


Article

A Fast Algorithm for Matching AIS Trajectories with Radar Point Data in Complex Environments

Jialuo Xu, Ying Suo, Yuqing Jiang and Qiang Yang * 

School of Electronic Information Engineering, Harbin Institute of Technology, Harbin 150001, China; 23s136014@stu.hit.edu.cn (J.X.); suoying@hit.edu.cn (Y.S.); 22b905045@hit.stu.edu.cn (Y.J.)

* Correspondence: yq@hit.edu.cn; Tel.: +86-0451-86418051

Abstract: In high-traffic port areas, vessel traffic-management systems (VTMS) are essential for managing ship movements and preventing collisions. However, inaccuracies and omissions in the Automatic Identification System (AIS), along with frequent false tracks generated by radar false alarms in complex environments, can compromise VTMS stability. To address the challenges of establishing consistent navigation and improving trajectory quality, this study introduces a novel method to directly identify AIS-matched trajectories from radar plots. This approach treats radar points as probability clouds, generating a multi-dimensional information layer by stacking these clouds after differential transformations based on AIS data. The resulting layer undergoes filtering and clustering to extract point sets that align with AIS data, effectively isolating matching trajectories. The algorithm, validated with simulated data, rapidly identifies target trajectories amid extensive interference without requiring strict parameter adjustments. In measured data, the algorithm rapidly provides matching trajectories, although further human judgment is still required due to the potential absence of true values in measured data.

Keywords: clustering algorithm; AIS; track establishment; trajectory extraction



Citation: Xu, J.; Suo, Y.; Jiang, Y.; Yang, Q. A Fast Algorithm for Matching AIS Trajectories with Radar Point Data in Complex Environments. *Remote Sens.* **2024**, *16*, 4360. <https://doi.org/10.3390/rs16234360>

Academic Editors: Weimin Huang, Martin Gade, Yonggang Ji, Yiming Wang and Weifeng Sun

Received: 30 September 2024
Revised: 12 November 2024
Accepted: 19 November 2024
Published: 22 November 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, marine transportation constitutes over 90% of the global cargo transportation volume [1], playing a pivotal role in international logistics. However, coastal, port, and inland waterway areas exhibit high vessel traffic density and complex navigational scenarios [2], posing considerable challenges to safe navigation and elevating the risk of collisions [3].

Vessel traffic-management systems primarily rely on two subsystems: AIS and radar system [4]. AIS data provide high accuracy and extensive coverage; however, AIS data arrival is unpredictable and random [5,6], especially in high-density traffic areas, where extended data gaps may occur [7]. In contrast, radar-based tracking serves as a supplementary source; however, data from radar systems, including shipborne radar, is susceptible to false alarms and missed detections. This, coupled with considerable variance in radar readings, can lead to missed and incorrect matches [8].

A common approach involves processing radar and AIS data separately within distributed systems to generate high-quality tracking information [9,10]. Subsequently, these data are matched and fused, as illustrated in Figure 1.

Traditional research in this field has relied on signal-processing and data-analysis techniques, including fuzzy decision theory [11,12], maximum likelihood estimation [13], Hough transform [14,15], and Kalman filtering [16,17], to achieve stable and precise data. However, data fusion typically occurs at a later stage, without fully leveraging the distinct characteristics of both sensor types. This approach introduces unnecessary computational load during the track-building phase, and due to the inconsistency in track quality, there may be a high incidence of missed or incorrect tracks in complex environments.

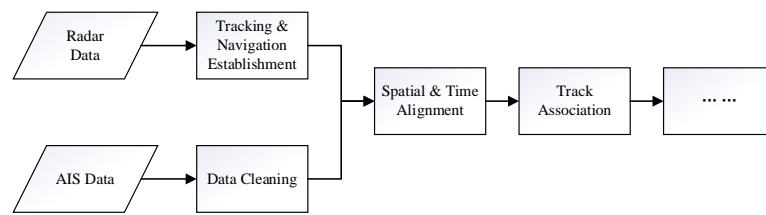


Figure 1. Traditional track-matching process.

This paper presents a novel approach that enables direct matching of AIS data and radar plot data in complex situations. The specific process is illustrated in Figure 2:

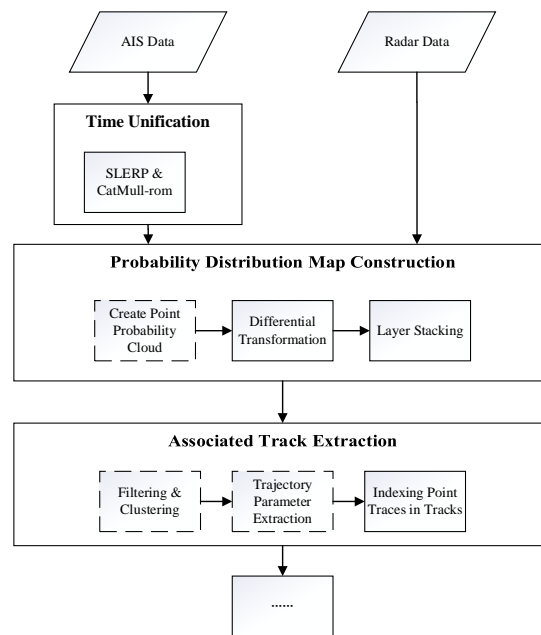


Figure 2. Algorithm processing flow.

In Section 2, each step of the process illustrated in Figure 2 is described in detail. Due to practical application requirements, the step of “create radar point probability clouds” has been moved up to facilitate the rapid extraction of parameters for different AIS data based on various indices. In the paper, this step is discussed in Section 2.3.

Section 2.1 presents the strategies adopted for time alignment. Depending on the time interval, two algorithms are employed: the Spherical Linear Interpolation (SLERP) algorithm [18], proposed by Ken Shoemake in 1985, and the Catmull–Rom algorithm [19], proposed by Edwin Catmull and Raphael Rom in 1974.

Section 2.2 explains how the problem of trajectory establishment is transformed into the “Smallest Color-Spanning Circle” problem through “difference transformation” and “layer stacking”. It also demonstrates that by incorporating additional parameters, the process of trajectory establishment can be included as well.

Section 2.3 describes how to construct a probability cloud map and extract the required parameters from it. In this paper, two parameters are extracted: the deviation of the trajectory from the true value and the variance of the trajectory, which correspond to the dashed parts in the flowchart.

As Section 2.4 shows, a simple nearest neighbor algorithm is employed in this paper to extract the corresponding radar trajectory based on these parameters, which are demonstrated in Section 3. Section 4 analyzes the impact of errors and hyperparameters, suggesting directions for future improvements.

2. Materials and Methods

2.1. Time and Spatial Unification

Due to the differences in how AIS and radar acquire information [16], they operate in different coordinate systems, necessitating spatial unification.

AIS data include longitude and latitude, while radar data consist of distance and angle. A unified coordinate system is necessary for effective comparison.

This paper chooses latitude and longitude for alignment between the two, projecting the radar's polar coordinates onto this coordinate system. In the subsequent steps of synchronous navigation and matching, the heading and speed coordinates need to be introduced and transformed.

Given that AIS and radar acquire information through different methods—radar through active scanning and AIS through individual ships uploading data—the acquisition intervals usually differ, making time unification necessary. Radar data are generally collected at regular intervals, which are typically longer than those for AIS data. AIS data upload times are irregular and may experience packet loss, as shown in Figure 3:

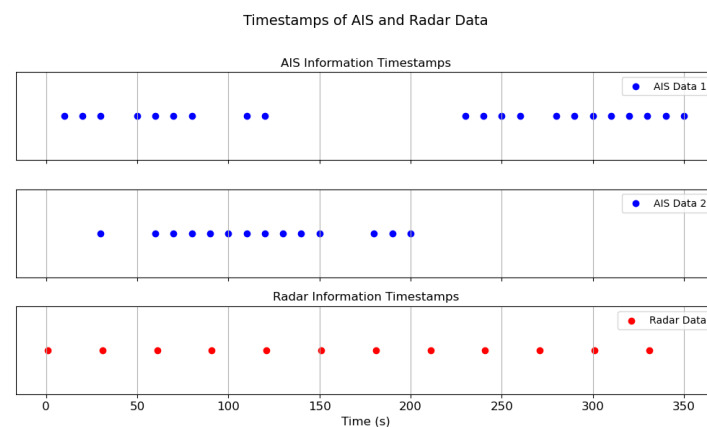


Figure 3. Illustrative diagram of timestamps in AIS and radar data.

Given that AIS data-acquisition times are governed by the transmission schedules of vessels, while radar data are collected at consistent intervals, it is standard practice to interpolate AIS trajectory data to synchronize them with radar scan intervals.

Additionally, AIS employs SOTDMA (Self-Organized Time Division Multiple Access), which may lead to the loss of AIS messages under adverse weather conditions or high traffic density, as well as potential data loss due to time slot conflicts between different AIS devices [20].

Given that the accuracy of this method heavily relies on the precision of AIS data, using a single interpolation method to align all points may introduce unnecessary errors. Ling-zhi Sang et al. proposed a method of employing different interpolation functions for different segments to obtain more accurate AIS data [21]. We also adopt varying interpolation schemes based on the temporal characteristics of the AIS data.

Linear interpolation does not account for the continuity of a vessel's speed, leading to abrupt speed changes. While spline interpolation considers speed continuity, the introduction of higher-order polynomials may result in an unrealistic trajectory when AIS data are continuously lost. The actual trajectory is influenced by various factors, such as the captain's navigation habits and sea conditions [22].

As shown in Figure 4, with known heading and position, the influence of the heading on the trajectory fitting should diminish as the distance between points increases.

Considering this scenario, the following strategy is employed for interpolation in this paper.

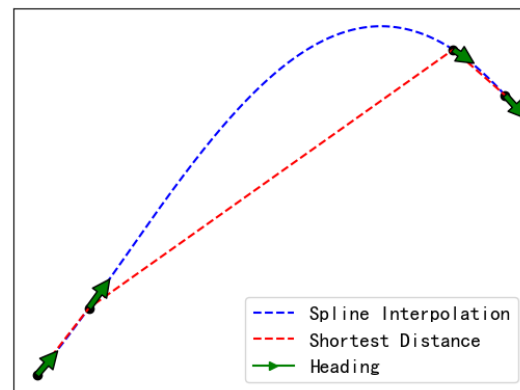


Figure 4. When significant data loss occurs, vessels are more likely to follow the shortest-distance path rather than a spline-interpolated trajectory.

When AIS data are continuous or there is minimal data loss, priority is given to maintaining the continuity of the vessel's speed, using spline interpolation. However, when there is significant data loss, the interpolation method takes the Earth's curvature into account, selecting the shortest path using SLERP interpolation, as shown in Figure 5.

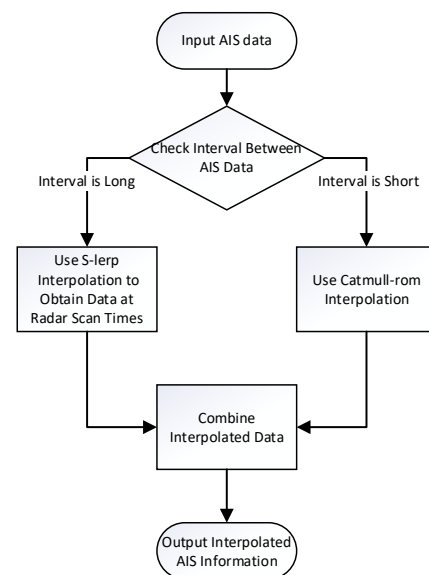


Figure 5. Interpolation strategy for continuous AIS data loss.

2.1.1. SLERP Interpolation

- Error Analysis of Interpolation Methods

When AIS data are continuously lost and the distance between two points is relatively large, consideration of the Earth's curvature becomes necessary for selecting the shortest path. Most current research conducts trajectory analysis on a zero-curvature plane [23,24]. However, The shortest distance between two points on a sphere is not a straight line but a geodesic [25].

When treating the Earth as a sphere, the geodesic (the term “geodesic” was coined by Bernhard Riemann, and it refers to the shortest path between two points on a curved surface; this concept originates from differential geometry, a branch of mathematics that focuses on the study of curves and surfaces) is the intersection of the plane formed by the two points and the sphere's center. However, directly performing linear interpolation on latitude and longitude introduces errors.

The relationship between the position of the linear interpolation point in latitude and longitude and its corresponding geodesic position is depicted in Figure 6.

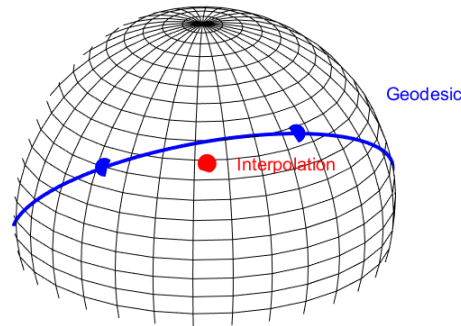


Figure 6. The result of linear interpolation does not lie on the geodesic.

Consequently, to minimize interpolation errors for the shortest path, especially when the distance between two points is significant, this paper adopts the S-lerp method [18,26]. Furthermore, it provides a straightforward proof through formulas to explain why this algorithm is suitable for latitude and longitude interpolation.

- SLERP-Based Latitude and Longitude Interpolation

SLERP interpolation utilizes quaternions. A quaternion $q = w + xi + yj + zk$ is an extension of complex numbers with the following properties:

$$\begin{cases} i^2 = j^2 = k^2 = ijk = -1 \\ ij = -ji = k \\ ji = -kj = i \\ ki = -ik = j \end{cases} \quad (1)$$

In academia, the imaginary components i, j, k of the quaternion are extracted and correspond to the coordinate system (x, y, z) , respectively, while the scalar component is conventionally set to zero.

Let P be a point in the Cartesian coordinate system, with the vector from the origin to point P denoted as \vec{OP}_{old} . Suppose this vector is rotated by an angle θ around a unit vector $\vec{u} = [u_x, u_y, u_z]$ that passes through the origin. When P is perpendicular to the unit vector \vec{u} , the results rotation can be expressed in the Cartesian coordinate system as follows:

$$\vec{OP}_{new} = \cos(\theta) \cdot \vec{OP}_{old} + \sin(\theta) \cdot \left(\vec{u} \times \vec{OP}_{old} \right) \quad (2)$$

Here, \vec{OP}_{new} represents the result of rotating the vector around the axis \vec{u} . To transform the point in the Cartesian coordinate system into a quaternion, we have $q_p = 0 + P_xi + P_yj + P_zk$. Similarly, when \vec{u} is also converted into a quaternion $q_u = 0 + u_xi + u_yj + u_zk$, it can be observed that the quaternion representation q_v of the vector $\vec{V} = \vec{OP}_{old} \times \vec{u}$ is the product of the quaternion representation of \vec{u} and \vec{OP}_{old} , as shown in Equation (3):

$$q_v = (0 + u_xi + u_yj + u_zk)(0 + P_xi + P_yj + P_zk) \quad (3)$$

At this point, the rotation can be simply represented as a left multiplication by a quaternion. Let q'_p be the quaternion representing the point P after it has been rotated by an angle θ around the unit vector \vec{u} . this results in the following expression:

$$q'_p = (\cos \theta + \sin \theta \cdot u_xi + \sin \theta \cdot u_yj + \sin \theta \cdot u_zk) \cdot q_p \quad (4)$$

Hence, considering the interpolation between two points P_1 and P_2 , on a sphere, by adjusting the value of θ , the rotation from \vec{OP}_1 to \vec{OP}_2 can be smoothly represented in the SLERP algorithm [26].

Let q_t be the quaternion of the interpolation point, with the known adjacent points in the trajectory being q_0 and q_1 . The SLERP interpolation formula is shown in Equation (5):

$$q_t = \frac{\sin(1-t)\theta}{\sin\theta}q_0 + \frac{\sin t\theta}{\sin\theta}q_1 \quad (5)$$

where $t \in [0, 1]$ is a proportional coefficient of the interpolation point, which is obtained from Equation (6):

$$t = \frac{t_{radar} - t_{ais,(k-1)}}{t_{ais,k} - t_{ais,(k-1)}} \quad (6)$$

where $t_{radar} \in [t_{ais,(k-1)}, t_{ais,k}]$ represents the radar scan time, with $t_{ais,k}, t_{ais,(k-1)}$ being the corresponding times for the k -th and $(k-1)$ -th points in the AIS sequence.

With the introduction of SLERP, the overall processing workflow can be summarized as follows:

1. Convert latitude and longitude to a quaternion-based Cartesian system.
2. Apply SLERP interpolation.
3. Reconvert the results to latitude and longitude.

The Earth is considered a sphere with its center O located at the origin of a Cartesian coordinate system. For the two known points at times $t_{ais,(k-1)}$ and $t_{ais,(k)}$, the corresponding quaternions q_0 and q_1 can be obtained by Equation (7):

$$\begin{cases} w = 0 \\ x = r \cdot \cos(lat_{rad}) \cdot \cos(lon_{rad}) \\ y = r \cdot \cos(lat_{rad}) \cdot \sin(lon_{rad}) \\ z = r \cdot \sin(lat_{rad}) \end{cases} \quad (7)$$

Here, lat_{rad}, lon_{rad} are the latitude and longitude converted to radians, while w, x, y, z are the components of the quaternion $q = w + xi + yj + zk$. r represents the radius of the Earth. With this, SLERP interpolation can be accomplished.

Let the quaternions for the two known points be defined as $q_0 = r \cdot x_0i + r \cdot y_0j + r \cdot z_0k$ and $q_1 = r \cdot x_1i + r \cdot y_1j + r \cdot z_1k$. The interpolated quaternion q_t at time t_{radar} can be expressed using Equation (5), where θ is calculated from Equation (8):

$$\theta = \text{acos}(x_0x_1 + y_0y_1 + z_0z_1) \quad (8)$$

Finally, the latitude and longitude at time t_{radar} can be calculated using Equation (9):

$$\begin{cases} lat_t = \frac{180}{\pi} \text{asin} \frac{z}{r} \\ lon_t = \frac{180}{\pi} \text{atan} \frac{y}{x} \\ \text{s.t. } lat_t \in [-180, 180], lon_t \in [0, 360] \end{cases} \quad (9)$$

2.1.2. Catmull–Rom

The Catmull–Rom curve is a type of spline that provides smooth interpolation through a set of control points [19]. The Catmull–Rom curve is specifically designed to pass exactly through each of the given control points, while also offering the advantages of simple computation and not requiring the calculation of derivatives.

The curve is defined by a set of points and uses a parameter to determine the curve's position between these points. The Catmull–Rom spline ensures continuity and smooth transitions between segments, making it a popular choice for creating smooth, flowing paths. And the formula is that:

$$\vec{C}(t) = \frac{1}{2} \left[\left(2\vec{P}_1 \right) - \left(\vec{P}_0 - \vec{P}_2 \right) t + \left(2\vec{P}_0 - 5\vec{P}_1 + 4\vec{P}_2 - \vec{P}_3 \right) t^2 - \left(\vec{P}_0 - 3\vec{P}_1 + 3\vec{P}_2 - \vec{P}_3 \right) t^3 \right] \quad (10)$$

where $\vec{P}_0, \vec{P}_1, \vec{P}_2, \vec{P}_3$, are the control points, $\vec{C}(t)$ is the interpolation point. $t \in [0, 1]$ is a proportional coefficient of the interpolation point, which is obtained from Equation (6):

Since the Catmull–Rom method requires four control points for interpolation, if fewer than four points are available, additional points are estimated based on the direction of travel to reach the required number of points.

2.2. Stacking & Diffusion Transformation

When establishing radar tracks in complex environments, due to the large amount of data, traditional parallel processing approaches include clustering division and spatial division [27].

“Spatial division” adopts the idea of multi-node regional processing, dividing the processing airspace equally according to the number of processing nodes obtained. Each block is associated separately, and the main node manages the track batch numbers. This method is obviously not suitable for track establishment in complex environments, as there is a possibility of missed or incorrect divisions.

“Clustering division” adopts the idea of dynamic irregular data division. Starting from the first batch of points, it associates the next batch of points according to a threshold range, then updates the positions, and repeats this process until the association is complete. This divides the space into multiple independent autocorrelated data areas, eliminating data coupling. When the point distribution is sparse, clustering division has the advantage of eliminating errors. The general formula for clustering division can be expressed as Equation (11):

$$P_{k+1,c}^{\rightarrow} = f_k \left(P_{k,c}^{\rightarrow}, cog_{k,set}, v_{k,set} \right) \quad (11)$$

where $P_{k,c}^{\rightarrow}$ is the estimated value of the k -th point in the radar track, and $cog_{k,set}$ and $v_{k,set}$ represent the course over ground (COG) and the velocity in the COG direction of the point trace $P_{k,set}^{\rightarrow}$ that is associated with the track in the k -th batch of point traces. The associated point $P_{k,set}^{\rightarrow}$ can be obtained from Equation (12):

$$P_{k,set}^{\rightarrow} = \underset{P_k \in D}{\operatorname{argmin}} \left(\left| P_k^{\rightarrow} - P_{k,c}^{\rightarrow} \right| \right) \quad (12)$$

where D is the set of all radar point traces in the $(k+1)$ -th batch. In some cases, $P_{k,set}^{\rightarrow}$ may not be a single point. To avoid deviations in the search, we may opt for the nearest set of the first n tracks.

When using $P_{k+1,c}^{\rightarrow}$ to associate points in the $(k+2)$ -th batch of radar data to this track, if there is an error δ_k in the track point of the center point $P_{k,c}^{\rightarrow}$, it may lead to wrong association of points, which can further result in a series of error. This issue is particularly evident during the initial stage of track establishment [28].

The Hough Transform addresses the problem of quickly and accurately establishing target tracks in a strong clutter environment during the initial phase of track formation. It employs methods of parameter space transformation and vote accumulation. Since Smith et al. first introduced the Hough Transform to track formation [29], numerous variants have emerged, transforming the track formation problem into a feature curve detection problem [30]. However, limited by the fact that the feature curves themselves are unknown, this method can easily generate false tracks in complex environments and may miss non-linear true tracks. By incorporating AIS, this paper reduces the computational

overhead during the navigation initialization phase while avoiding error propagation in complex environments. Initially, the point cloud data from each radar scan are transformed and then stacked. Subsequently, associations are made across all time-series scan points, which prevents the possibility of data drift starting from a wrong point and also reduces time complexity.

When incorporating trajectory and heading information from AIS, it becomes possible to bypass the radar initialization process and complete the initialization directly during the matching phase, thus avoiding information reuse.

2.2.1. Processing of Location Information

To uniformly represent the information from all batches, the radar scan points are stacked in the form of layers, as shown in Figure 7a. Each layer represents the result of a single radar scan, and the red dots indicate the positions of the AIS data after interpolation at the corresponding radar scan time points.

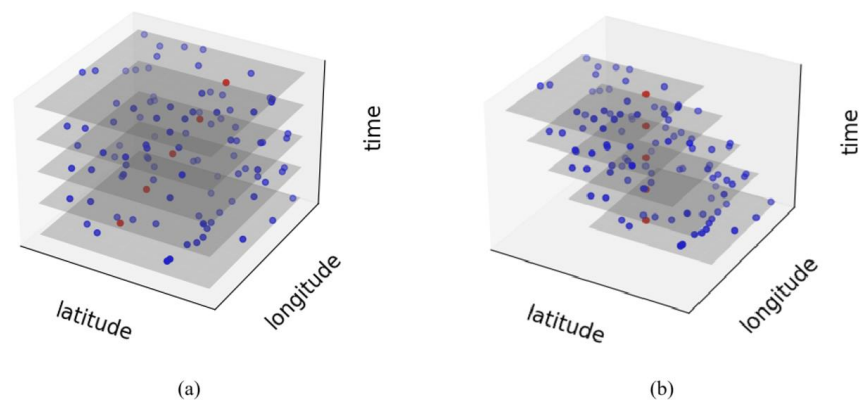


Figure 7. Illustration of point stacking and difference transformation, where red points represent AIS data and blue points represent radar data: (a) how the layers of radar and AIS position information are stacked; (b) the result after applying difference transformation to each layer, positioning the AIS data at the zero point.

After layer stacking, all radar data and AIS data appear in the same layer. A straightforward idea is as follows: Is it possible to find a group of points resembling the red dots from the point cloud directly?

To achieve this functionality, this paper applies the transformation shown in Equation (13) to the stacked AIS data, and the result is displayed in Figure 7b.

$$P_{new,i}^{\rightarrow} = P_{old,i}^{\rightarrow} - \vec{A}_i \quad (13)$$

Here, $p_{old,i}^{\rightarrow}$ represents the set of all radar points scanned during the i -th scan, \vec{A}_i represents the AIS data value at the time of the i -th radar scan, which is obtained through the interpolation method mentioned previously, and $P_{new,i}^{\rightarrow}$ represents the transformed radar points scanned during the i -th scan. For each scan, every layer undergoes a difference transformation.

Assign different colors to radar points from different batches, as shown in Figure 8a. The problem of ship track establishment is then transformed into the problem of “drawing a circle that encompasses one point of each color”, as illustrated in Figure 8b.

Here, due to our transformation, the AIS data are now located at the origin (0,0). We assume that the yellow circle in Figure 8b can be represented by Equation (14):

$$(x - x_t)^2 + (y - y_t)^2 = r_t^2 \quad (14)$$

At this point, $\sqrt{(x_i^2 + y_i^2)}$ can represent the average deviation of this set of points relative to the AIS data, while r_i^2 can be interpreted as the shape similarity of this set of points compared to the AIS track.

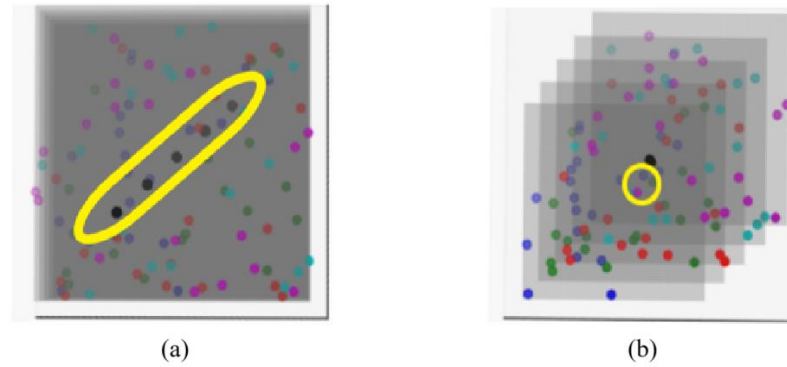


Figure 8. The curve fitting problem transforms into finding the optimal circle: (a) the curve fitting problem, with the search results located within the yellow ellipsoid; (b) the problem of finding a circle that “contains one point of each color”. The black dot represents the origin. Due to systematic errors in the radar, the center of the circle is typically not at the origin, and the search results are located within the yellow circle.

If we desire the shape of the radar points to be as similar as possible to AIS points, the problem further transforms into the “Smallest Color-Spanning Circle (Smallest Color-Spanning Objects)” [31] problem. Alternatively, we can also construct a function that incorporates the average deviation to calculate a score.

In this method, it is noteworthy that since the stacked points do not involve temporal order information, the radar track establishment is no longer influenced by the sequence order, exhibiting good robustness in complex environments. The process of drawing the circle in this method involves significant computational effort. To address this issue, this paper adopts a combination of probability clouds and clustering methods. For details, please refer to Section 2.3.

2.2.2. Transformation Based on Kinematics

Furthermore, if we introduce the track and heading information from AIS data into the aforementioned method, we can actually encompass both track establishment and matching within it.

In traditional track establishment methods, radar plots are used to establish tracks, which are then matched with time-aligned AIS data, as shown in Equation (15):

$$\underset{i}{\operatorname{argmin}} \left\| \begin{bmatrix} \sin(\operatorname{cog}_{n-1, \text{set}}) \\ \cos(\operatorname{cog}_{n-1, \text{set}}) \end{bmatrix} v_{n-1, \text{set}} + \begin{bmatrix} \operatorname{lon}_{n-1, c} \\ \operatorname{lat}_{n-1, c} \end{bmatrix} - \begin{bmatrix} \operatorname{lon}_{i, \text{ais}} \\ \operatorname{lat}_{i, \text{ais}} \end{bmatrix} \right\|_2 \quad (15)$$

where $(\operatorname{lon}_{i, \text{ais}}, \operatorname{lat}_{i, \text{ais}})^t$ represents the longitude and latitude coordinates at time n after interpolating the AIS trajectory. If we assume that the errors in AIS data are sufficiently small, then the results obtained by this method are numerically equivalent to those given by Equation (16):

$$\underset{i}{\operatorname{argmin}} \left\| \begin{bmatrix} \sin(\operatorname{cog}_{n, \text{set}}) \cdot v_{n, \text{set}} - \sin(\operatorname{cog}_{n, \text{set}}) \cdot v_{n, \text{set}} \\ \cos(\operatorname{cog}_{n, \text{set}}) \cdot v_{n, \text{set}} - \sin(\operatorname{cog}_{n, \text{set}}) \cdot v_{n, \text{set}} \\ \operatorname{lon}_{n, c} - \operatorname{lon}_{i, \text{ais}} \\ \operatorname{lat}_{n, c} - \operatorname{lat}_{i, \text{ais}} \end{bmatrix} \right\|_2 \quad (16)$$

Therefore, when directly matching combinations of track, heading, longitude, and latitude under Euclidean distance to find the optimal track, it essentially encompasses the functionality of traditional methods that use radar point trace associations for track establishment.

At this point, since the information in the point $\vec{P}_{k,c}$ in Equation (11) has been extended from merely including longitude and latitude to also including heading and speed, the meaning of the distance between longitude and latitude remains the same. However, heading and speed are not comparable to longitude and latitude in the same way. Therefore, simply using the “Smallest Color-Spanning Sphere” is not appropriate. Instead, we can consider introducing coefficients a and b as shown in Equation (17) or alternatively use a rectangular prism.

$$a(lat - lat_i)^2 + a(lon - lon_i)^2 + b(\cos(cog)v - \cos(cog_i)v_i)^2 + b(\sin(cog)v - \sin(cog_i)v_i)^2 = r^2 \quad (17)$$

2.3. Parameter Extraction

The previous section presented a method for simultaneously constructing and matching routes using AIS data. However, to achieve matching, there must be a way to determine the “optimal circle.” Assuming the existence of an optimal solution (true value), how can we search for it?

We know that when the search tree is sufficiently large, we can inevitably find the optimal solution (true value). According to the MIN-MAX principle, it is feasible to either first search for the distribution of all points that can be enclosed by a circle with radius r and then search for the circle closest to the center among these circles, or to first search for the top k circles closest to the center and then search for the circle with the smallest radius among these circles—as long as the search tree is sufficiently large. However, we don’t always have a sufficiently large search tree, and the optimal solution of the scoring function may not necessarily be the true value. Therefore, we consider traditional search methods [32] as described in Equation (18):

$$\begin{cases} S^*(r) = \min S(lat_i^2 + lon_i^2) \\ \max_r S^*(r) = \max \min S(lat_i, lon_i, r) \end{cases} \quad (18)$$

Noting the optical circle’s center (lat, lon) is equidistant from all points of different colors with a distance of r , this suggests that we can actually generate a circle with a radius of r for each color point and then find the intersection of all these circles, as illustrated in the Figure 9a:

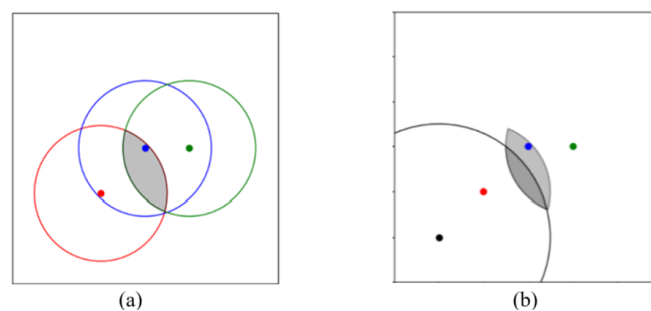


Figure 9. Visualization of the optimal solution search for Equation (18). The black dot represents the origin, which is the position of the AIS data after applying the difference transformation. The other colored dots represent radar track points at different time instances. (a) The intersection represents the fixation of the variable r . (b) When the set of shaded areas is reduced to a single element, it represents the optimal solution for the current scoring function.

For any point within the intersection (gray area) in Figure 9, a circle with a radius of r can be drawn to enclose all points of different colors. Subsequently, by taking the

intersection of this set with a circle centered at the origin and having a radius of r_o (denoted as Circle O), all points that meet the search requirements can be obtained, as illustrated in Figure 9b.

The search method for Equation (18), as depicted in the above figure, is equivalent to the following steps:

1. Adjust r_o . For any given r_o , find a suitable r such that the intersection becomes a single point. Due to the increased constraints, the solution set becomes a line or a point (in extreme cases).
2. Select the point on this line that is closest to the origin, narrowing down the solution set to a single optimal point.

However, the problem lies in the fact that the solution space remains vast. In a spatially prioritized search method, it is still necessary to solve for every r , which consumes a significant amount of computational resources. Additionally, the scoring function may not closely align with real-world scenarios, so the “optimal solution” obtained through this process may not necessarily reflect the true value.

- Algorithm complexity optimization

Assume that each plot no longer exists only at its original location but is instead dispersed as a probability cloud. The closer the measurement point is to the radar plot, the higher the likelihood, and vice versa. In this case, assuming that the likelihood I follows a normal distribution, it can be determined by Equation (19).

$$I = \begin{cases} \frac{1}{\sqrt{2\pi}r^2} e^{-\frac{(r-d)^2}{2r^2}}, & d \leq r \\ 0, & d > r \end{cases} \tag{19}$$

where d represents the distance from the measurement location to the radar plot. The reason for setting a threshold r is that we do not want to accumulate a value larger than the true value at the center of gravity of a batch of plots. The point trajectories in Figure 9 have been diffused into a probability cloud, as shown in Figure 10.

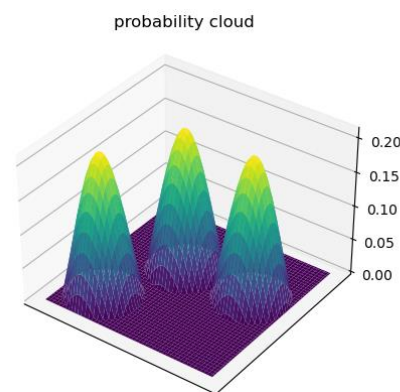


Figure 10. Diffusion of radar plots from Figure 9 into a probability cloud based on the probability distribution in Equation (21), where the likelihood I is represented by intensity.

For each batch of radar plots, perform the aforementioned transformation to obtain a layer similar to Figure 10. Likewise, for ease of visualization, different batches of radar plots are represented using different colors of intensity. By stacking these layers, the output is stored in the format described by Equation (20):

$$\begin{pmatrix} [I_{1,(1,1)} & I_{2,(1,1)} & I_{3,(1,1)}] & \cdots & [I_{1,(1,N)} & I_{2,(1,N)} & I_{3,(1,N)}] \\ \vdots & & & \ddots & \vdots & & \\ [I_{1,(M,1)} & I_{2,(M,1)} & I_{3,(M,1)}] & \cdots & [I_{1,(M,N)} & I_{2,(M,N)} & I_{3,(M,N)}] \end{pmatrix} \tag{20}$$

where in $I_{i,(x,y)}$, i represents the corresponding batch, and (x, y) represents the corresponding coordinates. $I_{i,(x,y)}$ depends on the likelihood of each point in the i -th batch of radar plot data at that point (x, y) .

Next, a filter is used to iterate through all points in space and search for S_{max} , as shown in Equation (21):

$$S_{max} = \max_{(x,y) \in D} \left(\min_{1 \leq i \leq k} I_i(x, y) \right) \tag{21}$$

where D represents the entire search space, and k is the number of radar scan batches. Thus, the problem of difficult searching is solved by combining the two-step search into one step.

- Function Optimal Solution Non-True Value Problem

In deep learning, to address the issue of optimal solutions that are not true values, the adopted method is to avoid overfitting by preventing over-learning. In this context, when optimization is not fully completed, the solution set is not constrained to a single point. This behavior can be abstracted as the actions of the following n filters, each of which traverses the entire space and selects all points that meet the requirements based on the rule in Equation (22):

$$\min_{1 \leq i \leq k} I_i(x, y) \geq S_n \tag{22}$$

where k represents the number of radar scan batches, S_n is the threshold for different filters, and the true value may fall within any point filtered by that filter.

By clustering all the results obtained from “avoiding over-learning”, the size of the circle can be determined based on the clustering radius. The average distance from the completed radar tracks to the corresponding AIS data can be obtained based on the distance between the cluster centers and the circles.

Especially when there are enough radar plots, considering that the random distribution of radar usually approaches a normal distribution, the center of the cluster at this time is the result of the true value plus the system bias.

2.4. Indexing Radar Trace

Finally, the “cluster center” obtained in Section 2.3 is used as the bias. Then, by applying the nearest-neighbor search [33] on AIS data with the bias added, radar point tracks can be extracted, as shown in Figure 11:

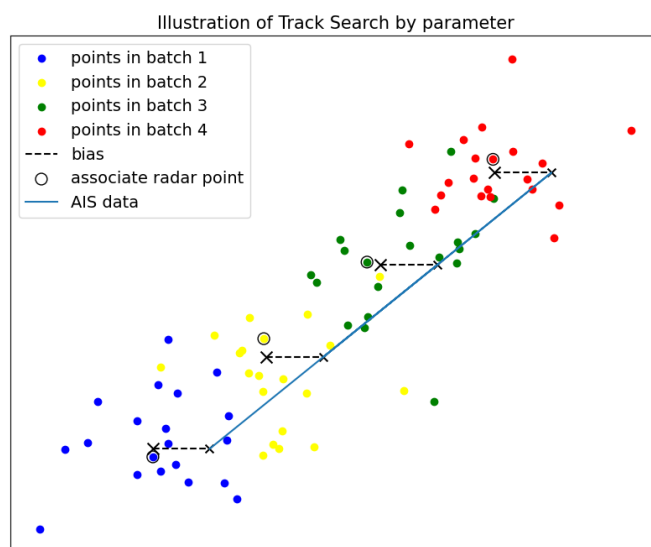


Figure 11. Illustration of Nearest-Neighbor Search.

As discussed earlier, performing a transformation on the space defined by $(lon, lat, \cos(cog_i)v_i, \sin(cog_i)v_i)$ effectively accomplishes track filtering based on kinematic princi-

ples. Therefore, in most cases, it is unnecessary to use the parameter r to create clustered indices, subsequently relying on kinematic analysis for filtering.

3. Results

- Simulation Data Validation

To support the findings and conclusions of this study, a simple demo of the algorithm has been uploaded to GitHub. Readers can access the code on GitHub (<https://github.com/Holygre/AIS-Trajectory-Matching-From-Radar-Points>, accessed on 26 October 2024) and follow the instructions in the README file to conduct tests. We encourage readers to test the code themselves to verify the reproducibility and accuracy of this study. Below, we present the results of one such test.

We constructed 20 batches of simulated radar plots, with parameters expressed in kilometers. The AIS trajectories were generated by uniformly sampling 20 points from the sine function $y = \sin(x)$ within the interval $x \in [0, 2\pi]$. Then, the radar trajectories were generated by adding a bias $\delta_{random} \sim \mathcal{N}(x = 0, y = 3, \rho = 0, \sigma_1 = \sigma_2 = 0.3)$ to each point of the AIS trajectories. The relationship between the AIS and radar trajectories is shown in Figure 12:

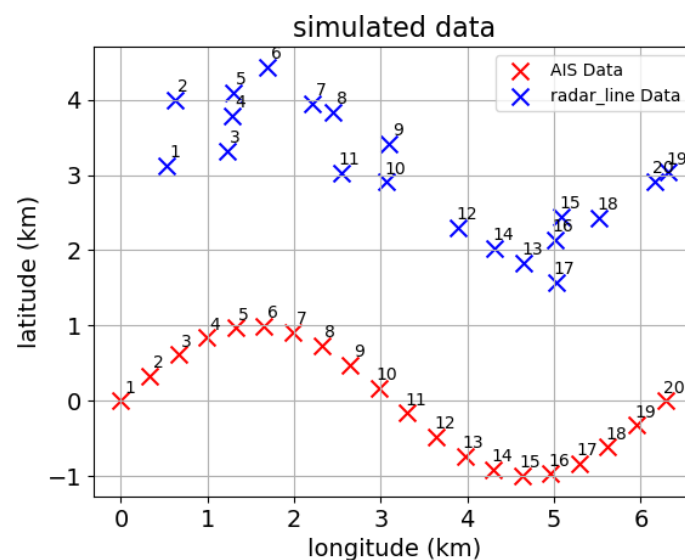


Figure 12. Radar data and simulated data. The numbers above the points indicate the results of the i -th batch of radar scans at that specific time instance.

Here, the i -th point in the radar trajectory corresponds to the result of the i -th radar scan and has been temporally aligned with the i -th batch of AIS trajectories. Furthermore, for each batch of radar plots, we introduced 100 uniformly distributed interference points within a rectangular region bounded by $(-3, -3)$ and $(8, 8)$ to simulate interference. In Figure 13, different colors are used to distinguish the interference points from various batches, as illustrated:

For each batch of radar plots, perform differential transformation relying on AIS data, and then stack these points. The results are shown in Figure 14:

Based on the distribution in Equation (21), radiate all points outward to form a density distribution, and store them separately in various layers. The probability distribution of the first batch of plots is shown in Figure 15:

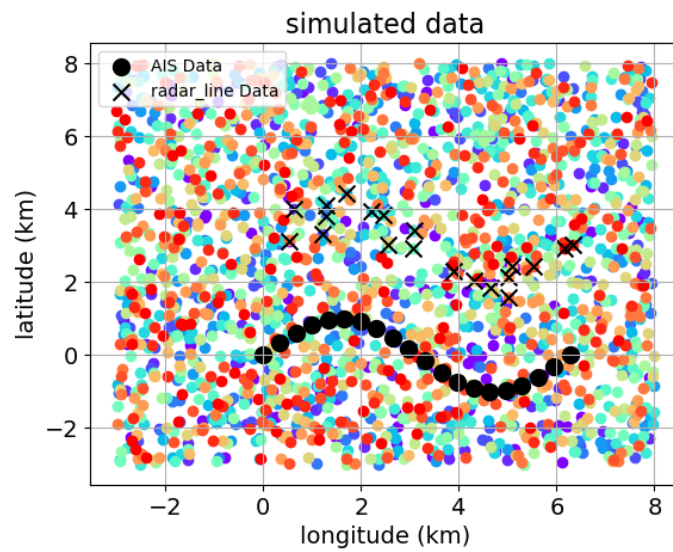


Figure 13. Simulated radar trace data, where points of the same color indicate they belong to the same batch of radar scans.

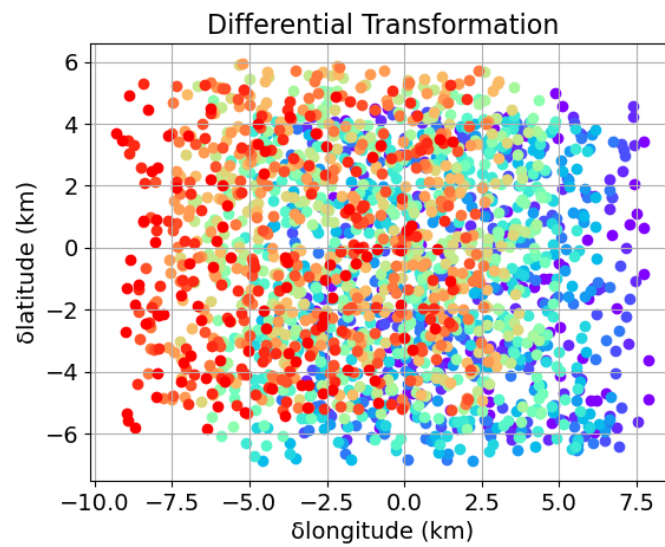


Figure 14. The superimposed results after differential transformation. After the differential transformation, the AIS data are positioned at the origin of the image.

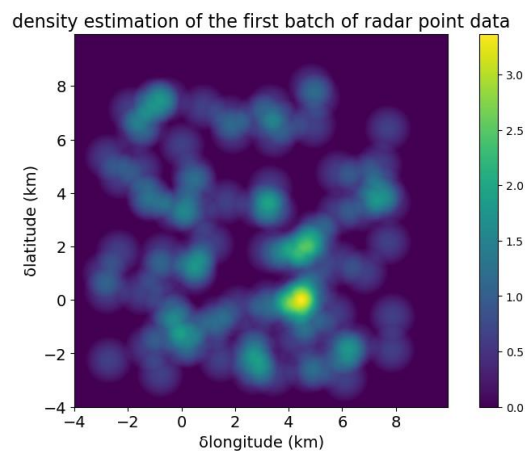


Figure 15. Probability cloud of the first batch of data, with a search radius of 1. The brightness of overlapping points is not superimposed.

Superimpose all layers, and then randomly select several thresholds for the superimposed layer. Use a filter to filter out the tracks that do not meet the requirements. The thresholds selected here are 0.4, with a search precision of 0.01. The obtained results are shown in Figure 16.

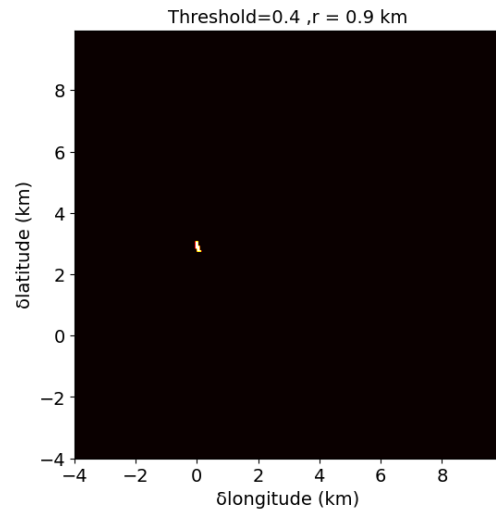


Figure 16. The result of filtering all layers, showing only unique tracks selected.

For Figure 16, when using this image for clustering, the cluster center C is at $(-5 \times 10^{-2}, 2.95)$. This implies that the average deviation between the radar data and AIS trajectories is approximately $(0, 2.95)$, which closely matches the set average deviation δ of $(0, 3)$. As the number of trajectory batches increases, the deviation approaches the true value. For 20 batches of radar trajectories, with a resolution of 0.01, multiple tests were conducted using different random seeds. In 10 tests, the difference between the calculated deviation and the set deviation was approximately 0.082. Furthermore, when the variance σ is reduced, the deviation difference decreases rapidly.

When a bias of $(0, 2.95)$ is added to the AIS data and the nearest points are selected, most of the points in Figure 12 can be recovered. This result is the same as when a bias of $(0, 3)$ is added, as shown in Figure 17.

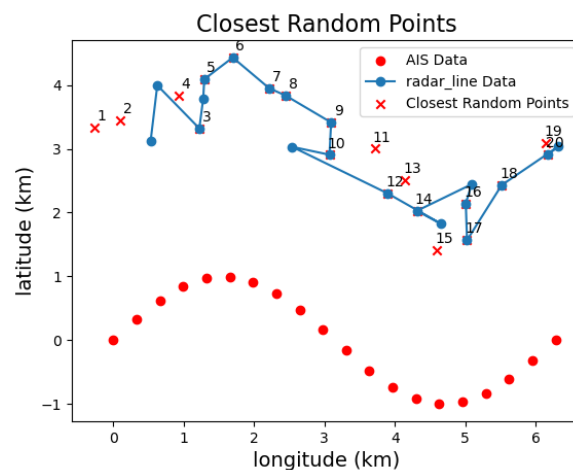


Figure 17. The algorithm’s selected trajectory points closely match the radar trajectory points. The numbers above the points indicate the results of the i -th batch of radar scans at that specific time instance. Some errors, however, are indistinguishable with only latitude and longitude. Introducing cog and v can help further screen these errors.

The algorithm demonstrates high tolerance to noise and permits significant deviations. Readers are encouraged to experiment with the program available on GITHUB. Below, we briefly present the results obtained with different thresholds, all of which yield cluster centers close to the true value of (0, 3).

For the results in Figure 18, using clustering algorithms with different numbers of clusters can all yield cluster centers close to (0, 3). Specifically, the results with thresholds of 0.2, 0.3, and 0.4 are all consistent with those in Figure 17. To eliminate such errors, drawing on traditional methods for establishing tracks, we need to consider introducing $\cos(cog) \cdot v$ and $\sin(cog) \cdot v$ for filtering.

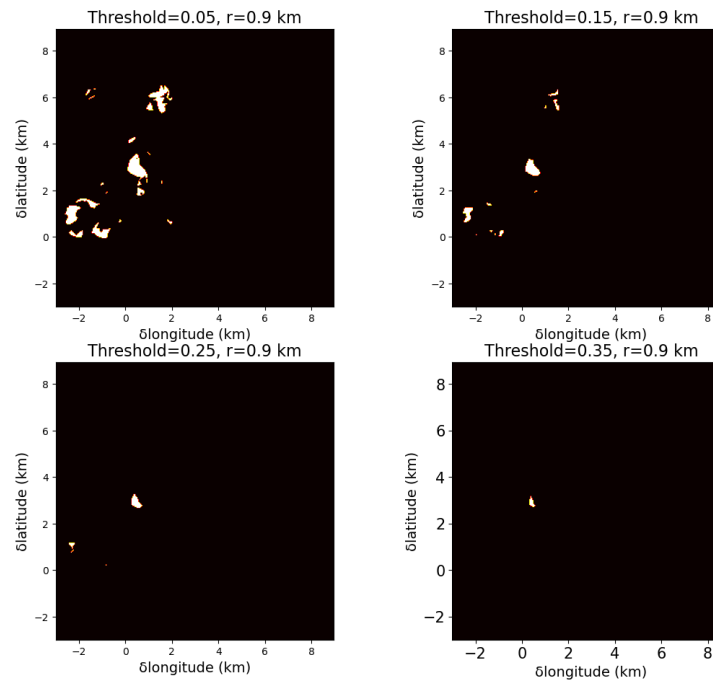


Figure 18. The positions of the tracks that meet the requirements after selecting different parameters when the random error is further increased. This method has a high tolerance for parameters.

Figure 19 shows the track identified based on the bias after adding noise with a mean of $\mu = (0, 3, 0, 0)$ and a variance of $\Sigma = \text{diag}(0.09, 0.09, 0.01, 0.01)$ to the points in the new coordinate system. Multiple tests confirmed the accuracy of the results.

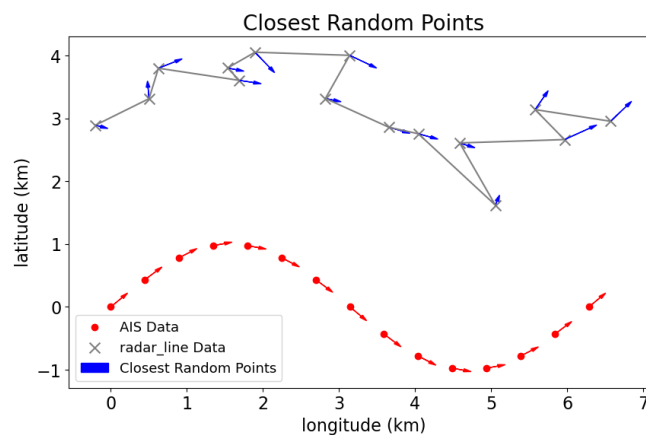


Figure 19. Nearest-neighbor algorithm after adding two additional dimensions, effectively eliminating the erroneous selections mentioned in Figure 17.

- Experimental Data Validation

A dataset was selected from high-frequency surface wave radar, with intervals of 2 min and covering a latitude and longitude span of 0.8 degrees. Differential transformations were then applied to this dataset based on an AIS trajectory within the selected area. The results are presented in Figure 20:

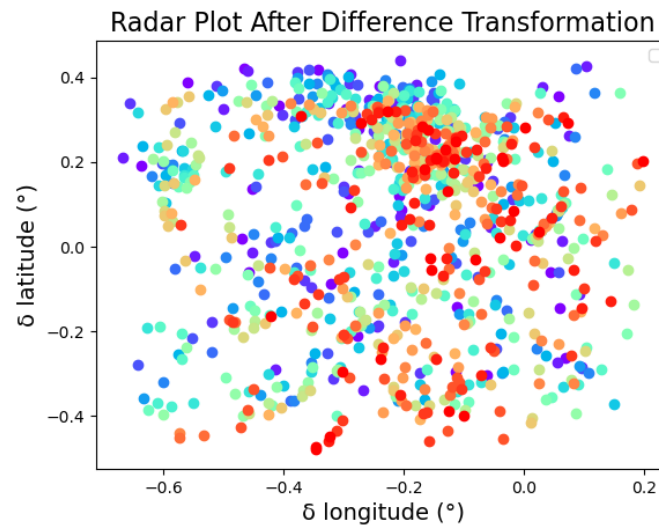


Figure 20. Radar plot after difference transformation. A total of 29 different colors are used to represent 29 batches of radar plot data.

Similar to Figure 14, the points in Figure 20 are color-coded according to different batches of radar track data. Subsequently, a probability cloud is generated. Notably, performing this step prior to the differential transformation can reduce computational load when matching multiple trajectories. Given that the L2 norm is not index-friendly, the L1 norm was used to generate the probability cloud in practice, which enhanced computation speed. Feature parameters were then extracted through clustering, as shown in Figure 21:

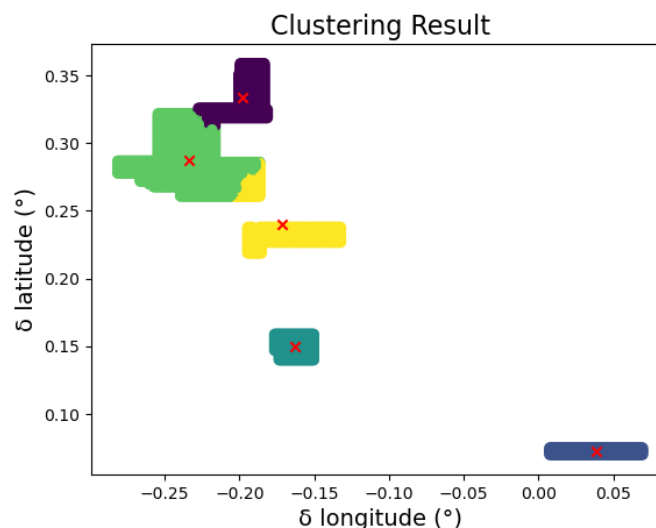


Figure 21. Clusters extracted from the L1 norm-based probability cloud, with blue cluster centers identified as true bias. with each cluster represented by a distinct color. The centroids of the clusters are indicated by cross marks of the red color.

The trajectories extracted by this algorithm and those obtained from the radar subsystem are shown in Figure 22. As observed, the two trajectories are quite similar; however,

due to the filtering operations required for accuracy in traditional algorithms, some discrepancies remain between the trajectories.

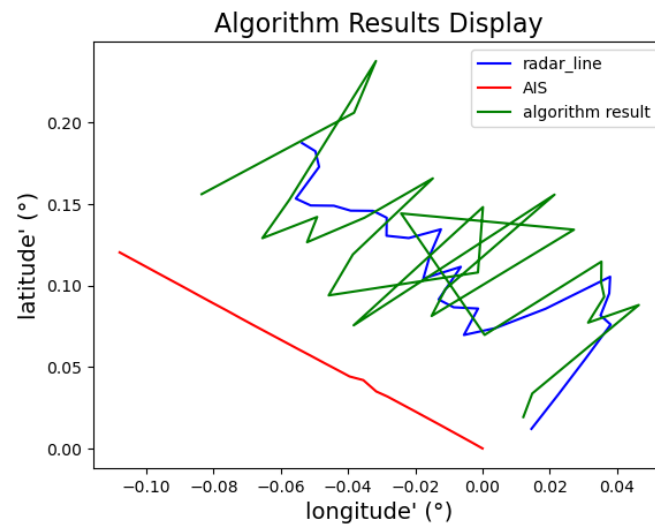


Figure 22. Results of the algorithm, with the starting point shown in the bottom right.

Figure 23 presents the deviations of both trajectories relative to AIS data. It is evident that the deviations from the true values are similar for both trajectories; however, the filtered data exhibit more stable deviations, while the trajectory extracted directly from point data shows greater fluctuation. Overall, the trends are aligned, indicating that trajectory establishment and matching based on this algorithm are successful.

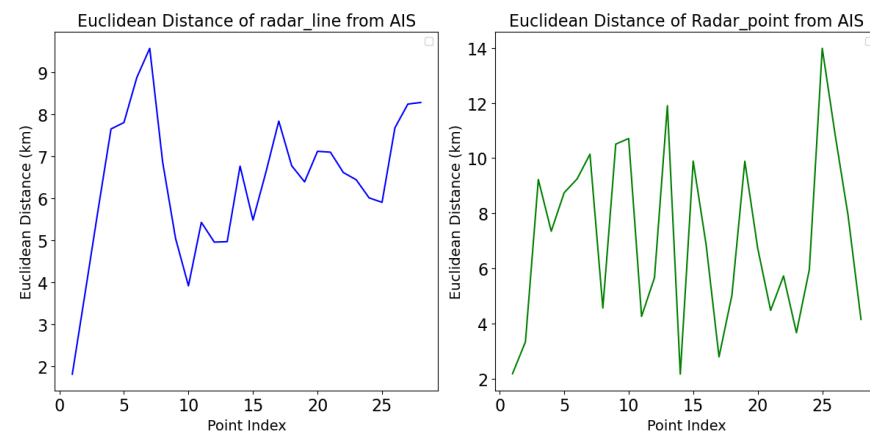


Figure 23. Error analysis of both trajectories compared to the AIS trajectory.

4. Discussion

While the experimental results on simulated data are accurate and stable, several issues have emerged when applying this algorithm to real data. The following is an analysis of the application results of this algorithm on actual data.

4.1. Impact of Hyperparameter

The hyperparameters introduced in this paper include resolution, threshold size, and the number of clusters. The following discussion will focus on the impact of these parameters in a latitude–longitude coordinate system.

- **Threshold size:**

The size of the threshold actually depends on the density of radar tracks in the target space. Excessively high or low thresholds can lead to different results, as shown in Figure 24.

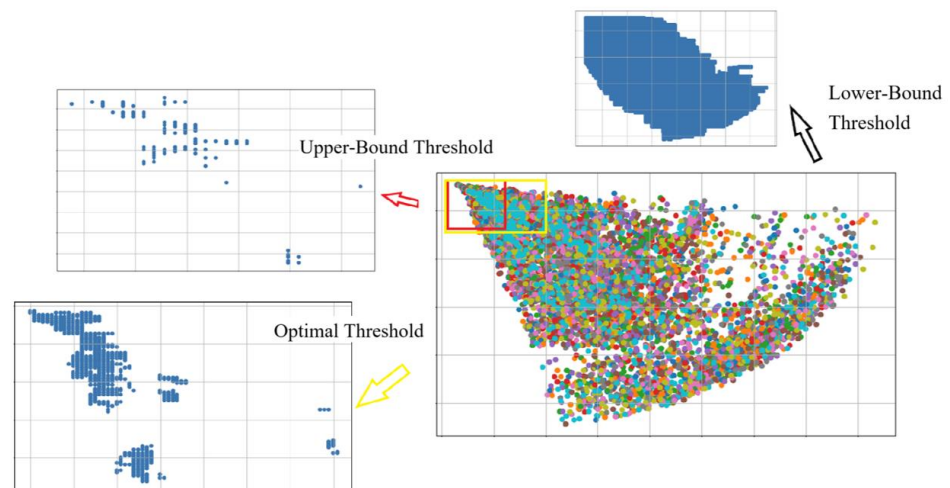


Figure 24. Different thresholds selected for the filter may result in different outcomes. Similar to this, different colors are used to represent different batches of radar plot data, with points of the same color indicating radar plots from the same batch.

- Number of clusters:

After filtering, selecting different numbers of clusters may also have different impacts, as shown in Figure 25:

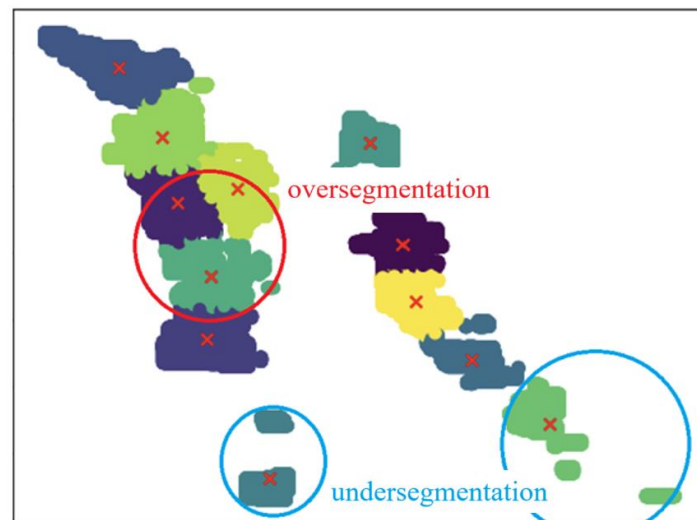


Figure 25. The number of clusters varies across different regions, with each cluster represented by a distinct color. The centroids of the clusters are indicated by cross marks of the red color.

As the area of application for this algorithm expands, issues may arise where some regions have an excessive number of clusters while others have too few. Such clustering errors can lead to significant fluctuations in deviation parameters, introducing errors. In Figure 22, this issue is temporarily addressed by subdividing the scanned region, though fundamentally, modifications to the clustering algorithm are required.

- Resolution:

This refers to the distance interval at which the intensity of various types of radiation is determined. A higher resolution results in more precise clustering outcomes and more accurate central positions. However, an excessively high resolution can lead to significant memory burdens.

Therefore, selecting an appropriate resolution is important. For instance, considering that the error of AIS is less than 50 m, it is reasonable to choose a resolution of either 50 m or 100 m intervals. For course and speed, the issue of resolution becomes relatively more complex.

- Weights:

When incorporating parameters $\cos(\text{cog})v$ and $\sin(\text{cog})v$ assigning different weights may introduce additional errors due to differing radar accuracies. Excessive weighting of any single parameter can lead to erroneous associations, as shown in Figure 17. This paper employs a simple approach, setting all weights to 1 for matching; however, further adjustments are needed in practical applications.

4.2. Error Analysis

Due to the assumptions made about noise in this paper, a certain amount of error is actually introduced:

This paper assumes that the error necessarily forms a pattern like Figure 26, where the track formed by connecting radar points is a translation of the true track plus noise. In reality, the error of the track formed by connecting radar points may not be exactly like this:

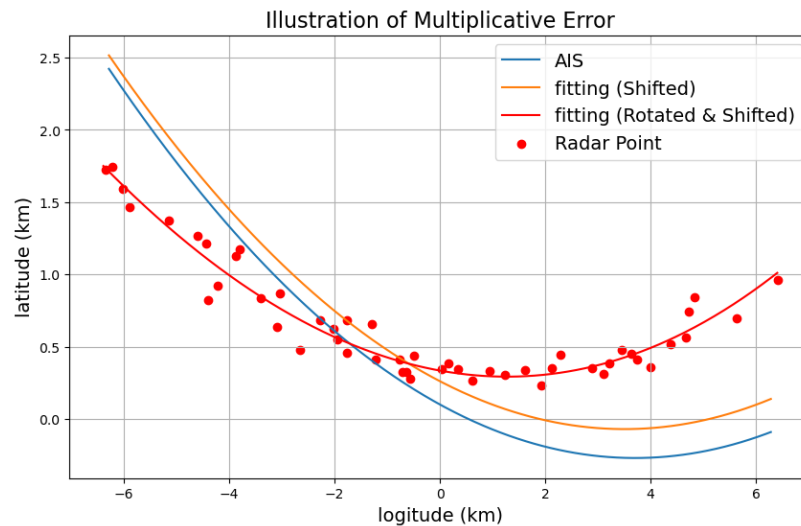


Figure 26. When the systematic error of radar points is related to the coordinate system and rotation occurs locally, using a rotation matrix and displacement for fitting will yield better results.

Of course, Some research points out that radar points can definitely be expressed as the superposition of true values and stationary noise [34], it is feasible to use AIS true values to fit the track. In this case:

$$(lon_{radar}, lat_{radar}) = (lon_{ais}, lat_{ais}) + n \tag{23}$$

However, noise n may not stationary [35]. For example, while the radar’s distance measurement may be accurate, there can be deviations in the angle. We assume that the angular error $\delta\theta$ can be expressed as a combination of the average bias θ_{bias} and a variance θ_n , as shown in Equation (24):

$$\delta\theta = \theta_{bias} + \theta_n \tag{24}$$

At this point, when the AIS moves away from the center of the circle, the deviation in the latitude and longitude coordinate system becomes:

$$\begin{cases} \delta lat = 2r \cdot (\sin(\theta + \delta\theta) - \sin\theta) \\ \delta lon = 2r \cdot (\cos(\theta + \delta\theta) - \cos\theta) \end{cases} \tag{25}$$

Here, part of the additive error in angle measurement becomes a multiplicative error. Similarly, when there is additive error in latitude and longitude, this error transforms into a mixed error upon conversion to the range-bearing coordinate system. Secondly, this paper assumes that noise follows a normal distribution. However, due to coordinate transformations, this assumption may not hold for either the radar or latitude–longitude coordinate systems.

Furthermore, when using higher-dimensional spaces include COG and v , the errors may become increasingly complex

4.3. Directions for Improvement

- Multi-Track Fitting

Currently, in the case of multiple trajectories, interpolation can only be performed using latitude and longitude due to the trade-off between memory usage and computational speed, as mentioned in Section 2.1:

If a probability cloud is constructed only for the plots near each AIS trajectory, memory consumption would be significantly reduced. However, for each AIS trajectory, the probability cloud needs to be recalculated, which would greatly slow down the computation time.

On the other hand, if a probability cloud is generated for all radar plots at once and then superimposed based on the deviation of each AIS, the computational speed would be greatly improved. In this case, only one pass through the probability cloud with different AIS offsets is needed to complete the matching of all AIS and radar trajectories. However, the memory consumption would be extremely high.

The introduction of additional parameters, however, can significantly improve the matching accuracy. Therefore, it is necessary to research a method to minimize memory consumption.

- Track Interruption

In cases where some points are missing, the method used in this paper is to directly interpolate to fill in the gaps. However, this actually introduces a relatively large error compared to the original AIS data.

However, if the filter threshold is adjusted to find new clusters when some points are missing, the time complexity will increase to α times the original:

$$\alpha = \sum_{i=1}^a C_k^i O(nk) \quad (26)$$

Here, k is the number of radar point batches used for association, and α is the maximum number of missing points allowed. The time cost is too high, so a compromise needs to be found to achieve a balance between complexity and accuracy.

- Extraction of Parameter

As mentioned earlier, the experimental results presented in this paper (see Figure 25) indicate that two clusters were incorrectly divided due to the selection of the clustering threshold. While the approach employed a larger number of clusters to mitigate this issue, excessive clustering can lead to centroid displacement, resulting in errors in parameters such as bias and variance.

To address these challenges, further research will focus on modifying the clustering algorithm or exploring alternative methods for scaling the feature space. For instance, applying the algorithm in range-bearing space may yield more accurate parameter extraction, as radar errors tend to be minimized in this domain.

5. Conclusions

This paper presents a novel method for efficiently and accurately matching target AIS trajectories with corresponding tracks from multiple batches of radar point data in

complex environments. The simulation data and experimental results demonstrate that the proposed algorithm effectively extracts trajectories from radar point data, achieving a high level of accuracy. However, the algorithm's stability, particularly in real-time applications, warrants further enhancement. Future work will focus on optimizing the stability and robustness of the algorithm, enabling its practical deployment in dynamic environments.

Author Contributions: Writing—original draft, J.X.; Validation, Y.J.; Formal analysis, Y.S.; Writing—review & editing, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62031014. The APC was funded by the National Natural Science Foundation of China, grant number 62031014.

Data Availability Statement: The data presented in this study are not publicly available due to confidentiality restrictions. Requests for access to the data should be directed to the corresponding author.

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

References

1. Wang, C.M.; Li, Y.; Min, L.; Chen, J.; Lin, Z.; Su, S.; Zhang, Y.; Chen, Q.; Chen, Y.; Duan, X.; et al. Intelligent marine area supervision based on AIS and radar fusion. *Ocean. Eng.* **2023**, *285*, 115373. [[CrossRef](#)]
2. Du, L.; Goerlandt, F.; Kujala, P. Review and analysis of methods for assessing maritime waterway risk based on non-accident critical events detected from AIS data. *Reliab. Eng. Syst. Saf.* **2020**, *200*, 106933. [[CrossRef](#)]
3. Qu, X.; Meng, Q.; Suyi, L. Ship collision risk assessment for the Singapore Strait. *Accid. Anal. Prev.* **2011**, *43*, 2030–2036. [[CrossRef](#)] [[PubMed](#)]
4. Wang, J.; Qi, L.; Wang, W.; Lei, F.; Zhu, H. Research and Implementation of AIS and Radar Information Fusion Method. In Proceedings of the 2022 6th Annual International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 14–18 October 2022; pp. 532–539.
5. Last, P.; Bahlke, C.; Hering-Bertram, M.; Linsen, L. Comprehensive Analysis of Automatic Identification System (AIS) Data in Regard to Vessel Movement Prediction. *J. Navig.* **2014**, *67*, 791–809. [[CrossRef](#)]
6. Norris, A. AIS Implementation—Success or Failure? *J. Navig.* **2007**, *60*, 1–10. [[CrossRef](#)]
7. Last, P.; Hering-Bertram, M.; Linsen, L. How automatic identification system (AIS) antenna setup affects AIS signal quality. *Ocean. Eng.* **2015**, *100*, 83–89. [[CrossRef](#)]
8. Lei, J.; Sun, Y.; Wu, Y.; Zheng, F.; He, W.; Liu, X. Association of AIS and Radar Data in Intelligent Navigation in Inland Waterways Based on Trajectory Characteristics. *J. Mar. Sci. Eng.* **2024**, *12*, 890. [[CrossRef](#)]
9. Katsilieris, F.; Braca, P.; Coraluppi, S. Detection of malicious AIS position spoofing by exploiting radar information. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July 2013; pp. 1196–1203.
10. Habtemariam, B.; Tharmarasa, R.; McDonald, M.; Kirubarajan, T. Measurement level AIS/radar fusion. *Signal Process.* **2015**, *106*, 348–357. [[CrossRef](#)]
11. Lin, C.; Dong, F.; Hai, L.; Le, L.; Zhou, J.; Ou, Y. AIS information decoding and fuzzy fusion processing with marine radar. In Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 12–14 October 2008; pp. 1–5.
12. Liu, C.; Lin, B.; Liu, X.-M.; Suo, J.-D.; Liu, R.-J. Fuzzy correlation algorithm for multi-target fusion of automatic identification system and radar. *J. Comput. Theor. Nanosci.* **2013**, *10*, 2826–2830. [[CrossRef](#)]
13. Zhang, H.; Liu, Y.; Ji, Y.; Wang, L.; Zhang, J. Multi-feature maximum likelihood association with space-borne SAR, HFSWR and AIS. *J. Navig.* **2017**, *70*, 359–378. [[CrossRef](#)]
14. Ballard, D. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit.* **1981**, *13*, 111–122. [[CrossRef](#)]
15. Carlson, B.; Evans, E.; Wilson, S. Search radar detection and track with the Hough transform. I. system concept. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 102–108. [[CrossRef](#)]
16. Chen, D.; Chen, P.; Zhou, C. Research on AIS and Radar Information Fusion Method Based on Distributed Kalman. In Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS), Liverpool, UK, 14–17 July 2019; pp. 1482–1486.
17. Wen, J.; Yu, G.; Gao, L.; Liu, Y.; Nie, X. HFSWR ship trajectory tracking and fusion with AIS using Kalman filter. In Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 456–461.
18. Shoemake, K. Animating rotation with quaternion curves. In Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, San Francisco, CA, USA, 22–26 July 1985; pp. 245–254.
19. Catmull, E.; Rom, R. A class of local interpolating splines. In *Computer Aided Geometric Design*; Barnhill, R.E., Riesenfeld, R.F., Eds.; Academic Press: Cambridge, MA, USA, 1974; pp. 317–326.
20. Harati-Mokhtari, A.; Wall, A.; Brooks, P.; Wang, J. Automatic Identification System (AIS): Data Reliability and Human Error Implications. *J. Navig.* **2007**, *60*, 373–389. [[CrossRef](#)]

21. Sang, L.-Z.; Wall, A.; Mao, Z.; Yan, X.-P.; Wang, J. A novel method for restoring the trajectory of the inland waterway ship by using AIS data. *Ocean. Eng.* **2015**, *110*, 183–194. [[CrossRef](#)]
22. Zhen, R.; Pan, J.; Shao, Z.-P. Advance in Character Mining and Prediction of Ship Behavior based on AIS Data. *J. Geo-Inf. Sci.* **2021**, *23*, 2111–2127. [[CrossRef](#)]
23. Nguyen, V.-S.; Im, N.-K.; Lee, S.-M. The Interpolation Method for the missing AIS Data of Ship. *J. Navig. Port. Res.* **2015**, *39*, 377–384. [[CrossRef](#)]
24. Chen, X.; Yang, J. Analysis of the uncertainty of the AIS-based bottom-up approach for estimating ship emissions. *Mar. Pollut. Bull.* **2024**, *199*, 115968. [[CrossRef](#)]
25. Vincenty, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Surv. Rev.* **1975**, *23*, 88–93. [[CrossRef](#)]
26. Kremer, V.E. Quaternions and SLERP. In Proceedings of the Embots. dfki. de/doc/seminar ca/Kremer Quaternions. pdf. 2008. Available online: https://d1wqtxts1xzle7.cloudfront.net/47081754/lerp_slerp_nlerp-libre.pdf?1467902732=&response-content-disposition=inline;+filename=Quaternions_and_SLERP.pdf&Expires=1732248355&Signature=PukXdhFoxlxylwLaCCBewIFtMwL o-CGJrH Fzqp6fJTcFiDAyIV5AEgEOyIKruI2jsCqbmVPUz7B4p-h1cilHq M6RiDiJnujKQHm7LvIX3AmmMijt5SJHUwenDhPwbeE giSSXojwNPSIWuVxmKA033wLGk2CMahbs4VoWdg4JdEkPd-xjFzzVP2JBRQu9DyvNNEJJuoguGPDe80j-YFpy 1P-UMoK6txIb1 5um2Ivffb5QaxzNGRYfwbNLKhJdfXi0HK2sxYN0X8FXgf5gH4q3YvDbJVAfHMnvcyUAG2XUfCVkpWkGh4eckzWxuZx7y eRzw fZqXioSG1Cpt9LIOW_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (accessed on 18 November 2024).
27. Kazimierski, W.; Stateczny, A. Radar and Automatic Identification System Track Fusion in an Electronic Chart Display and Information System. *J. Navig.* **2015**, *68*, 1141–1154. [[CrossRef](#)]
28. Yan, J. Track Initiation of Radar Detection Data Points Based on a Single Platform. Master’s Thesis, Jiangnan University, Wuxi, China, 2011. (In Chinese with English abstract).
29. Hough, P.V. Method and Means for Recognizing Complex Patterns. U.S. Patent 3,069,654, 18 December 1962.
30. Mukhopadhyay, P.; Chaudhuri, B.B. A survey of Hough Transform. *Pattern Recognit.* **2015**, *48*, 993–1010. [[CrossRef](#)]
31. Abellanas, M.; Hurtado, F.; Icking, C.; Klein, R.; Langetepe, E.; Ma, L.; Palop, B.; Sacristán, V. Smallest Color-Spanning Objects. In *Algorithms—ESA 2001: 9th Annual European Symposium Århus, Denmark, 28–31 August 2001 Proceedings 9*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 278–289.
32. Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.H.; Li, Z.; Duerig, T. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. *arXiv* **2021**, arXiv:2102.05918. [[CrossRef](#)]
33. Fix, E.; Hodges, J.L., Jr. Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. *Int. Stat. Rev.* **1989**, *57*, 238–247. [[CrossRef](#)]
34. Jiang, B.; Sun, L.; Zhou, W.; Guan, J.; He, Y. A multi-target joint estimation method for radar calibration based on real-time AIS data. In Proceedings of the 2016 CIE International Conference on Radar (RADAR), Guangzhou, China, 10–13 October 2016; pp. 1–5.
35. Kotsakis, C. Spatial coordinate transformations with noisy data. In *Geospatial Analyses of Earth Observation (EO) Data*; IntechOpen: London, UK, 2019.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.