

Article

# A Real-Time Mismatch Detection Method for Underwater Database-Referenced Navigation

Tian Dai , Lingjuan Miao and Yanbing Guo 

School of Automation, Beijing Institute of Technology, Beijing 100081, China; miaolingjuan@bit.edu.cn (L.M.); 20080089@bit.edu.cn (Y.G.)

\* Correspondence: daitian@bit.edu.cn; Tel.: +86-177-105-88879

Received: 3 December 2018; Accepted: 10 January 2019; Published: 14 January 2019



**Abstract:** Database-referenced navigation (DBRN) using geophysical information is often implemented on autonomous underwater vehicles (AUVs) to correct the positional errors of the inertial navigation system (INS). The matching algorithm is a pivotal technique in DBRN. However, it is impossible to completely eliminate mismatches in practical application. Therefore, it is necessary to perform a mismatch detection method on the outputs of DBRN. In this paper, we propose a real-time triple constraint mismatch detection method. The proposed detection method is divided into three modules: the model fitting detection module, the spatial structure detection module, and the distance ratio detection module. In the model fitting detection module, the navigation characteristics of AUVs are used to select the fitting model. In the spatial structure detection module, the proposed method performs the mismatch detection based on the affine transformation relationship between the INS-indicated trajectory and the corresponding matched trajectory. In the distance ratio detection module, we derive the distance ratio constraint between the INS-indicated trajectory and the corresponding matched trajectory. Simulations based on an actual geomagnetic anomaly base map have been performed for the validation of the proposed method.

**Keywords:** database-referenced navigation; mismatch detection; underwater navigation; inertial navigation system; autonomous underwater vehicles

## 1. Introduction

Autonomous underwater vehicles (AUVs) are widely used in a variety of tasks, including oceanographic surveys, bathymetric data collection in marine and riverine environments, patrol and reconnaissance missions in the military field, and rescue duties [1]. The availability of a precise and robust navigation system is the foundational prerequisite for the vehicle to successfully execute missions [2,3]. The quality of the navigation system not only influences the position errors between the desired path and the executed path, but also affects the outcome of the georeferencing process of the data acquired by the onboard sensors (e.g., the acquisition of terrain and geomagnetic data) [4].

In general, an AUV uses an inertial navigation system (INS) as the primary navigation system [5]. INS is a navigation aid that uses accelerometers and gyroscopes to continuously calculate by dead reckoning the position, the orientation, and the velocity (direction and speed of movement) of a moving object without the need for external references. Nevertheless, INSs have positional errors growth that is unbounded. This effect can be reduced by using accurate acceleration, heading, and velocity sensors, but these sensors cannot be made arbitrarily accurate. During long endurance underwater missions, these inaccuracies become significant [6]. Therefore, other navigation and positioning technologies are adopted by AUVs as auxiliary navigation systems. Above water, the global navigation satellite system (GNSS) is the most commonly used method to compensate for INS-accumulated errors [7]. However, due to the rapid attenuation of higher frequency signals and the unstructured nature of

the undersea environment, GNSS signals can only propagate within short distance under water [1]. Database-referenced navigation (DBRN) systems, which use geophysical information such as gravity, magnetic, and terrain data for navigation, hold strong potential as underwater auxiliary navigation systems [8,9]. Take the terrain-referenced navigation (TRN) system (one of the most popular DBRN systems) as an example. TRN obtains a range measurements by using sonar sensors installed on a vehicle. These measurements are matched with a priori digital map of the terrain elevation, to estimate the vehicle position. The principle of gravity-referenced navigation (GRN) and magnetic-referenced navigation (MRN) is the same as TRN, except for the priori digital maps and sensors used. GRN uses gravity anomaly maps and gravimeters as a priori digital maps and sensors. MRN uses geomagnetic anomaly maps and magnetometers as a priori digital maps and sensors. DBRN systems are completely passive and difficult to interfere with; these advantages totally meet the requirements of underwater vehicles [10].

A DBRN system consists of a priori geophysical information database (terrain elevation maps, gravity anomaly maps, etc.), a measurement unit (sonar, gravimeter, etc.), and a navigation algorithm [11,12]. The navigation algorithm is one of the key factors in DBRN [13,14]. There are two types of navigation algorithms in DBRN systems: filtering algorithms and matching algorithms. Filtering algorithms make use of the INS model and the measurement model to construct filters. In DBRN systems, commonly used filtering algorithms are the extended Kalman filter (EKF), the particle filter (PF) and the Rao-Blackwellized particle filter (RBPF) [15–18]. EKF is more effective for low nonlinear estimation problems. A PF can effectively handle highly nonlinear or non-Gaussian estimation problems. RBPF is a hybrid filter combining EKF and PF. Terrain contour matching (TERCOM) algorithm and iterated closest contour point (ICCP) algorithm are two conventional matching algorithms in DBRN systems [12,19]. TERCOM algorithm, realized via group correlation analysis, possesses better robustness. The ICCP algorithm, which uses rigid transformation to match the multilateral arc, has a high matching accuracy when the initial INS error is small. It should be noted that matching algorithms-based DBRN systems can operate effectively on the premise that the matching area has obvious features. Features can be discrete landmark objects, numerical changes, or texture variations in the digital map. If the features are smooth, the above matching algorithms have a high probability of mismatching [19,20]. Since it is impossible to completely eliminate mismatches, a mismatch detection method after the matching algorithm deserves to be studied.

Mismatch detection methods are widely used in the field of feature points-based image registration. In DBRN systems, the actual trajectory and the DBRN output trajectory are denoted as the original image and the transformed image, respectively, then mismatch detection algorithms in image registration can be used to solve the mismatch detection problem in DBRN systems. Random sample consensus (RANSAC)-based algorithms [21–23] and graph transformation matching (GTM)-based algorithms [24,25] are two types of commonly used mismatch detection methods in the field of feature points based image registration. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. Therefore, it can be interpreted as an outlier detection method. GTM uses the distribution of matching points and the neighborhood relationship in the local area to detect mismatched points. RANSAC based algorithms and GTM based algorithms can be directly used for detecting mismatched points in DBRN systems. In addition to the above two commonly used types of algorithms, the restricted spatial order constraints (RSOC) algorithm [26] is also a simple and robust outlier detection algorithm. In RSOC, the two-way spatial order constraints are used to determine the candidate outliers, whereas the two decision criteria restrictions are used to confirm which dubious match pairs should be removed. Han et al. [9] proposed an RSOC-based mismatch diagnostic method specifically for an underwater navigation system. The spatial structure constraints and transformation error restrictions are utilized to select outliers in a matched sequence. However, there exists several disadvantages of using these algorithms in DBRN systems. On one hand, these approaches can only detect a certain type of mismatch, RANSAC-based algorithms cannot detect

the mismatched point that fits the model, GTM based algorithms and RSOC based algorithms cannot detect the mismatched point with the same structure. On the other hand, applying these approaches requires sampling enough points, which leads to poor real-time performance.

In order to solve the problem of a single detection mode and poor real-time performance, we suggest a real-time triple constraint mismatch detection method. The proposed method, considering the model fitting constraint, spatial structure constraint, and distance ratio constraint, is specifically presented to deal with mismatched point detection problem in matching algorithms-based DBRN systems for AUVs. In model-fitting detection module, the proposed method eliminates points that deviate significantly from the matched trajectory. In the spatial structure detection module, the K-nearest neighbor (KNN) graph of the INS-indicated trajectory and the matched trajectory are established. The affine transformation relationship between the INS-indicated trajectory and the corresponding matched trajectory is used to determine whether the current matched point is an abnormal point. In the distance ratio detection module, the underwater INS error propagation model is established to calculate the distance ratio constraint between the actual trajectory and the INS-indicated trajectory. The matched points beyond the constraint range are regarded as outliers. It should be noticed that the above three detection processes perform mismatch detection point-by-point instead of detecting all mismatched points at one time. These are some important contributions for our work.

The rest of the paper is organized as follows. The second part briefly describes M-estimator sample consequence (MSAC) and weighted graph transformation matching (WGTM). The INS error propagation model employed in this paper is described in the third part. The fourth part presents the procedure of the proposed method. A comprehensive discussion on the experimental settings and simulation results are provided in the fifth part. Conclusions are given in the last part.

## 2. MSAC and WGTM

### 2.1. RANSAC and MSAC

RANSAC is a robust approach that is used in many applications for extracting shapes and for estimating the model parameters from data that contains outliers. RANSAC classifies data into inliers (data that can be described by the model) and outliers (data that deviate from the normal range of the model or fail to meet the model). It looks for a minimal subset with maximal support (the number of data points that match with the model). RANSAC operates in two steps iteratively to remove outliers: hypothesize and test. First, a minimal subset is randomly selected from the data and the required model parameters are estimated based on the subset. Next, it tests the model against all other data to find inliers. The iteration stops when the probability of obtaining a model with better support than the current best model is below a given threshold. RANSAC can be expressed as calculating the minimum cost function:

$$C = \sum_i \rho(e_i) \quad (1)$$

where  $e_i$  is the error for the  $i$ th observation relative to the model,  $\rho(e)$  represents the loss function, and:

$$\rho(e) = \begin{cases} 0 & |e| < T \\ \text{constant} & |e| \geq T \end{cases} \quad (2)$$

$T$  is the correct error threshold. If  $T$  is set too high, then the robust estimate can be very poor.

MSAC [27] extends the boundaries of the loss function in RANSAC; a robust loss function is defined as below:

$$\rho_2(e) = \begin{cases} e^2 & e^2 < T^2 \\ T^2 & e^2 \geq T^2 \end{cases} \quad (3)$$

The advantage of using MSAC in point cloud data analysis has been demonstrated [28]. Choi et al. [29] evaluated the RANSAC family and showed that MSAC is one of the most accurate methods.

## 2.2. GTM and WGTM

GTM assumes two sets of match correspondences between two images:  $P = \{p_i\}$  and  $Q = \{q_i\}$  of size  $N$  where  $p_i$  matches  $q_i$ . The median KNN graph of the two images are  $G_p = \{V_p, E_p\}$  and  $G_q = \{V_q, E_q\}$ , respectively.  $G_p = \{V_p, E_p\}$  is created by the following steps: first, define vertex  $v_i$  for each match correspondence  $p_i$ , such that  $V_p = \{v_1, v_2, \dots, v_N\}$ ; second, a non-directed edge between  $v_i$  and  $v_j$  exists when  $p_j$  is one of the  $K$ -nearest neighbors of  $p_i$ , and also,  $\|p_i - p_j\| \leq \eta$ . Here,  $\eta$  is defined by the following expression:

$$\eta = \underset{(l,m) \in V_p \times V_p}{\text{median}} \|p_l - p_m\| \quad (4)$$

The adjacency matrix corresponding to  $G_p$  is denoted with  $A_p$ :

$$A_p(i, j) = \begin{cases} 1 & (i, j) \in E_p \\ 0 & (i, j) \notin E_p \end{cases} \quad (5)$$

Similarly, the graph  $G_q$  and the adjacency matrix  $A_q$  are generated in the same way. If all correspondences were correct, the two graphs would be isomorphic. The residual adjacency matrix  $R = |A_p - A_q|$  is used to calculate the similarity of two graphs.

GTM completes the detection of mismatched points through iteration. Firstly, an outlier is selected from the matching set by the following expression:

$$j^{out} = \underset{j=1,2,\dots,N}{\text{argmax}} \sum_{i=1}^N R(i, j) \quad (6)$$

Secondly, the outlier and its correspondences are removed, and  $N$  is decreased by one; thirdly, both KNN graphs are regenerated, and the next iteration is continued. The algorithm stops when  $R(i, j) = 0, \forall i, j$ .

Instead of relying only on the relationship between adjacent features, WGTM utilizes angular distances between the neighboring features. WGTM can be described by following steps [26].

- (1) Like GTM, WGTM also generates two median KNN graphs  $G_p = \{V_p, E_p\}$  (with adjacency matrix  $A_p$ ) and  $G_q = \{V_q, E_q\}$  (with adjacency matrix  $A_q$ ) for the sets  $P = \{p_i\}$  and  $Q = \{q_i\}$ . The difference is that the two graphs of  $G_p$  and  $G_q$  are directed graphs. A directed edge  $(i, j)$  exists when  $p_j$  is one of the  $K$ -nearest neighbors of  $p_i$ , and also,  $\|p_i - p_j\| \leq \eta$ ,  $\eta$  is defined by (4). The adjacency matrix is defined by (5).
- (2) Find all vertices of  $G_p$  with at most one edge with other vertices, and remove them and their correspondences from  $G_p$  and  $G_q$ . Regenerate  $G_p$  and  $G_q$ . Repeat this process until all vertices of  $G_p$  have a minimum of two edges.
- (3) For the edge that connects  $v_i$  to  $v_m$  compute a weight value using the following equation:

$$W(i, m) = \left| \arccos \left( \frac{(p_m - p_i) \cdot (\text{Rot}(\theta(k_{\min}, i))(q_m - q_i))}{\|p_m - p_i\| \|q_m - q_i\|} \right) \right| \quad (7)$$

where:

$$\text{Rot}(\theta(k_{\min}, i)) = \begin{bmatrix} \cos(\theta(k_{\min}, i)) & -\sin(\theta(k_{\min}, i)) \\ \sin(\theta(k_{\min}, i)) & \cos(\theta(k_{\min}, i)) \end{bmatrix} \quad (8)$$

Here,  $k_{\min}$  represents the optimal rotation angle between each pair of matches.  $p_i$  and  $p_m$  are the 2D vectors of the image coordinates for vertices  $v_i$  and  $v_m$ .

- (4) For each vertex  $v_i$  in  $V_p$ , find the percentage of edges that are connected to  $v_i$  with their correspondences connected to  $v'_i$  in  $V_q$ . The weight value is set to be  $\pi$  if the percentage is smaller than 50%.
- (5) For each vertex  $v_i$  in  $V_p$ , compute the mean of all weights by (9). Remove the vertex corresponding to the maximum value of  $w$  and all of its corresponding vertices from  $P$  and  $Q$ :

$$w(i) = \text{mean}_{\forall j, (i,j) \in E_p} (W(i, j)) \quad (9)$$

- (6) If the maximum value of  $w$  is less than  $\pi$ , and the change in the mean value of  $w(i)$  is less than the threshold, the iteration is terminated; otherwise, it goes to the next iteration.

### 3. The INS Error Propagation Model

An INS is available in this paper; hence, an INS error propagation model [30] is employed in this paper. The INS error propagation model in [30] is modified in this paper, since the n-reference frame adopted in underwater vehicles (north, east and down reference frames) is different from the n-reference frame adopted in [30] (east, north, and up reference frames).

Let  $\alpha_z, \alpha_y, \alpha_x$  be the three rotational angles from n-reference frame (ideal mathematics platform) to the n'-reference frame (actual mathematics platform),  $\alpha = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z \end{bmatrix}$ ; the rotation matrixes can be expressed as:

$$\begin{aligned} C_{\alpha_z} &= \begin{bmatrix} \cos \alpha_z & -\sin \alpha_z & 0 \\ \sin \alpha_z & \cos \alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ C_{\alpha_y} &= \begin{bmatrix} \cos \alpha_y & 0 & -\sin \alpha_y \\ 0 & 1 & 0 \\ \sin \alpha_y & 0 & \cos \alpha_y \end{bmatrix} \\ C_{\alpha_x} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_x & \sin \alpha_x \\ 0 & -\sin \alpha_x & \cos \alpha_x \end{bmatrix} \end{aligned} \quad (10)$$

Then, the coordinate transformation matrix from the n-reference frame to the n'-reference frame is derived:

$$1C_n^{n'} = C_{\alpha_x} C_{\alpha_y} C_{\alpha_z} \quad (11)$$

Denoting with  $\omega_{nn'}^{n'}$ , the relative angular velocity of the n'-reference frame in the n-reference frame:

$$\omega_{nn'}^{n'} = C_{\alpha_x} C_{\alpha_y} \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha}_z \end{bmatrix} + C_{\alpha_x} \begin{bmatrix} 0 \\ \dot{\alpha}_y \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\alpha}_x \\ 0 \\ 0 \end{bmatrix} = C_{\omega} \begin{bmatrix} \dot{\alpha}_x \\ \dot{\alpha}_y \\ \dot{\alpha}_z \end{bmatrix} \quad (12)$$

Hence, the differential equation of Euler platform error angles can be expressed as follows:

$$\dot{\alpha} = C_{\omega}^{-1} \omega_{nn'}^{n'} \quad (13)$$

$$\begin{aligned} C_{\omega} &= \begin{bmatrix} 1 & 0 & -\sin \alpha_y \\ 0 & \cos \alpha_x & \sin \alpha_x \cos \alpha_y \\ 0 & -\sin \alpha_x & \cos \alpha_x \cos \alpha_y \end{bmatrix} \\ C_{\omega}^{-1} &= \frac{1}{\cos \alpha_y} \begin{bmatrix} \cos \alpha_y & \sin \alpha_x \sin \alpha_y & \cos \alpha_x \sin \alpha_y \\ 0 & \cos \alpha_x \cos \alpha_y & -\sin \alpha_x \cos \alpha_y \\ 0 & \sin \alpha_x & \cos \alpha_x \end{bmatrix} \end{aligned} \quad (14)$$

Denoting with  $\omega_{ie}^n$  the rotational angular velocity of the earth,  $L$  and  $h$  are the latitude and depth, respectively.  $R_M$  is the local radius of the curvature in the meridian, and  $R_N$  is the local radius of the curvature in the prime vertical.  $\hat{L} = L + \delta L$ ,  $\hat{h} = h + \delta h$ .  $\delta L$  and  $\delta h$  are both slight errors. The following equations have been derived:

$$\omega_{ie}^n = \begin{bmatrix} \omega_{ie} \cos L & 0 & -\omega_{ie} \sin L \end{bmatrix}^T \quad (15)$$

$$\omega_{en}^n = \begin{bmatrix} \frac{v_E^n}{R_N - h} & -\frac{v_N^n}{R_M - h} & -\frac{v_E^n}{R_N - h} \tan L \end{bmatrix}^T \quad (16)$$

$$\delta \omega_{ie}^n = \begin{bmatrix} -\omega_{ie} \sin \hat{L} \delta L \\ 0 \\ -\omega_{ie} \cos \hat{L} \delta L \end{bmatrix} \quad (17)$$

$$\delta \omega_{en}^n = \begin{bmatrix} \delta v_E^n / (\hat{R}_N - \hat{h}) \\ -\delta v_N^n / (\hat{R}_M - \hat{h}) \\ -(\tan \hat{L} \delta v_E^n + \hat{v}_E^n \sec^2 \hat{L} \delta L) / (\hat{R}_N - \hat{h}) \end{bmatrix} \quad (18)$$

On the basis of formulas above, the INS error propagation model that is used in the underwater environment have been derived [30]:

$$\begin{cases} \delta \dot{L} = \frac{\delta v_N^n}{R_M - h} + \delta h \frac{v_N^n}{(R_M - h)^2} \\ \delta \dot{\lambda} = \frac{\delta v_E^n}{R_N - h} \sec L + \delta L \frac{v_E^n}{R_N - h} \tan L \sec L + \delta h \frac{v_E^n \sec L}{(R_N - h)^2} \\ \delta \dot{h} = \delta V_D \\ \dot{\alpha} = C_\omega^{-1} [(I - C_n') \hat{\omega}_{in}^n + C_n' \delta \omega_{in}^n - C_b' \varepsilon^b] - C_\omega^{-1} C_b' w_g^b \\ \delta \dot{v}^n = [I - (C_n')^T] C_b' f_{sf}^b + (C_n')^T C_b' \nabla^b - (2\delta \omega_{ie}^n + \delta \omega_{en}^n) \times (\hat{v}^n - \delta v^n) \\ \quad - (2\hat{\omega}_{ie}^n + \hat{\omega}_{en}^n) \times \delta v^n + (C_n')^T C_b' w_a^b \\ \dot{\varepsilon}^b = 0 \\ \dot{\nabla}^b = 0 \end{cases} \quad (19)$$

where  $\varepsilon^b$  and  $w_g^b$  are the constant bias and zero mean Gaussian white noise of the gyroscope, respectively, and  $\nabla^b$  and  $w_a^b$  are the constant bias and zero mean Gaussian white noise of the accelerometer, respectively.

#### 4. The Real-Time Triple Constraint Mismatch Detection Method

In DBRN, the matching process involves a digital base map. The resolution of the base map and the matching features greatly affect the matching accuracy. In the case of a base map with a large resolution, or in the case of the smooth matching features, mismatch has a high probability of occurrence. Therefore, it is necessary to develop a mismatch detection method for matching algorithms-based DBRN systems.

As the primary navigation system of AUV, INS has the characteristics of short-term high precision. INS can give continuous accurate positions to begin with, but the accumulated position errors are unacceptable after a long period underway. The position error of INS increases slowly with time, but it does not increase suddenly and rapidly in a short period, which means that in a period of time, the relative positions between the INS-indicated positions and the corresponding actual positions can be regarded as being accurate and reliable. The shape of the INS-indicated trajectory is supposed to be similar to the corresponding actual vehicle trajectory [31,32]. Therefore, an INS-indicated trajectory and the corresponding matched trajectory can be used as inputs for WGTM to perform mismatch detection. In addition, considering that AUVs are able to maintain a direct (linear) trajectory during

navigation underwater [5], the matched points can be fitted by using a linear model. It is thus possible to detect mismatched points using MSAC. However, the direct use of WGTM and MSAC on AUV for mismatch detection has the following drawbacks:

- (1) MSAC and WGTM have poor real-time performance. These two algorithms eliminate all of the outliers by iteration. Conversely, in a navigation system, the most important thing is to judge whether the current position is mismatched.
- (2) MSAC cannot detect the mismatched point that fits the model, and WGTM cannot detect the mismatched point with the same structure.
- (3) Both MSAC and WGTM do not take into account the distance constraint between the sampling points.

In view of the limitations of MSAC and WGTM applied on AUV, this paper proposes a real-time triple constraint mismatch detection method. The real-time triple constraint mismatch detection method is divided into three modules: the model fitting detection module, the spatial structure detection module, and the distance ratio detection module. The proposed method changes multi-point simultaneous detection into real-time detection of the current point. Due to the short-term high-precision of INS, the INS-indicated trajectory can be approximated as the actual trajectory during the initial period of time (assuming that the constant drift and random drift of gyro are both  $0.02^\circ/\text{hr}$ , the accelerometer's constant bias and random bias are both  $100\ \mu\text{g}$ , and then the position error of INS after 10 min of navigation is about  $0.2'$ , which is a resolution that most of the digital base map cannot reach). This feature will be used to select the input trajectory. In this paper, MSAC and WGTM are simplified to make it suitable for AUV real-time mismatch detection. In addition, the distance ratio constraint between the INS-indicated trajectory and the corresponding matched trajectory is established according to the INS error propagation model. The flow of the proposed method is shown in Figure 1.

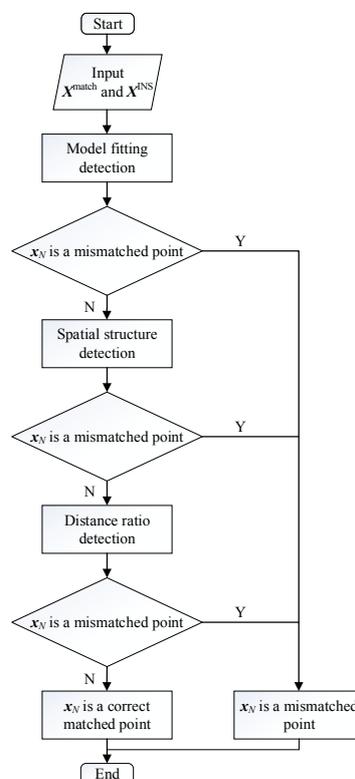


Figure 1. The flow of the proposed mismatch detection method.

#### 4.1. Model Fitting Detection

Based on MSAC, the proposed detection method detects the fitting degree of the current matching result with the optimal linear model. In the model fitting detection module,  $\mathbf{X}^{\text{match}} = \{x_1^{\text{match}}, x_2^{\text{match}}, \dots, x_N^{\text{match}}\}$  are the input point sets. Here  $x_N^{\text{match}}$  is the current point (the output of matching algorithm at the current moment), and  $x_i^{\text{match}}$  ( $i = \{1, 2, \dots, N-1\}$ ) are  $N-1$  correct matched points closest to  $x_N^{\text{match}}$ . If there are fewer than  $N-1$  points successfully matched before  $x_N^{\text{match}}$ , the remaining points are supplemented by the INS-indicated position that can be considered the actual position. As mentioned before, considering the short-term high-precision characteristics of INS, as long as the accumulated position error calculated by (19) is smaller than one grid, the INS-indicated position can be regarded as the actual position. Define  $e_{x_i^{\text{match}}}$  as the Euclidean distance between  $x_i^{\text{match}}$  and the current linear model  $M$ . Functions (1) and (3) are used to generate the value of cost function  $C_M$ .  $M$  with the largest number of inliers, and the smallest value of  $C_M$  is the best linear model  $M_{\text{best}}$ . In practical experiments, the correct error threshold  $T$  is set to three grids (found empirically). Finally, the error  $e_{x_i^{\text{match}}}$  is calculated for each point in  $\mathbf{X}^{\text{match}}$  relative to  $M_{\text{best}}$ . If  $x_N^{\text{match}}$  is an outlier and  $e_{x_N^{\text{match}}}$  is the largest error among all points in  $\mathbf{X}^{\text{match}}$ , then  $x_N^{\text{match}}$  is a mismatched point. The pseudo-code of model fitting detection is shown in Algorithm 1.

---

#### Algorithm 1. Model fitting detection

---

##### Begin

1. select input point set  $\mathbf{X}^{\text{match}}$
2. initialize  $N_{M_{\text{best}}}^{\text{inliers}} = 0$
3. for  $I = 1$  to  $N-1$
4.   for  $j = 2$  to  $N$ 
  - calculate the parameters of the linear model  $M$  according to  $x_i^{\text{match}}$  and  $x_j^{\text{match}}$
  - compute  $N_M^{\text{inliers}}$  (the number of inliers) in  $\mathbf{X}^{\text{match}}$  based on  $M$  and  $\rho_2$
  - calculate the cost function  $C_M$
  - if  $N_M^{\text{inliers}} > N_{M_{\text{best}}}^{\text{inliers}}$  or ( $N_M^{\text{inliers}} = N_{M_{\text{best}}}^{\text{inliers}}$  and  $C_M < C_{M_{\text{best}}}$ )
  - $M_{\text{best}} = M$
  - end if
5. if  $x_N^{\text{match}}$  is an outlier and  $e_{x_N^{\text{match}}}^{\text{max}} = e_{x_N^{\text{match}}}$ ,  $i = \{1, 2, \dots, N\}$ ,  $x_N^{\text{match}}$  is a mismatched point

##### End

---

#### 4.2. Spatial Structure Detection

The shape of the INS-indicated trajectory is supposed to be similar to the actual vehicle trajectory. In other words, the INS-indicated trajectory has a similar spatial structure to the corresponding actual trajectory. In the spatial structure detection module, the WGTm algorithm is simplified to make it more suitable for handling mismatch detection problems in matching algorithms-based DBRN systems.

In image processing, some of the points go through dramatic changes, and they therefore do not retain the invariance property with respect to rotation or scale under large perspective transformation. This problem does not exist in the mismatch detection problem in navigation systems, because the rotation angle between the INS-indicated trajectory and the corresponding actual trajectory is small. Therefore, the second step in WGTm is removed. In addition, the proposed method simplifies the computation by using the rotation angle between the INS-indicated trajectory and the corresponding matched trajectory as the optimal rotation angle. Let  $\mathbf{X}^{\text{INS}} = \{x_1^{\text{INS}}, x_2^{\text{INS}}, \dots, x_N^{\text{INS}}\}$

be the INS-indicated trajectory, and let  $\mathbf{X}^{\text{match}} = \{\mathbf{x}_1^{\text{match}}, \mathbf{x}_2^{\text{match}}, \dots, \mathbf{x}_N^{\text{match}}\}$  be the corresponding matched trajectory. Denoting with  $\mathbf{Q}$  the covariance matrix:

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \sum_{k=1}^N \left( \mathbf{x}_k^{\text{INS}} - \tilde{\mathbf{x}}^{\text{INS}} \right) \left( \mathbf{x}_k^{\text{match}} - \tilde{\mathbf{x}}^{\text{match}} \right)^T \quad (20)$$

where  $\tilde{\mathbf{x}}^{\text{INS}}$  and  $\tilde{\mathbf{x}}^{\text{match}}$  are, respectively, the average values of all position vectors in  $\mathbf{X}^{\text{INS}}$  and  $\mathbf{X}^{\text{match}}$ . The eigenvalues of  $\mathbf{Q}$  and the rotation angle from  $\mathbf{X}^{\text{INS}}$  to  $\mathbf{X}^{\text{match}}$  are calculated through a quaternion algorithm [33]:

$$\lambda_{1,2} = \pm \left[ (Q_{11} + Q_{22})^2 + (Q_{21} - Q_{12})^2 \right]^{1/2} \quad (21)$$

$$\lambda_{3,4} = \pm \left[ (Q_{11} - Q_{22})^2 + (Q_{21} + Q_{12})^2 \right]^{1/2}$$

$$\text{tg}\left(\frac{\tau}{2}\right) = (S_{11} + S_{22} - \lambda_m) / (S_{12} - S_{21}) \quad (22)$$

where  $\lambda_\mu (\mu = \{1, 2, 3, 4\})$  are four eigenvalues of  $\mathbf{Q}$ , and  $\lambda_m$  is the maximum eigenvalue.  $\tau$  is the rotation angle from  $\mathbf{X}^{\text{INS}}$  to  $\mathbf{X}^{\text{match}}$ . The spatial structure detection module can be described by the following steps.

- (1) Two median KNN graphs  $\mathbf{G}^{\text{INS}} = \{\mathbf{V}^{\text{INS}}, \mathbf{E}^{\text{INS}}\}$  (with the adjacency matrix  $\mathbf{A}^{\text{INS}}$ ) and  $\mathbf{G}^{\text{match}} = \{\mathbf{V}^{\text{match}}, \mathbf{E}^{\text{match}}\}$  (with adjacency matrix  $\mathbf{A}^{\text{match}}$ ) are generated for the trajectories  $\mathbf{X}^{\text{INS}} = \{\mathbf{x}_1^{\text{INS}}, \mathbf{x}_2^{\text{INS}}, \dots, \mathbf{x}_N^{\text{INS}}\}$  and  $\mathbf{X}^{\text{match}} = \{\mathbf{x}_1^{\text{match}}, \mathbf{x}_2^{\text{match}}, \dots, \mathbf{x}_N^{\text{match}}\}$ . A directed edge  $(i, j)$  exists when  $\mathbf{x}_j$  is one of the K-nearest neighbors of  $\mathbf{x}_i$  and also  $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \eta$ ,  $\eta$  is defined by (4). The adjacency matrix is defined by (5).
- (2) For the edge that connects  $v_i^{\text{match}}$  to  $v_m^{\text{match}}$  compute a weight value using the following equation:

$$W(i, m) = \left| \arccos \left( \frac{(\mathbf{x}_m^{\text{INS}} - \mathbf{x}_i^{\text{INS}}) \cdot (\mathbf{R}_\tau (\mathbf{x}_m^{\text{match}} - \mathbf{x}_i^{\text{match}}))}{\|\mathbf{x}_m^{\text{INS}} - \mathbf{x}_i^{\text{INS}}\| \|\mathbf{x}_m^{\text{match}} - \mathbf{x}_i^{\text{match}}\|} \right) \right| \quad (23)$$

where:

$$\mathbf{R}_\tau = \begin{bmatrix} \cos \tau & -\sin \tau \\ \sin \tau & \cos \tau \end{bmatrix} \quad (24)$$

Here,  $\tau$  represents the optimal rotation angle calculated by (22).  $\mathbf{R}_\tau$  is the corresponding rotation matrix.

- (3) For vertex  $v_N^{\text{match}}$ , find the percentage of edges that are connected to  $v_N^{\text{match}}$ , with their correspondences connected to  $v_N^{\text{INS}}$ . If the percentage is smaller than 50%, the weight value of all different edges should be replaced by  $\pi$ .
- (4) For each vertex  $v_i^{\text{match}}$ , compute the mean of all weights by the following expression:

$$w(i) = \text{mean}_{\forall j, (i,j) \in E^{\text{match}}} (W(i, j)) \quad (25)$$

- (5) If the maximum value of  $w$  is not  $w(N)$ ,  $\mathbf{x}_N$  pass the spatial structure detection; otherwise, set:

$$\mu_{\text{old}} = \text{mean}_{\forall i} (w(i)) \quad (26)$$

and let  $\mathbf{X}^{\text{INS}} = \{\mathbf{x}_1^{\text{INS}}, \mathbf{x}_2^{\text{INS}}, \dots, \mathbf{x}_{N-1}^{\text{INS}}\}$  and  $\mathbf{X}^{\text{match}} = \{\mathbf{x}_1^{\text{match}}, \mathbf{x}_2^{\text{match}}, \dots, \mathbf{x}_{N-1}^{\text{match}}\}$ . Repeat step (1) to step (4), and  $\mu_{\text{new}}$  is calculated by (26). If  $|\mu_{\text{new}} - \mu_{\text{old}}| < \varepsilon$ ,  $\mathbf{x}_N$  passes the spatial structure detection; otherwise,  $\mathbf{x}_N$  is a mismatched point. The value of  $\varepsilon$  is set to 0.01 (found empirically)

for all of the results presented in this work. The pseudo-code of the spatial structure detection is shown in Algorithm 2.

---

**Algorithm 2.** Spatial Structure Detection

---

**Begin:**

1. select input point sets  $\mathbf{X}^{\text{INS}} = \{x_i^{\text{INS}}\}$  and  $\mathbf{X}^{\text{match}} = \{x_i^{\text{match}}\}, i = \{1, 2, \dots, N\}$
2. create  $\mathbf{G}^{\text{INS}}, \mathbf{G}^{\text{match}}, \mathbf{A}^{\text{INS}}, \mathbf{A}^{\text{match}}$
3. compute  $W$
4.  $\forall (i, j) \in \mathbf{E}^{\text{match}},$  compute  $w(i) = \text{mean}(W(i, j))$
5. if  $\max(w(i)) \neq w(N)$ 
  - $x_N$  pass the spatial structure detection
  - else
    - $\mu_{\text{old}} = \text{mean}_{\forall i}(w(i))$
    - delete  $x_N$  and regenerate  $\mathbf{X}^{\text{INS}} = \{x_i^{\text{INS}}\}, \mathbf{X}^{\text{match}} = \{x_i^{\text{match}}\}, i = \{1, 2, \dots, N - 1\}$
    - repeat steps 1 to 4, calculate  $\mu_{\text{new}}$
    - if  $|\mu_{\text{new}} - \mu_{\text{old}}| < \varepsilon$ 
      - $x_N$  pass the spatial structure detection
      - else
        - $x_N$  is a mismatched point
        - end if
      - end if

**End**

---

#### 4.3. Distance Ratio Detection

There is a scaling error between the INS-indicated trajectory and the corresponding actual trajectory [34]. To better describe the scaling error, let  $\mathbf{X}^{\text{INS}} = \{x_1^{\text{INS}}, x_2^{\text{INS}}, \dots, x_N^{\text{INS}}\}$  be the INS-indicated trajectory and let  $\mathbf{X}^{\text{actual}} = \{x_1^{\text{actual}}, x_2^{\text{actual}}, \dots, x_N^{\text{actual}}\}$  be the corresponding actual trajectory. The INS-indicated distance and the actual distance in a short period are expressed by the following equations:

$$d_{k-1,k}^{\text{INS}} = \frac{\|v_{k-1}^{\text{INS}} + v_k^{\text{INS}}\|}{2} \cdot \Delta t_{k-1,k} \quad (27)$$

$$d_{k-1,k}^{\text{actual}} \approx \frac{\|v_{k-1}^{\text{actual}} + v_k^{\text{actual}}\|}{2} \cdot \Delta t_{k-1,k} \quad (28)$$

$d_{k-1,k}^{\text{INS}}$  and  $d_{k-1,k}^{\text{actual}}$  are, respectively, the INS-indicated distance and the actual distance between  $x_{k-1}$  and  $x_k$ , respectively;  $k \in \{2, 3, \dots, N\}$ .  $v_k^{\text{INS}}$  and  $v_k^{\text{actual}}$  are the INS-indicated linear velocity and the actual linear velocity, respectively, of  $x_k$ .  $\Delta t_{k-1,k}$  is the time interval between  $x_{k-1}$  and  $x_k$ . Then, the relationship between  $d_{k-1,k}^{\text{INS}}$  and  $d_{k-1,k}^{\text{actual}}$  is derived:

$$\frac{d_{k-1,k}^{\text{actual}}}{d_{k-1,k}^{\text{INS}}} = \frac{\|v_{k-1}^{\text{actual}} + v_k^{\text{actual}}\|}{\|v_{k-1}^{\text{INS}} + v_k^{\text{INS}}\|} \quad (29)$$

where  $v_k^{\text{INS}}$  is described by an equation in the form:

$$v_k^{\text{INS}} = v_k^{\text{actual}} + \delta v_k \quad (30)$$

Here  $\delta v_k$  is the INS-accumulated velocity error of  $x_k$ . Accordingly, (29) is developed into the following form:

$$\frac{d_{k-1,k}^{\text{actual}}}{d_{k-1,k}^{\text{INS}}} = \frac{\|v_{k-1}^{\text{INS}} + v_k^{\text{INS}} - \delta v_{k-1} - \delta v_k\|}{\|v_{k-1}^{\text{INS}} + v_k^{\text{INS}}\|} \quad (31)$$

Through (19) and (31), the range of  $\frac{d_{k-1,k}^{\text{actual}}}{d_{k-1,k}^{\text{INS}}}$  during the entire sailing period can be estimated. Define  $S_{\min}$  as the lower bound of the range and  $S_{\max}$  as the upper bound of the range. The range can be expressed by the following expression:

$$\frac{d_{k-1,k}^{\text{actual}}}{d_{k-1,k}^{\text{INS}}} \in (S_{\min}, S_{\max}) \quad (32)$$

Based on the premise described above, for a matched trajectory  $X^{\text{match}}$ , if  $X^{\text{match}}$  is a correctly matched trajectory,  $\frac{d_{k-1,k}^{\text{match}}}{d_{k-1,k}^{\text{INS}}}$  should meet the range described in (32),  $k = \{1, 2, \dots, N\}$ . Here  $d_{k-1,k}^{\text{match}}$  is the distance between  $x_{k-1}^{\text{match}}$  and  $x_k^{\text{match}}$ . The pseudo-code of the distance ratio detection is shown in Algorithm 3.

---

**Algorithm 3.** Distance ratio detection

---

**Begin:**

1. select input point sets  $X^{\text{INS}} = \{x_i^{\text{INS}}\}$  and  $X^{\text{match}} = \{x_i^{\text{match}}\}, i = \{1, 2, \dots, N\}$
2. compute  $\frac{d_{N-1,N}^{\text{match}}}{d_{N-1,N}^{\text{INS}}}$
3. if  $S_{\min} \leq \frac{d_{N-1,N}^{\text{match}}}{d_{N-1,N}^{\text{INS}}} \leq S_{\max}$   
 $x_N$  passes the distance ratio detection  
else  
 $x_N$  is a mismatched point  
end if

**End**

---

## 5. Simulation and Analysis

In order to test the feasibility of our approach, numerous simulation experiments have been performed. Firstly, an INS-indicated trajectory was generated. Secondly, the vector iterated closest contour point (VICCP) algorithm [34] was conducted to obtain a matched trajectory. Thirdly, the proposed mismatch detection method was performed. Fourthly, we used the RSOC-based mismatch diagnostic [9] algorithm as a reference, comparing the results of the RSOC-based mismatch diagnostic algorithm with the proposed algorithm. Finally, we analyzed the impact of thresholds on the proposed method.

### 5.1. Simulation of INS

An INS-indicated trajectory starting from (156.4°E, 19.7°N) was generated, and the INS-relevant parameters are listed in Table 1. The INS position error is shown in Figure 2. As seen from Figure 2, the position error of the generated trajectory exhibited a Schuler oscillation of 84.4 min.

**Table 1.** Simulation conditions of the inertial navigation system (INS).

Parameters	Quantity	Unit
Gyro constant drift	0.02	°/hr
Gyro random drift ( $1\sigma$ )	0.02	°/hr
Accelerometer constant bias	100	$\mu\text{g}$
Accelerometer random bias ( $1\sigma$ )	100	$\mu\text{g}$
Velocity	7.71	m/s
Acceleration	0	m/s
Initial angle error	0	°
Azimuth angle	60	°
Initial longitude error	0.1	'
Initial latitude error	0.1	'
Simulation time	10	hr

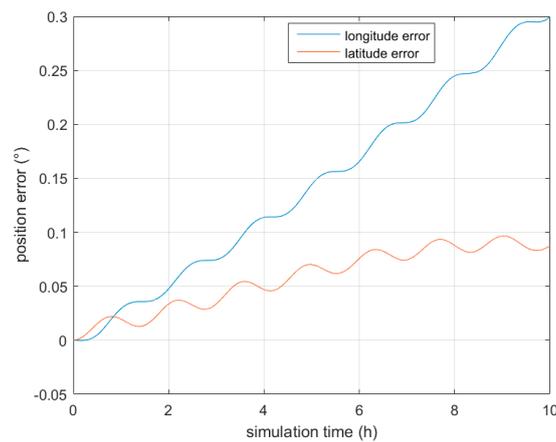


Figure 2. The INS position error.

### 5.2. Simulation of the Proposed Algorithm

To perform the mismatch detection method presented in this work, in addition to the INS-indicated trajectory, a matched trajectory is required. In the simulation experiment, we used the VICCP algorithm as the matching algorithm. The EMAG2 model [35] was used to calculate the geomagnetic anomaly in the area from (152.4°E, 19.5°N) to (156.6°E, 23.7°N). After interpolation, the grid step was converted to 0.3'. The 3D maps of the geomagnetic anomaly data are shown in Figure 3. The geomagnetic anomaly-relevant parameters are shown in Table 2.

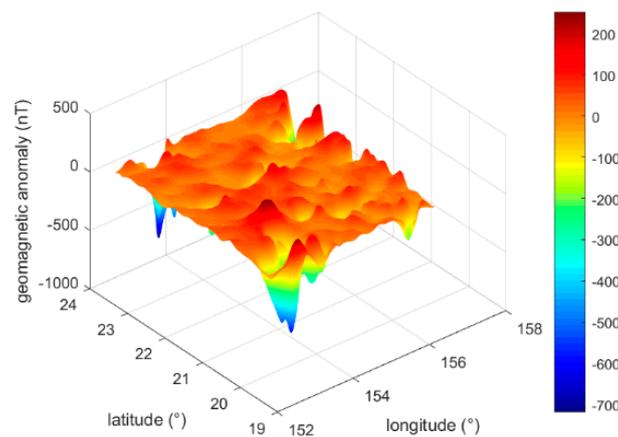


Figure 3. 3D map of the geomagnetic anomaly data.

Table 2. Parameters of the geomagnetic base map.

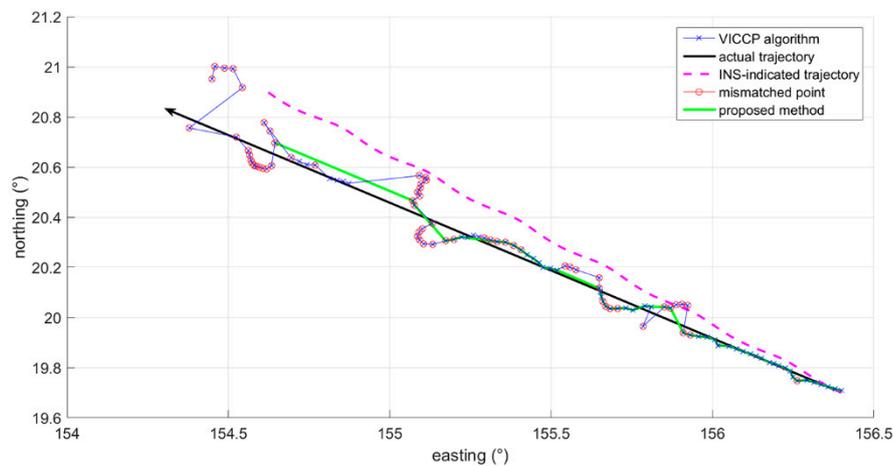
Parameters	Quantity	Unit
Number of grid points	840 × 840	points
Grid step	0.3	'
Minimum value	−719.21	nT
Maximum value	253.44	nT
Mean	−2.46	nT

Table 3 shows the configuration of the VICCP algorithm. After the INS-indicated trajectory and the corresponding matched trajectory calculated by VICCP are gained, their position information is delivered to the proposed method.  $T$  and  $\varepsilon$  in the proposed method are set to 3 grid and 0.01, respectively. Figure 4 shows the INS-indicated trajectory, the matched trajectory calculated by VICCP, and the trajectory after performing the proposed mismatch detection method within nine hours. In

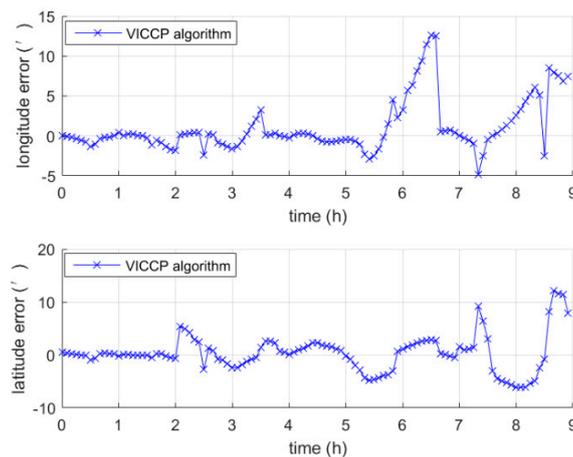
Figure 4, the red cycle in VICCP trajectory represents the mismatched point, which is defined as the point where the position error is greater than five grids. The black arrow in Figure 4 indicates the direction of navigation. Obviously, compared to the VICCP trajectory in Figure 4, the matched trajectory obtained by the proposed method was significantly closer to the actual trajectory. Figure 5 indicates the longitude error and latitude error of each matching point. At the beginning, the VICCP algorithm could hold a highly successful matching rate. However, with the initial position error increase, the successful matching rate decreased significantly.

**Table 3.** Configuration of the vector iterated closest contour point (VICCP) algorithm.

Parameters	Parameter Values
Maximum number of iterations	500
Number of sampling points per sequence	13
Sampling interval	5 min
The variance of the magnetic anomaly measurement noise	1 nT



**Figure 4.** Matching results within nine hours.



**Figure 5.** Longitude error and latitude error within nine hours.

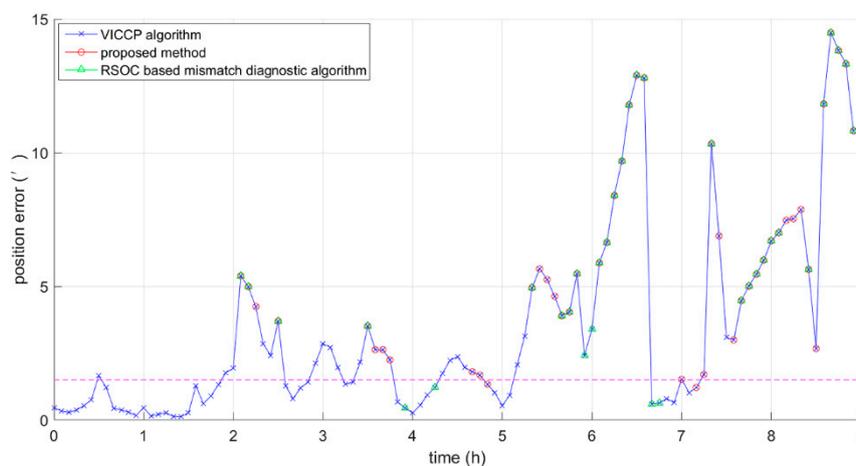
### 5.3. Comparison between the RSOC-Based Mismatch Diagnostic Algorithm and the Proposed Method

In the comparative experiment of the RSOC-based mismatch diagnostic algorithm and the proposed method, we used the RSOC-based mismatch diagnostic algorithm and the proposed method to detect the same trajectory.  $a$  and  $b$  in the RSOC based mismatch diagnostic algorithm were set to 1 grid and 0.5 grid, respectively. Figure 6 shows the results of the two detection methods, the green circles

represent the detected results obtained by the RSOC-based mismatch diagnostic algorithm, and the red circles represent the detected results obtained by the proposed method. In Figure 6, the pink dotted line is the boundary between the correct match and the mismatch, the mismatched points are above the dotted line, and correct matched points are below the dotted line. In order to better evaluate the proposed method, the statistical results of the two methods are given in Table 4. In Table 4, a matching result is considered to be a successful match if the position error is within 5 grids (1.5'). The results of the two mismatch detection methods include correct detection (points with a position error of greater than 5 grids) and error detection (points with a position error of less than 5 grids). The proportion of the correct detection in the detection results can measure the performance of a mismatch detection algorithm. In addition, the detection results do not necessarily include all of the mismatched points on the trajectory. The proportion of the correct detection results in all mismatched points was also used to measure the performance of the algorithm. Column *CD* (Correct Detection) represents the number of points in the detection results where the position error is greater than five grids. The column *TN* (total number) stands for the total number of points obtained by the corresponding detection algorithm. *TN'* is defined as the total number of the points with position errors of greater than five grids on the trajectory. *CR* (correctness rate) and *DR* (detection rate) are defined by the following expressions:

$$CR = \frac{CD}{TN} \quad (33)$$

$$DR = \frac{CD}{TN'} \quad (34)$$



**Figure 6.** Mismatch detection results of the restricted spatial order constraints (RSOC)-based mismatch diagnostic algorithm and the proposed method.

**Table 4.** Statistic results of the RSOC-based mismatch diagnostic algorithm and the proposed method.

Detection Method	<i>CD</i>	<i>TN</i>	<i>CR</i>	<i>DR</i>
RSOC-based mismatch diagnostic algorithm	30	34	88.24%	46.88%
Proposed algorithm	45	54	83.33%	70.31%

It can be concluded from Figure 6 and Table 4 that the proposed method can detect more mismatched points than the RSOC-based mismatch diagnostic algorithm. Moreover, since the proposed method can perform mismatch detection point-by-point, the proposed method can perform a mismatch detection every time the current location is updated. However, the RSOC-based mismatch diagnostic algorithm performs a mismatch detection sequence by sequence, and it cannot perform mismatch

detection immediately when the location is updated. Therefore, the proposed method can detect the matching result in real time.

#### 5.4. Influence of the Threshold Value

In the proposed method,  $T$  and  $\varepsilon$  are two threshold values that need to be set. The impact on the detection results of the threshold value is discussed in this part. The statistic results obtained by the different thresholds are shown in Table 5.

**Table 5.** Threshold value settings and results.

Threshold Value		CD	TN	CR	DR
$T$	$\varepsilon$				
3 grid	0.001	59	86	68.60%	92.19%
3 grid	0.1	41	47	87.23%	64.06%
0.5 grid	0.01	62	97	63.92%	96.88%
7 grid	0.01	36	41	87.90%	56.25%

It was determined that the smaller the values of  $T$  and  $\varepsilon$ , the more mismatched points are detected, and the greater the probability of false detection. The larger the values of  $T$  and  $\varepsilon$ , the smaller the role of the model-fitting detection module and the spatial structure detection module in the proposed mismatch detection method.

The settings of the threshold values depended on the resolution of the base map and the accuracy of the INS. Under the simulation conditions of this paper, when  $T$  was set to 3 grid~5 grid and  $\varepsilon$  was set to 0.01~0.05, the detection results varied slightly.

## 6. Conclusions

In this paper, a real-time mismatch detection method for DBRN is proposed. The proposed method consists of model-fitting detection module, spatial structure detection module, and a distance ratio detection module. In the model-fitting detection module, we used the navigation characteristics of AUV to select the model for fitting. In the spatial structure detection module and the distance ratio detection module, we used the short-time high-precision characteristics of INS and the error propagation model of INS to perform mismatch detection. Simulation tests are performed on a geomagnetic map with resolution of  $0.3' \times 0.3'$ . The simulation results show that the proposed method can detect the mismatched points accurately in real time. The matched trajectory gains better reliability after eliminating the mismatched points.

The proposed method can correct the output of DBRN, but it does not improve DBRN itself. The method of performing closed-loop correction on the position error of INS should be discussed in the next step.

**Author Contributions:** Conceptualization, T.D. and L.M.; methodology, T.D. and L.M.; data curation, Y.G.; validation, T.D. and Y.G., writing, T.D.

**Funding:** This work was supported by the National Natural Science Foundation of China (grant number 61473039).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Ocean. Eng.* **2013**, *39*, 131–149. [[CrossRef](#)]
2. Ramesh, R.; Jyothi, V.B.N.; Vedachalam, N.; Ramadass, G.A. Development and performance validation of a navigation system for an underwater vehicle. *J. Navig.* **2016**, *69*, 1097–1113. [[CrossRef](#)]
3. Caffaz, A.; Caiti, A.; Casalino, G.; Turetta, A. The hybrid glider/AUV folaga. *IEEE Robot. Autom. Mag.* **2010**, *17*, 31–44. [[CrossRef](#)]

4. Allotta, B.; Caiti, A.; Costanzi, R.; Fanelli, F.; Fenucci, D.; Meli, E.; Ridolfi, A. A new AUV navigation system exploiting unscented Kalman filter. *Ocean Eng.* **2016**, *113*, 121–132. [[CrossRef](#)]
5. Wynn, R.B.; Huvenne, V.A.I.; Le Bas, T.P.; Murton, B.J.; Connelly, D.P.; Bett, B.J.; Ruhl, H.A.; Morris, K.J.; Peakall, J.; Parsons, D.R.; et al. Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Mar. Geol.* **2014**, *352*, 451–468. [[CrossRef](#)]
6. Stutters, L.; Liu, H.; Tiltman, C.; Brown, D.J. Navigation technologies for autonomous underwater vehicles. *IEEE Trans. Syst. Man Cybern. Part C* **2008**, *38*, 581–589. [[CrossRef](#)]
7. Pérez-Alcocer, R.; Torres-Méndez, L.A.; Olguín-Díaz, E.; Maldonado-Ramírez, A.A. Vision-based autonomous underwater vehicle navigation in poor visibility conditions using a model-free robust control. *J. Sens.* **2016**, *2016*, 1–16. [[CrossRef](#)]
8. Canciani, A.; Raquet, J. Airborne magnetic anomaly navigation. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 67–80. [[CrossRef](#)]
9. Han, Y.; Wang, B.; Deng, Z.; Wang, S.; Fu, M. A mismatch diagnostic method for TERCOM-based underwater gravity aided navigation. *IEEE Sens. J.* **2017**, *17*, 2880–2888. [[CrossRef](#)]
10. Rice, H.; Kelmenson, S.; Mendelsohn, L. Geophysical navigation technologies and applications. In Proceedings of the IEEE Position Location and Navigation Symposium, Monterey, CA, USA, 26–29 April 2004; pp. 618–624. [[CrossRef](#)]
11. Canciani, A.; Raquet, J. Absolute positioning using the Earth’s magnetic anomaly field. *Navigation* **2016**, *63*, 111–126. [[CrossRef](#)]
12. Lee, J.; Kwon, J.H.; Yu, M. Performance evaluation and requirements assessment for gravity gradient referenced navigation. *Sensors* **2015**, *15*, 16833–16847. [[CrossRef](#)]
13. Hegrehaes, O.; Hallingstad, O. Model-aided INS with sea current estimation for robust underwater navigation. *IEEE J. Ocean. Eng.* **2011**, *36*, 316–337. [[CrossRef](#)]
14. Wu, L.; Wang, H.; Chai, H.; Zhang, L.; Hsu, H.; Wang, Y. Performance Evaluation and Analysis for Gravity Matching Aided Navigation. *Sensors* **2017**, *17*, 769. [[CrossRef](#)]
15. Kim, T.; Kim, J.; Byun, S.W. A comparison of nonlinear filter algorithms for terrain-referenced underwater navigation. *Int. J. Control Autom. Syst.* **2018**, *16*, 2977–2989. [[CrossRef](#)]
16. Simanek, J.; Reinstein, M.; Kubelka, V. Evaluation of the ekf-based estimation architectures for data fusion in mobile robots. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 985–990. [[CrossRef](#)]
17. Kim, Y.; Park, J.; Bang, H. Terrain-referenced navigation using an interferometric radar altimeter. *Navig. J. Inst. Navig.* **2018**, *65*, 157–167. [[CrossRef](#)]
18. Kim, Y.; Hong, K.; Bang, H. Utilizing out-of-sequence measurement for ambiguous update in particle filtering. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 493–501. [[CrossRef](#)]
19. Han, Y.; Wang, B.; Deng, Z.; Wang, S.; Fu, M. An improved TERCOM-based algorithm for gravity-aided navigation. *IEEE Sens. J.* **2016**, *16*, 2537–2544. [[CrossRef](#)]
20. Yang, Y.; Wang, K. Mismatching judgment using PDAF in ICCP algorithm. In Proceedings of the Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008. [[CrossRef](#)]
21. Niedfeldt, P.; Beard, R. Convergence and complexity analysis of recursive-RANSAC: A new multiple target tracking algorithm. *IEEE Trans. Autom. Control* **2016**, *61*, 456–461. [[CrossRef](#)]
22. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the weighted RANSAC approaches for building roof plane segmentation from lidar point clouds. *Remote Sens.* **2015**, *8*, 5. [[CrossRef](#)]
23. Li, Y.; Gans, N.R. Predictive ransac: Effective model fitting and tracking approach under heavy noise and outliers. *Comput. Vis. Image Underst.* **2017**, *161*, 99–113. [[CrossRef](#)]
24. Aguilar, W.; Frauel, Y.; Escolano, F.; Martinez-Perez, M.E.; Espinosa-Romero, A.; Lozano, M.A. A robust graph transformation matching for non-rigid registration. *Image Vis. Comput.* **2009**, *27*, 897–910. [[CrossRef](#)]
25. Izadi, M.; Saeedi, P. Robust weighted graph transformation matching for rigid and nonrigid image registration. *IEEE Trans. Image Process.* **2012**, *21*, 4369–4382. [[CrossRef](#)]
26. Liu, Z.; An, J.; Yu, J. A simple and robust feature point matching algorithm based on restricted spatial order constraints for aerial image registration. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 514–527. [[CrossRef](#)]
27. Torr, P.H.S.; Zisserman, A. MLESAC: A new robust estimator with application to estimating image geometry. *J. Comput. Vis. Image Underst.* **2000**, *78*, 138–156. [[CrossRef](#)]
28. Nurunnabi, A.; Belton, D.; West, G. Robust statistical approaches for local planar surface fitting in 3d laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 106–122. [[CrossRef](#)]

29. Choi, S.; Kim, T.; Yu, W. Performance evaluation of RANSAC family. *Br. Mach. Vision Conf.* **2009**. [[CrossRef](#)]
30. Yan, G.; Yan, W.; Xu, D. Application of simplified UKF in SINS initial alignment for large misalignment angles. *J. Chin. Inert. Technol.* **2008**, *16*, 253–264.
31. Wu, L.; Wang, H.; Chai, H.; Hsu, H.; Wang, Y. Research on the relative positions-constrained pattern matching method for underwater gravity-aided inertial navigation. *J. Navig.* **2015**, *68*, 937–950. [[CrossRef](#)]
32. Claus, B.; Bachmayer, R. Terrain-aided navigation for an underwater glider. *J. Field Robot.* **2015**, *32*, 935–951. [[CrossRef](#)]
33. Berthold, K.P.H. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A* **1987**, *4*, 629–642. [[CrossRef](#)]
34. Song, Z.; Zhang, J.; Zhu, W.; Xi, X. The vector matching method in geomagnetic aiding navigation. *Sensors* **2016**, *16*, 1120. [[CrossRef](#)]
35. Maus, S.; Barckhausen, U.; Berkenbosch, H.; Bournas, N.; Brozina, J.; Childers, V.; Dostaler, F.; Fairhead, J.D.; Finn, C.; von Frese, R.R.B.; et al. Emag2: A 2-arc min resolution earth magnetic anomaly grid compiled from satellite, airborne, and marine magnetic measurements. *Geochem. Geophys. Geosyst.* **2009**, *10*. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).