) Incomplete BSLAM (without weak data association) bathymetric map (e) BSLAM bathymetric map (f) BSLAM error history.

was slight. The loop closure between the current state and the initial state was detected at C, so the registration error reduced rapidly and converged to 10 m after state D. When the trajectory is associated with the initial state, the overall error converges to an acceptable range.

In addition to the real-time registration error, the time consumed is also an important part in judging the real-time performance of the BSLAM. First, we define that a good real-time performance of BSLAM is when BSLAM can complete the calculations including graph

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

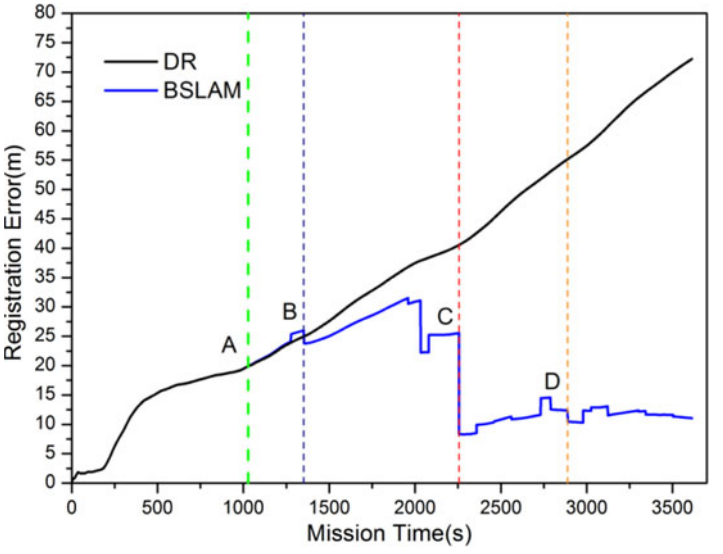

Figure 11. Registration error in real time.

construction and optimisation of all historical submaps before the next submap is stored. We describe the time period to build a submap #*i* as calculation period #*i*.

As shown in Figure 12(a), the available time is the whole length of current period, and terrain matching (which is the most time consuming), graph construction (except terrain matching) and graph optimisation constitute the process of the BSLAM.

Time consumed in terrain matching is quite small at the beginning owing to the small search circle, but with the time increasing, the search circle expands so the matching time consumed also increases. In the MATLAB environment, the terrain matching is quite time consuming, but in practical engineering applications, real-time terrain matching has been proved by Hagen and Ånonsen (2014).

For the other part of graph construction, time consumed accounts for only 2·9% of the total length of the current period. Figure 12(b) shows the time consumed for graph optimisation. After the first data association, the time consumed increases but the maximum value is still less than 0·09 seconds per second of the BSLAM mission.

In summary, BSLAM time consumption increases with time but is still obviously less than the available time for most periods. Although there is 7·8, 32·4 and 12 seconds delay at periods #48, #49 and #50, all the time delay disappears at period #51. Hence, it is proved that the BSLAM has a good real-time performance.

6. CONCLUSIONS.

In this paper, a BSLAM algorithm including graph construction and optimisation is proposed, and a play-back experiment has proved the validity, real-time performance and high mapping precision of BSLAM. The following conclusions can be made:

(1) Although local trajectory correction only provides corrections on the basis of the global trajectory correction results, it is an essential part and has a great impact on the final navigation and mapping results.

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

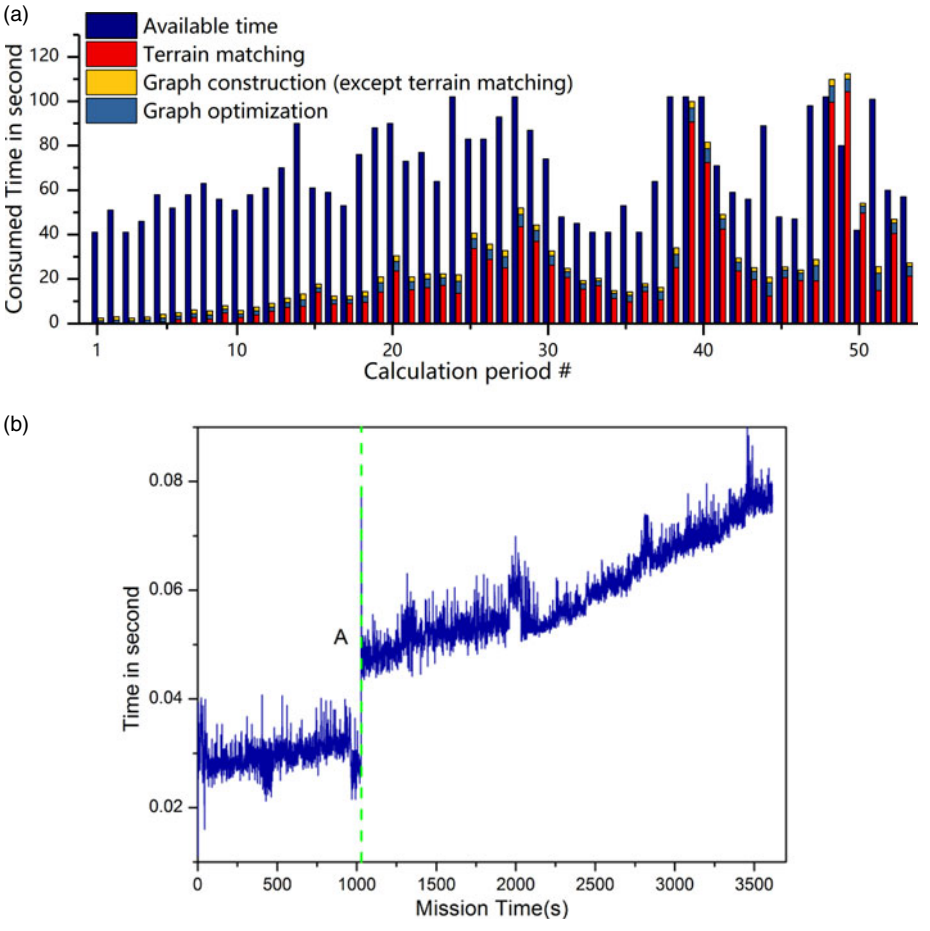

Figure 12. Time consumed results (a) Total time consumed in the BSLAM (b) Time consumed in graph optimisation.

(2) The BSLAM registration error is convergent after obtaining sufficient data associations. The error in the whole trajectory is related to the error at the first *key state*; hence, the minimum error and best mapping effect will be obtained when the loop closure between the current state and the initial one is detected.
(3) Time consumed by BSLAM is obviously less than the available time for most calculation periods and nearly no time delay exists after the vehicle has sailed around 8 km during one hour, and this result proves the real-time performance of the BSLAM.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China (2017YFC0305700), open fund project of the Qingdao National Laboratory for Marine Science and Technology (QNLM2016ORP0406) and National Natural Science Foundation of China (51309066/E091002).

https://doi.org/10.1017/S0373463319000286 Published online by Cambridge University Press

REFERENCES

Ånonsen, K. B., Hagen, O. K., Hegrenæs, Ø. and Hagen, P. (2013). The HUGIN AUV Terrain Navigation Module. *MTS/IEEE OCEANS - San Diego*, San Diego, CA, 1–8.

Barkby, S., Williams, S., Pizarro, O. and Jakuba, M. (2009). An efficient approach to bathymetric SLAM. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 219–224.

Barkby, S., Williams, S. B., Pizarro, O. and Jakuba, M. V. (2012). Bathymetric Particle Filter SLAM Using Trajectory Maps. *International Journal of Robotics Research*, **31**(12), 1409–1430.

Chen, P. (2016). *Study on Seabed Terrain Matching Navigation with Multi-Senor for AUV*. Ph.D. dissertation, Harbin Engineering University. (in Chinese)

Dellaert, F. and Kaess, M. (2006). Square Root SAM : Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, **25**(12), 1181–1204.

Doble, M. J., Forrest, A. L., Wadhams, P. and Laval, B.E. (2009). Through-ice AUV Deployment: Operational and Technical Experience from Two Seasons of Arctic Fieldwork. *Cold Regions Science & Technology*, **56**(2–3), 90–97.

Donovan, G. T. (2012). Position Error Correction for an Autonomous Underwater Vehicle Inertial Navigation System (INS) Using a Particle Filter. *IEEE Journal of Oceanic Engineering*, **37**(3), 431–445.

Durrant-whyte H. and Bailey T. (2006). Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, **13**(3), 108–117.

Feng, Q. T. (2004). *The Research on New Terrain Elevation Matching Approaches and Their Applicability*. Ph.D. dissertation, Nation University of Defense Technology. (in Chinese)

Hagen, O. K. and Ånonsen, K. B. (2014). Using Terrain Navigation to Improve Marine Vessel Navigation Systems. *Marine Technology Society Journal*, **48**(2), 45–58(14).

Hurtós, N., Ribas, D., Cufí, X., Petillot, Y. and Salvi, J. (2015). Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments. *Journal of Field Robotics*, **32**(1), 123–151.

Ila V., Polok L., Solony M. and Svoboda P. (2017). SLAM++ -A highly efficient and temporally scalable incremental SLAM framework. *International Journal of Robotics Research*, **36**(3), 210–230.

Johannsson, H., Kaess, M., Englot, B., Hover, F. and Leonard, J. (2010). Imaging sonar-aided navigation for autonomous underwater harbor surveillance. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan.

Kaess, M., Ranganathan, A. and Dellaert, F. (2008). iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics*, **24**(6), 1365–1378.

Kim, A. and Eustice, R. M. (2013). Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency. *IEEE Transactions on Robotics*, **29**(3), 719-733.

Kownacki C. (2016). A concept of laser scanner designed to realize 3D obstacle avoidance for a fixed-wing UAV. *Robotica*, **34**(2):243–257.

Li, Y., Ma, T., Wang, R., Chen, P. and Zhang, Q. (2017). Correction Method for AUV Seabed Terrain Mapping. *Journal of Navigation*, **70**, 1062–1078.

Mallios, A., Ridao, P., Ribas, D. and Hernández, E. (2014). Scan Matching SLAM in Underwater Environments. *Autonomous Robots*, **36**(3), 181–198.

Nygren, I. and Jansson, M. (2004). Terrain Navigation for Underwater Vehicles Using the Correlator Method. *IEEE Journal of Oceanic Engineering*, **29**(3), 906–915.

Nygren, I. (2005). *Terrain Navigation for Underwater Vehicles*. Ph.D. dissertation, Royal Institute of Technology.

Palomer, A., Ridao, P., Ribas, D., Mallios, A., Gracias, N. and Vallicrosa, G. (2013). Bathymetry-based SLAM with Difference of Normals Point-cloud Subsampling and Probabilistic ICP Registration. *MTS/IEEE OCEANS - Bergen*, Bergen, Norway, 1–7.

Palomer, A., Ridao, P., Romagós, D. R. and Vallicrosa, G. (2015). Multi-beam Terrain/Object Classification for Underwater Navigation Correction. *MTS/IEEE OCEANS – Genova*, Genova, Spain, 1–5.

Palomer, A., Ridao, P. and Ribas, D. (2016). Multibeam 3D Underwater SLAM with Probabilistic Registration. *Sensors*, **16**, 560.

Paull, L., Saeedi, S., Seto, M. and Li, H. (2014). AUV Navigation and Localization: A Review. *IEEE Journal of Oceanic Engineering*, **39**, 131–149.

Ribas, D., Ridao, P., Tardós, J. D. and Neira, J. (2008). Underwater SLAM in Man-made Structured Environments. *Journal of Field Robotics*, **25**(11–12), 898–921.

Roman, C. and Singh, H. (2010). A Self-Consistent Bathymetric Mapping Algorithm. *Journal of Field Robotics*, **24**(1–2), 23–50.

Rosen, D. M., Kaess, M. and Leonard, J. J. (2012). An incremental trust-region method for Robust online sparse least-squares estimation, *2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 1262–1269.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press, Inc.

Stuckey, R. A. (2012). Navigational Error Reduction of Underwater Vehicles with Selective Bathymetric SLAM. *Navigation Guidance & Control of Underwater Vehicles*, **45**(5), 118–125.

Wang, N., Lv, S., Zhang, W., Liu, Z. and Er, M. J. (2017a). Finite-time observer based accurate tracking control of a marine vehicle with complex unknowns. *Ocean Engineering*, **145**, 406–415.

Wang, N., Su, S. F., Yin, J., Zheng, Z. and Meng, J. E. (2017b). Global Asymptotic Model-Free Trajectory-Independent Tracking Control of an Uncertain Marine Vehicle: An Adaptive Universe-Based Fuzzy Control Approach. *IEEE Transactions on Fuzzy Systems*, DOI:10.1109/TFUZZ.2017.2737405.

Zhou, L., Cheng, X., Zhu, Y., Dai, C. and Fu, J. (2017). An Effective Terrain Aided Navigation for Low-Cost Autonomous Underwater Vehicles. *Sensors*, **17**, 680.