Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 8831371, 18 pages
https://doi.org/10.1155/2023/8831371

Review Article

Vessel Trajectory Data Compression Algorithm considering Critical Region Identification

Xinliang Zhang and Shibo Zhou

College of Navigation, Jimei University, Xiamen 361021, China

Correspondence should be addressed to Shibo Zhou; zhoushibo@jmu.edu.cn

Received 18 July 2023; Revised 15 November 2023; Accepted 6 December 2023; Published 30 December 2023

Academic Editor: Zhenning Li

Vessel trajectory data are currently the most important data source for vessel trajectory data mining research. However, vessel AIS data have a short sampling time interval and a large amount of data redundancy, which hampers the efficient utilization of AIS data. In order to effectively remove redundant information from AIS data and improve its usage efficiency, a compression algorithm for vessel trajectory data compression algorithm considering critical region identification (VATDC_CCRI) is proposed. The VATDC_CCRI algorithm identifies the critical regions of a vessel's trajectory by analyzing the distribution of node variation rates. It employs the Douglas–Peucker (DP) algorithm to compress the data in these critical regions, reducing the distortion of the trajectory after compression. Additionally, the algorithm utilizes a sliding window approach to process the initial trajectory to improve the quality of the compressed vessel trajectories and retain as many spatiotemporal characteristics of the original trajectories as possible. It combines the feature nodes from the crucial regions in the vessel's trajectory with the results obtained from the sliding window algorithm, effectively compressing the vessel's trajectory. Experiments conducted on individual and multiple trajectories demonstrate that the VATDC_CCRI algorithm achieves higher compression rates and exhibits faster processing speeds compared to other classical vessel trajectory compression algorithms while preserving the shape of the vessel's trajectory significantly.

1. Introduction

Due to the relevant requirements of the SOLAS Convention 2002 amendment, vessels engaged in international voyages are currently equipped with AIS equipment [1]. Access to vessel trajectory data is becoming more and more convenient, and the application of vessel AIS data in traffic network research [2], vessel trajectory prediction [3], route planning [4], and abnormal vessel behavior detection [5] is becoming more and more widespread. However, vessel trajectory data have a high sampling frequency, increasing data volume. This requires significant storage space and consumes more computational power during data processing [6]. Therefore, in practical applications, AIS trajectory data compression is one of the essential preprocessing steps for vessel data. However, during data compression, while eliminating redundant data, it is inevitable to lose some critical information, which can have an impact on subsequent research. Therefore, the efficient compression of vessel trajectory data while minimizing distortion is a research focus in the maritime field. Vessel trajectory data compression can be broadly classified into two categories: offline compression algorithms with the Douglas–Peucker (DP) algorithm as the core and online compression algorithms with the sliding window algorithm as the core [7].

The DP algorithm [8] is a classic vector data compression algorithm proposed by Douglas and Peucker in 1973. This algorithm is known for its ability to preserve the shape characteristics of vector data to the maximum extent [9]. However, the DP algorithm is limited when compressing vessel trajectory data. These limitations include difficulties in determining the distance threshold, high algorithmic time complexity, and compression solely based on the shape characteristics of the trajectory.

In order to solve the problem that the input parameters of the DP algorithm are difficult to determine, some researchers have directly used evaluation metrics to judge the compression results and then determined the appropriate parameters [10–13]. Zhang et al. [10] carried out research on the evaluation method of the smallest vessel field based on AIS data, calculated the relative distances of 962 vessels in the Qiongzhou Strait, drew the scatter distribution map of the center vessel to other vessels, and determined and verified the DP. The optimal compression effect can be obtained when the algorithm is 0.8 times the length of the vessel. However, the method is not generalizable, and the effect is not ideal when used in waters outside the Qiongzhou Strait. Liu et al. [11] proposed an adaptive threshold DP algorithm for batch processing trajectory data. They transformed the circular neighborhood of the DBSCAN algorithm into a square and calculated the similarity between trajectories. In this approach, it is only necessary to determine the distance threshold for one trajectory, and based on the similarity, thresholds for other trajectories can be generated. This enables the generation of different thresholds for different trajectories. However, this method is only suitable for large-scale data processing and does not achieve high compression quality. Gao et al. [13] proposed a trajectory data compression algorithm based on the vessel's sailing state as well as acceleration change, which determines the vessel's sailing state according to the sailing speed and then adaptively determines the threshold of DP algorithm by using the acceleration change function, which obtains a better compression effect. It is one of the compression algorithms that have the slightest increase in time complexity on the basis of realizing the parameter adaption at present.

The DP algorithm, used for trajectory compression, involves the repetitive calculation of the Euclidean distance between trajectory points, resulting in high time complexity and low computational efficiency. In response to this issue, several researchers have proposed improvements [14–18]. Zhao and Shi [14] proposed an improved vessel trajectory compression algorithm by sliding window algorithm, taking five trajectory points as a window, calculating the course direction change within the window, determining the critical nodes of the trajectory data given the change threshold, dividing the trajectory into two parts by the nodes into straight and curved parts, and then performing trajectory compression for these two parts, respectively, with 0.8 times captain as the threshold of DP algorithm. The experimental results show that this improved algorithm can shorten the running time by nearly 50%. Huang et al. [17] proposed a DP algorithm that utilizes parallel computation on GPUs, thereby achieving faster execution time. This approach is suitable for processing large volumes of vessel trajectory data. However, it has higher hardware requirements.

Although the problem of low threshold input and low operating efficiency of the DP algorithm has been solved, it can only ensure the similarity of the trajectory shape when performing trajectory compression, ignoring the index of the time dimension, and the reliability of the compressed data is low. Some scholars have attempted to consider additional information beyond vessel position data during the compression process [19–22]. Shi and Liu [19] proposed a multifactor DP algorithm that considers speed variations in AIS data and sets a turning threshold. They retain trajectory points with speeds exceeding a certain threshold or turning angles exceeding the threshold, resulting in better compression results. However, this approach introduces more threshold parameters and increases the time complexity on top of the DP algorithm. Building on this, Zhou et al. [20] proposed a multiobjective peak-based DP algorithm. They incorporated peak sampling strategies during the compression process, considering spatial characteristics, heading, and speed information of vessel trajectories. They also introduced multiobjective optimization, obstacle detection, and overlap region determination mechanisms, ensuring high compression rates while enhancing trajectory integrity. However, as these mechanisms are added, the algorithm's time complexity increases significantly. Wei et al. [21] combined the DP algorithm with the sliding window algorithm and proposed an AIS trajectory compression algorithm that considers vessel behavior. In this approach, the DP algorithm uses 0.8 times the vessel length as the distance threshold, and the sliding window has a classic size of three trajectory points. The window threshold, chosen based on statistical theory, is set at 1.6 standard deviations of indicator changes. Finally, the results of both algorithms are merged. Experimental results demonstrate that this method considers vessel behavior and exhibits good overall performance but has a higher time complexity. Yan et al. [22] proposed a vessel trajectory denoising and compression algorithm based on statistical theory and sliding window, which can recognize and retain the burrs in the original trajectory, which prevents the vessel trajectory from crossing the land after compression, but the algorithm lacks flexibility. The compression effect will be reduced when the trajectory is close to the coastline.

The offline compression algorithm can maintain the shape of the trajectory before and after compression but ignores some critical information in the time dimension, such as the speed and course mutation of the vessel, which are of great significance. Vessel speed information helps judge the vessel type, and sudden changes in course can be used as the basis for channel planning. Therefore, the researchers proposed an online compression algorithm with the sliding window as the core, which can detect various indicators of vessel operation in real time, and then extract critical points according to the degree of change of the indicators in the window, and adopt the idea of gradual compression. The streaming form performs online compression for the transmitted data [23]. However, the sliding window algorithm also has the problems that the compression result is sensitive to the input parameters and the compression quality is low.

In order to reduce the influence of input parameters, researchers have determined parameters adaptively [24, 25] or modified the input parameters to a specified range [26, 27]. Gao and Shi [24] proposed a key feature based on a vessel spatio-temporal data Point extraction algorithm, set the distortion index to judge the extraction effect of feature points, and then selected the appropriate threshold according to to the spatio-temporal data point extraction algorithm. The algorithm can achieve a compression rate close to that of the DP algorithm, but the threshold determination process requires multiple experiments. The threshold determination process requires multiple experiments. Zhang et al. [26] proposed an online multidimensional simplification algorithm, which comprehensively considered multiple vessel indicators, including the position, speed, and direction changes of trajectory points, and determined a threshold variation range by statistical methods to extract trajectory points. Finally, the Ningbo vessel trajectory data of the port verify the algorithm's effectiveness, but there is a problem with the low compression rate.

In addition to the fact that the compression results are greatly affected by parameters, the sliding window algorithm only considers the operating mechanism of the trajectory points in the window, which may result in a greater degree of trajectory deformation before and after compression. Many researchers have improved this problem [28–31]. Sánchez-Heres and Sánchez [28] proposed a trajectory simplification algorithm based on behavior recognition for an equivalent passage plan (EPP). First, the behavior of the vessel was divided into three types: stop, sail, and turn, and then, the trajectory data corresponded to these three behaviors. First, the vessel's behaviour is classified into three states: stopping, sailing, and turning, then the trajectory data is mapped to these three states, and finally, the trajectory data in the sailing and stopping states are compressed by a sliding window algorithm. The EPP algorithm can ultimately preserve the extraordinary trajectory changes of the vessel's navigation, but the algorithm has a high time complexity and a low compression rate. Zhong et al. [30] proposed a data compression algorithm based on the spatiotemporal characteristics of trajectory data (CASC). The algorithm takes azimuth difference, velocity difference, and time interval as input thresholds and calculates the azimuth difference and velocity difference of trajectory points within the input time interval. Retain the trajectory points with azimuthal and velocity differences exceeding a threshold value. This algorithm improves the utilization rate of trajectory information and obtains a better compression effect but increases the input parameters. Han et al. [31] proposed a pattern-accumulated compression (PAC) algorithm, which divides the compression process into two parts. First, a sliding window algorithm divides the initial trajectory into a series of spatial and velocity components. Then, the original trajectory is described by selecting representative components to obtain the effect of compression, which can preserve the trajectory information more thoroughly. However, when the study area changes, the pattern map needs to be reconstructed, which can significantly increase the workload of the study.

In summary, the offline compression algorithm can consider the shape change of vessel trajectory, standardize all the trajectory points, and then segment them according to the input threshold, which can maximally retain the shape characteristics of the trajectory, but due to the consideration of spatial feature changes only, the utilization rate of trajectory information is low, and it is easy to ignore the time characteristics of the vessel during navigation, which ultimately reduces the reliability of the compressed data. In contrast, the online compression algorithm is able to consider a variety of factors during the vessel's traveling process, including angular velocity and speed changes, and retains the trajectory points that undergo significant changes during the vessel's traveling process by means of the input threshold value. This algorithm fully considers the temporal characteristics of the trajectory data, but the compression effect is not good in some trajectory data with minor changes in angular velocity and speed, resulting in a large gap between the shapes before and after compression. Based on this, this paper proposes a vessel trajectory data compression algorithm taking into account the critical region identification, which extracts the regions with significant changes in vessel nodes by evaluating the degree of changes in the trajectory nodes, then compresses the data in the crucial subregions using the DP algorithm to try to retain the shape of the critical regions, and, on this basis, integrates the results of the simplification of the initial vessel trajectory data by sliding window. It realizes to improve the compression rate while retaining the shape feature points of the vessel trajectory.

2. Related Work

2.1. Evaluation of Trajectory Node Variability. The visual representation of the vessel trajectory is a vector line. Each node on the trajectory has different importance in displaying the trajectory graph. For some critical turning points in the trajectory graph, the shape of the trajectory may be changed if deleted during the trajectory compression process. To maximise the original shape of the trajectory during the trajectory compression process, these critical nodes must be retained. This paper uses the method of node change degree evaluation to distinguish the importance of each node of the vessel trajectory to identify the critical regions of the vessel trajectory. The degree of change of the trajectory point is the ratio of the distance from the middle vertex to the connecting line between its front and rear vertices and the length of the connecting line, and the greater the degree of change, the more significant the change of the point is proved to be. In order to describe the method of evaluating the degree of change in detail, this paper takes five trajectory points of $\{T_1, T_2, T_3, T_4, T_5\}$ as an example, and the structure is shown in Figure 1. First, the distance Distance (T3) between the adjacent points T2 and T4 of T3 is calculated, and the vertical line passing through the intersection line segment $\overline{T_2T_4}$ of T3 is found. The vertical point is T_3'. At this time, vertical (T3) is the length of the line segment $\overline{T_3T_3'}$. The calculation method of the change degree is shown in equation (1), and the detailed information on the node change degree evaluation is shown in Algorithm1.

$$\text{Degree}\left(T_i\right) = \frac{\text{Vertical}\left(T_i\right)}{\text{Distance}\left(T_i\right)}. \tag{1}$$

2.2. Douglas–Peucker (DP) Algorithm. Conventional implementations of the Douglas–Peucker algorithm in single-core computers usually use serial algorithms. This can be implemented in either a recursive or a nonrecursive form. The nonrecursive form differs from the recursive form only

```
Input: Ship latitude and longitude coordinates information Point.
Output: Node Importance NI.
$N$ = Len(Point); NI = list();
for $i = 1$ to $N - 1$ then;
    Calculate the distance $D$ between Point($i - 1$) and Point($i$+1);
    Calculate the vertical distance $V$ of Point($i$);
    $I = V/D$
    NI.append(I)
end for
return NI
```

Algorithm 1: Node importance evaluation algorithm.

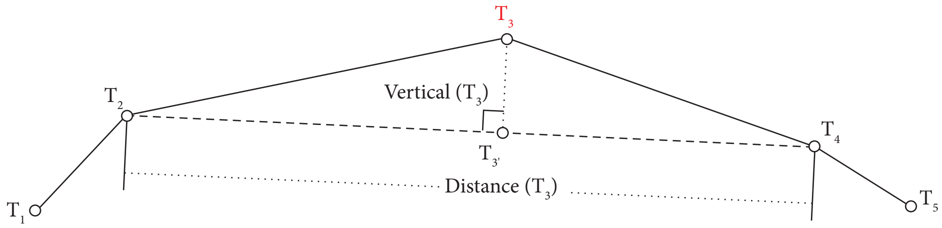


Figure 1: Schematic diagram of the evaluation of the degree of change of nodes.

in that it requires a stack space to hold the starting and ending vertices of the vector line that is broken into two segments each time the farthest away vertex is found, thus converting the recursion into a loop. Each time the loop takes the starting and ending vertices of a segment off the stack, a segment is generated connecting the two vertices, and the distance between the remaining intermediate points to the segment is calculated. If the stack is emptied, then all segments have been processed, and the loop can then be terminated to obtain the desired sequence of selected vertices [32]. Assume that the current AIS trajectory is set D, which contains 14 trajectory points $\{T_1, \cdots\cdots, T_{14}\}$. The schematic diagram of the simplification process of the Douglas–Peucker (DP) algorithm is shown in Figure 2, where $d_{\max}$ represents the maximum distance from the trajectory node to the head-to-tail connection, and threshold represents the input threshold. When $d_{\max} \geq$ threshold, the point is retained and the trajectory is divided into two parts, and then $d_{\max}$is calculated and threshold comparison of the two parts of the trajectory is done; when $d_{\max} <$ threshold, any points in this part of the trajectory are no longer retained and the comparison of $d_{\max}$ and threshold is repeated until $d_{\max} <$ threshold of all trajectories. Stop simplifying, and output dataset $\{T_1, T_3, T_6, T_9, T_{11}, T_{14}\}$. The details of the DP algorithm are shown in Algorithm 2.

2.3. Sliding Window Algorithm. Vessel trajectory data are a kind of time series data, including time stamp, course, speed, position, angular velocity of turning head, and other information that characterizes the vessel's motion state and maneuvering process [33]. Traditional trajectory compression only considers position changes, and the utilization rate of trajectory information is low. To solve this problem, more and more researchers have applied sliding window-type compression algorithms to the field of vessel trajectory data compression [34]. The classical sliding window-type trajectory compression algorithm uses the idea of segmented stepwise compression. After determining the initial window, the data transmitted synchronously is compressed in real time according to the characteristics and properties of the trajectory points. The classical sliding window-type trajectory compression algorithm uses the idea of segmented stepwise compression. After determining the initial window, the data transmitted synchronously is compressed in real time according to the characteristics and properties of the trajectory points. Compared with the offline algorithm, this algorithm does not need to specify the end point of the trajectory segment, which can effectively improve processing efficiency. As shown in Figure 3, the initial window is set as {T1, T2, T3}, in which T1 is the starting point, T2 is the point to be compressed, and T3 is the endpoint. We calculate the angle of deflection of the line from T1 and T2 to T2 and T3 and compare it with the set angle threshold. If the deflection angle of the line between T2 and T3 to T1 and T4 is greater than the threshold value, T3 is retained, and the window is slid backwards, and T3 is used as the starting point of the sliding window, which is updated to {T3, T4, T5}. We continue to make the same judgment as above for subsequent trajectory points until the final issue of the trajectory segment updates. The details of the classical sliding window algorithm are shown in Algorithm 3.

3. Vessel Trajectory Data Compression Algorithm considering Critical Region Identification

The vessel trajectory data compression algorithm considering critical region identification (VATDC_CCRI) is

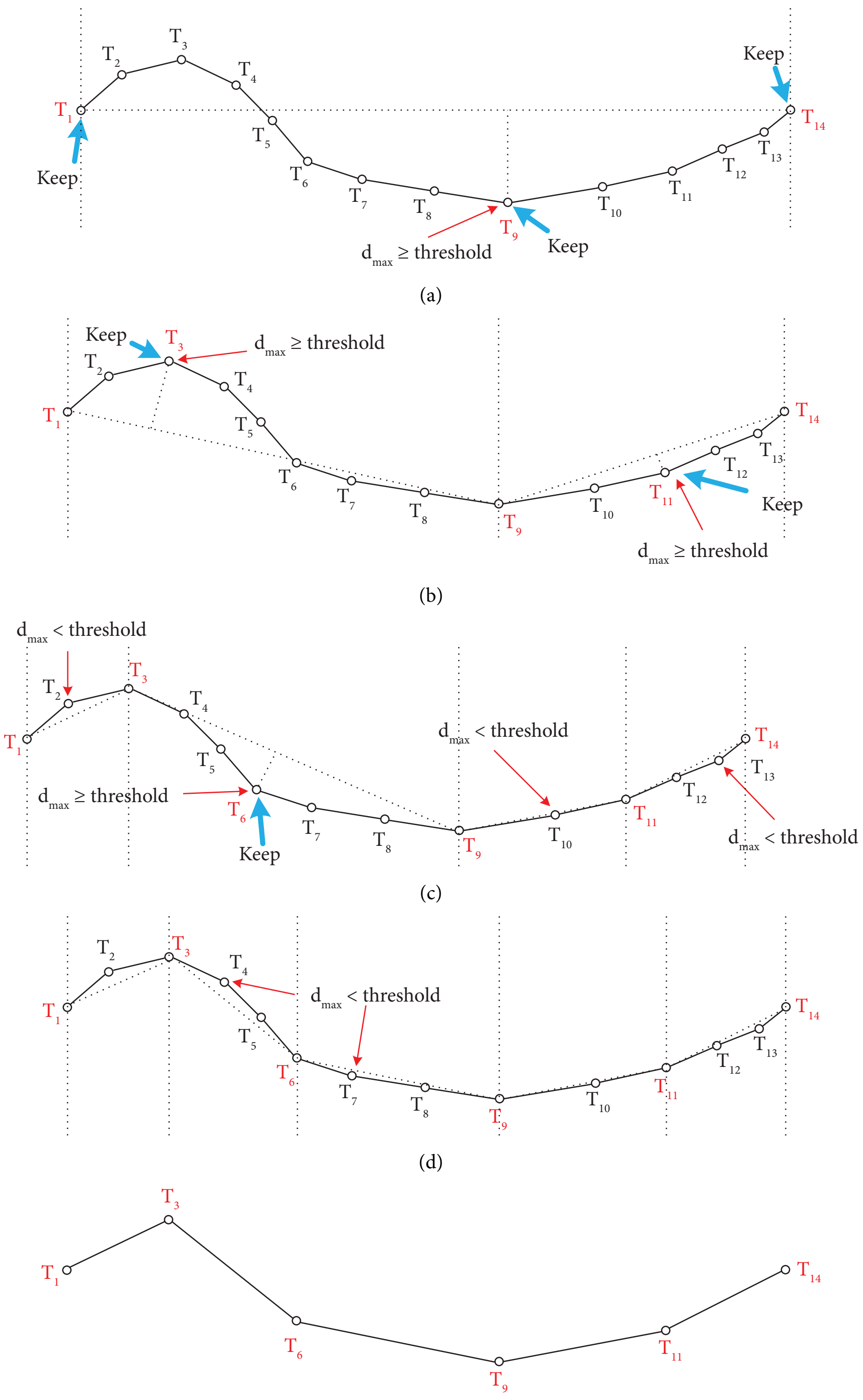


Figure 2: Schematic diagram of the DP algorithm trajectory simplification process. (a) Step 1: find the maximum distance. (b) Step 2: retains the maximum distance point and divides the trajectory into two parts. (c) Step 3: part of the trajectory stops iterating when the maximum value is less than the threshold. (d) Step 4: repeat iteration step 1, step 2, and step 3. (e) Step 5: iterations complete and get the compressed trajectory.

proposed in this paper to preserve the essential data nodes in the trajectory when compressing the vessel trajectory data, reduce the distortion of the course after compression, and improve the operation speed of the trajectory data compression algorithm. When VATDC_CCRI compresses the vessel trajectory data, it fully considers critical nodes in the vessel trajectory, such as critical information such as turning points, sudden changes in course, and abnormal angular velocity of the turning head. The relatively essential trajectory nodes are determined by evaluating the change

```
Input: Ship latitude and longitude coordinates information Point.
    Distance Threshold $T$
Output: Compressed ship latitude and longitude information Pointset.
Pointset = list()
Calculate the list of distances from the point to the first and last line $d_{\text{list}}$
Get the maximum value of the list species $d_{\max}$ and the maximum position index
If $\max(d_{\text{list}}) < T$ then
    Return to the first and last point
Else
L-Point = DP(Point(0: index), T)
R-Point = DP(Point(index: end), T)
Pointset. append(L-Point)
Pointset. append(R-Point)
end if
return Pointset
```

Algorithm 2: Douglas–Peucker algorithm.

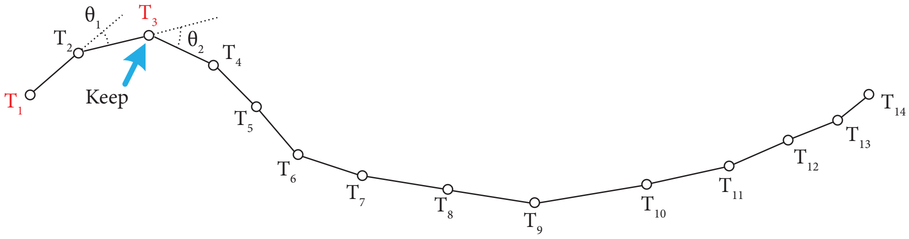


Figure 3: Schematic diagram of the classical sliding window algorithm.

```
Input: Ship latitude and longitude coordinates information Points.
    Vessel speed information {Speed}.
Output: Compressed ship latitude and longitude information $\text{DataSet}_{\text{Slide-Window}}$.
CV = list();
$N$ = Len(Point);
for $i = 1$ to $N - 1$ do
    v1 = Point($i + 1$)-Point($i$)
    v2 = Point($i + 2$)-Point($i + 1$)
        $t$ = arccos(Paradigm(v1, v2)/(|v1| * |v2|))
CV = [CV; $t$]
end for;
    $\{\mu_\theta, \sigma_\theta\}$ = Parameter_Estimation(CV);
    $\{\mu_v, \sigma_v\}$ = Parameter_Estimation(Speed);
    $\text{DataSet}_{\text{Slide-Window}}$ = [];
for $i = 0$ to N − 2 do
    if $|\text{Angle}(i) - \mu_\theta| > 1.6\sigma_\theta$ or $|Vs(i) - \mu_v| > 1.6\sigma_v$ then
            $\text{DataSet}_{\text{Slide-Window}}$ = [$\text{DataSet}_{\text{Slide-Window}}$; Points($i$+1)]
        end if
end for
return $\text{DataSet}_{\text{Slide-Window}}$
```

Algorithm 3: Sliding window algorithm.

degree of the trajectory points in the vessel's trajectory. Then, the critical regions of the trajectory are determined by the statistical distribution method. Finally, the DP algorithm extracts the features of the essential sites of the original course. By identifying critical regions, the target of the DP algorithm is transferred from a single trajectory to multiple crucial regions on the circuit, which significantly improves the running speed of the algorithm. To enhance the quality of the compressed course and retain the spatiotemporal characteristics of the original trajectory, on the basis of the critical region trajectory processing, the sliding window algorithm is used to process the initial course [21].

Ultimately, the feature nodes of critical regions in the vessel's trajectory are fused with the results extracted by the sliding window algorithm. The VATDC_CCRI algorithm proposed in this paper realizes the effective compression of the vessel's trajectory on the basis of retaining the shape of the vessel's trajectory to the greatest extent. Figure 4 describes the algorithm flow of VATDC_CCRI, and Algorithm 4 describes the implementation process of VATDC_CCRI.

In Figure 4, IP represents the extracted trajectory points in the important region, SOG represents the ship-to-ground speed, COG represents the ship-to-ground heading, and PART represents the trajectory dataset of the original trajectory in the key region. To describe the execution process of the VATDC_CCRI algorithm, the AIS trajectory data of a vessel with MMSI 210698000 on 13–15 April 2019 are used as an example for detailed illustration, and the shape of the vessel's trajectory is shown in Figure 5. The geographic location data involved are adjusted using the Mercator projection [35], and equations (2)–(5) give the calculation process of the Mercator projection.

$$r = \frac{acos\phi_0}{\sqrt{1 - e^2 \sin^2 \phi_0}}, \quad (2)$$

$$q = \ln\left(\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\left(\frac{1 - e\sin\varphi}{1 + e\sin\varphi}\right)^{e/2} \right), \quad (3)$$

$$x = r \cdot \lambda, \quad (4)$$

$$y = r \cdot q, \quad (5)$$

(λ, φ) in equations (3) and (4) represents the latitude and longitude data of the acquired geographical coordinates, ϕ_0 represents the standard latitude in the Mercator projection, and (x, y) represents the adjusted coordinate data.

3.1. Identification of Critical Region. As the sliding window algorithm can retain the vessel's heading and speed information, the evaluation of the degree of change of the vessel's trajectory points involves only spatial position operations. The degree of change of trajectory points is generally higher for the critical regions that maintain the trajectory shape. According to the trajectory node variation evaluation method proposed in Section 2.1, the variation evaluation result curve of vessel trajectory points is shown in Figure 6(a). In Figure 6(a), each node has its corresponding degree of change, and the value fluctuates wildly, so it is impossible to divide critical regions directly according to the degree of change. Therefore, to eliminate the influence of this volatility, this paper counts the distribution of the degree of change of trajectory points, sets a threshold to judge whether all trajectory points are essential, assigns important trajectory points to the same level, and discards unimportant trajectory points. In the process of single-dimensional data processing, the box plot in the statistical distribution method is often used to describe, as shown in Figure 6(b), and the specific position of the box plot is marked. Therefore, this article discusses the commonly used average and the upper, median, and lower quartile involved in the box plot as thresholds. Figure 7 shows the judgment of critical trajectory points under different thresholds. The red " $*$ " in the figure indicates the important trajectory points filtered out, while the blue "*O*" indicates the trajectory points with a small degree of change.

Figure 8 demonstrates the judgment of important trajectory points under different thresholds, from which it can be seen that using the upper quartile can maximise the classification of trajectory points, and the final result can show more obvious regional characteristics. While other thresholds are conditions for judgment, the number of retained trajectory points is too much. Although the initial judgment of trajectory points is achieved, the number of essential trajectory points is small and has temporal characteristics, which clustering methods cannot classify. In order to identify important areas, important trajectory points that are close in spatial and temporal distance must first be classified. In this paper, after calculating the distances of neighboring important trajectory points separately according to the temporal order of trajectory points, the outliers in the box plot are calculated based on the distance distribution. The outliers represent the distances between two neighboring trajectory points significantly beyond other neighboring points. Therefore, we use the outliers to divide all the extracted data points into six regions $\{S_1, S_2, S_3, S_4, S_5, S_6\}$, as shown in Figure 8, and the number of critical trajectory points in each region is $\{1, 2, 2, 8, 119, 14\}$. However, there are some regions with fewer important trajectory points, and these regions only have partial location mutations, so only essential trajectory points can be retained. In this paper, through several experiments, it is determined that it is most reasonable when using 10 important trajectory points as the threshold value. Therefore, the critical region of the trajectory is $\{S_5, S_6\}$.

3.2. Trajectory Simplification. The efficiency of the compression algorithm determines whether the algorithm can be applied to large volumes of vessel trajectory data, while the degree to which the compression algorithm retains valid information in the vessel trajectory data determines whether the compressed data can be used for further analysis, where the degree to which valid information is retained is also the confidence level of the compressed data. Most modern compression algorithms are unable to balance operational efficiency with confidence, as the only way to improve confidence is to traverse the original trajectory multiple times to retain as many valid points as possible, while multiple iterations reduce the efficiency of the algorithm on a single trajectory and are not suitable for compression of large amounts of data. The information in vessel trajectory data includes both spatial and temporal latitude. Therefore, this section is divided into two main parts. One part is to compress the spatial features of the trajectory according to the critical region in Section 3.1. The other part is to extract the temporal features of the trajectory and finally unify the results of the two parts.

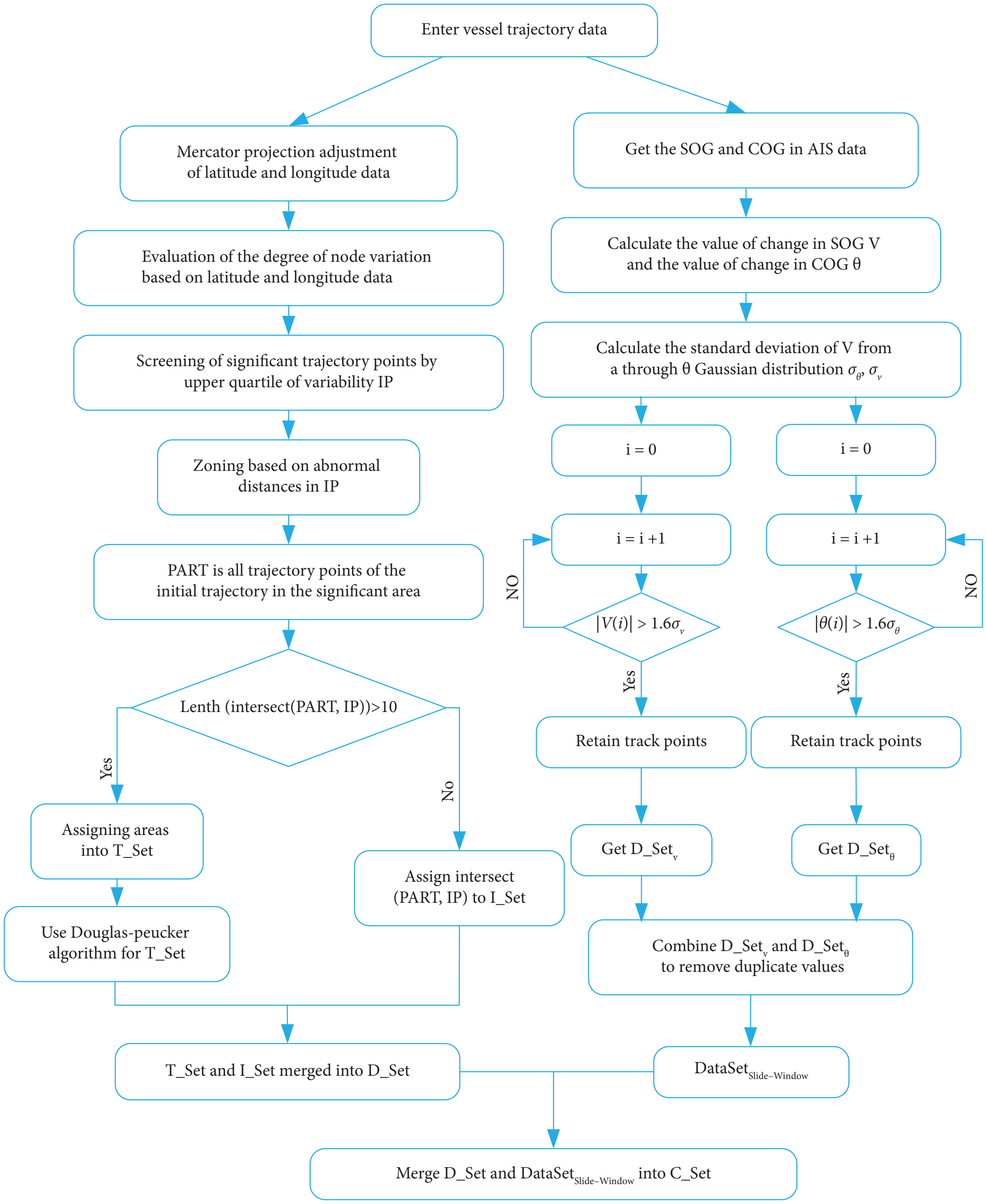

Enter vessel trajectory data
Mercator projection adjustment of latitude and longitude data
Evaluation of the degree of node variation based on latitude and longitude data
Screening of significant trajectory points by upper quartile of variability IP
Zoning based on abnormal distances in IP
PART is all trajectory points of the initial trajectory in the significant area
Lenth (intersect(PART, IP))>10
Yes
No
Assigning areas into T_Set
Use Douglas-peucker algorithm for T_Set
Assign intersect (PART, IP) to I_Set
T_Set and I_Set merged into D_Set
Get the SOG and COG in AIS data
Calculate the value of change in SOG V and the value of change in COG θ
Calculate the standard deviation of V from a through θ Gaussian distribution σ_θ, σ_v
i = 0
i = 0
i = i +1
i = i +1
NO
NO
$|V(i)| > 1.6\sigma_v$
$|\theta(i)| > 1.6\sigma_\theta$
Yes
Yes
Retain track points
Retain track points
Get D_Set$_v$
Get D_Set$_\theta$
Combine D_Set$_v$ and D_Set$_\theta$ to remove duplicate values
DataSet$_{\text{Slide-Window}}$
Merge D_Set and DataSet$_{\text{Slide-Window}}$ into C_Set

Figure 4: Flowchart of VATDC_CCRI algorithm operation.

3.2.1. Trajectory Compression of Critical Regions considering Spatial Characteristics. After identifying critical regions is completed, the maintenance of the original trajectory shape mainly depends on the critical regions and critical trajectory points outside the regions. Therefore, this paper uses the DP algorithm to compress only the critical regions of the original trajectory and combines the results with the critical trajectory points to obtain the compression results on the spatial location. The advantage of this method is that it can optimize the compression rate and compression quality of the trajectory, and the operating efficiency is much higher than that of the DP algorithm to execute the entire original trajectory. For the input threshold of the DP algorithm, this paper uniformly uses 0.8 times the vessel length as the input

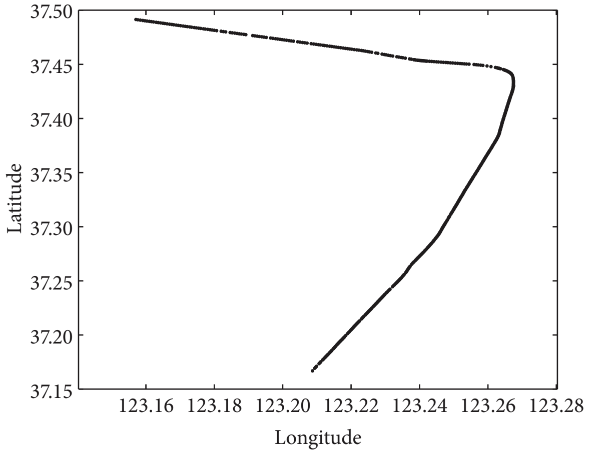


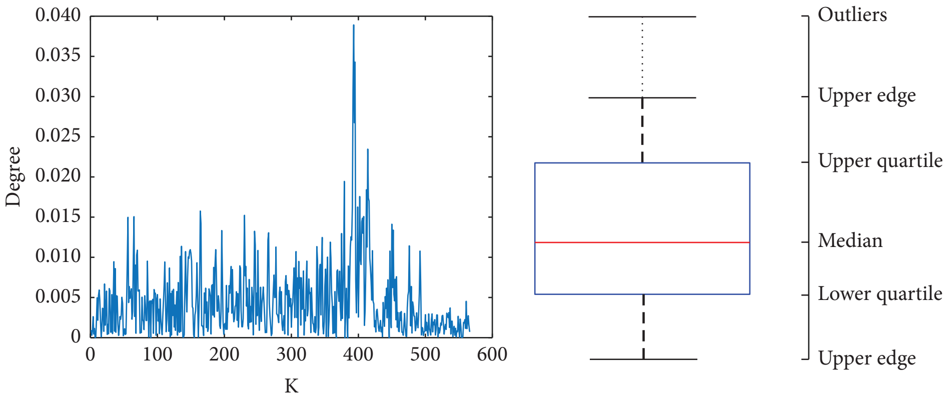
Figure 5: Vessel trajectory data with MMSI of 210698000.

```
Input Ship latitude and longitude coordinates information Point, Vessel speed(Vs).
Output Compressed ship latitude and longitude information C_set.
ID = [], D_set = [], Angle = [];
NI = Node Importance Evaluation (Point) (Algorithm 1)
NL = find(NI > p(,NI, 75))
/*$p$(,NI, 75) is the upper quartile point in the calculation NI */
IP = Point(NL)
for $i$ = 1 to len(IP) do
    $d$ = dist(IP($i$), IP($i$ + 1))
    ID = [ID; d]
end for
Outliers = p(ID, 75) + 1.5*(p(ID, 75) – p(ID, 25))
fal = [1, len(ID)]
Lbp = find(ID > Outliers)
Lbp* = ID(sort([Lbp; fal]))/*sort() is used to keep track points in order*/
Bp = find(Point = Lbp*)
for $i$ = 1 to len(Bp)
    part = Point(Bp($i$): Bp($i$ + 1))
    If lenth(intersect(part, IP)) > 10 then/* intersect () is used to find the intersection of matrices */
        Integration of part into $T$_set tuple
    else if
        Integration of intersect (part, IP) into I_set tuple
      end if
end for
for $i$ = 1 to len($T$_set) do
    Bp_set = Douglas-Peucker ($T$_set($i$), 0.8 times the length of the boat) (Algorithm 2)
    D_set = [D_set; Bp_set]
end for
D_set = union(D_set, I_set)/* union() is used to find the union of matrices*/
    $\text{DataSet}_{\text{Slide–Window}}$ = SW(Point, Vs) (Algorithm 3)
D_set = sort(Rpm(D_set, $\text{DataSet}_{\text{Slide–Window}}$))/ * Rpm for merging duplicate track points */
return C_set.
```

Algorithm 4: Ship trajectory compression algorithm considering region importance.

threshold proposed by Zhang et al. [10]. The comprehensive compression effect is shown in Figure 9. The red point in the figure is the compressed trajectory point, and the blue curve is the shape of the trajectory. The red point includes the deflection position of the trajectory shape, and the overall similarity of the trajectory before and after compression is relatively high.

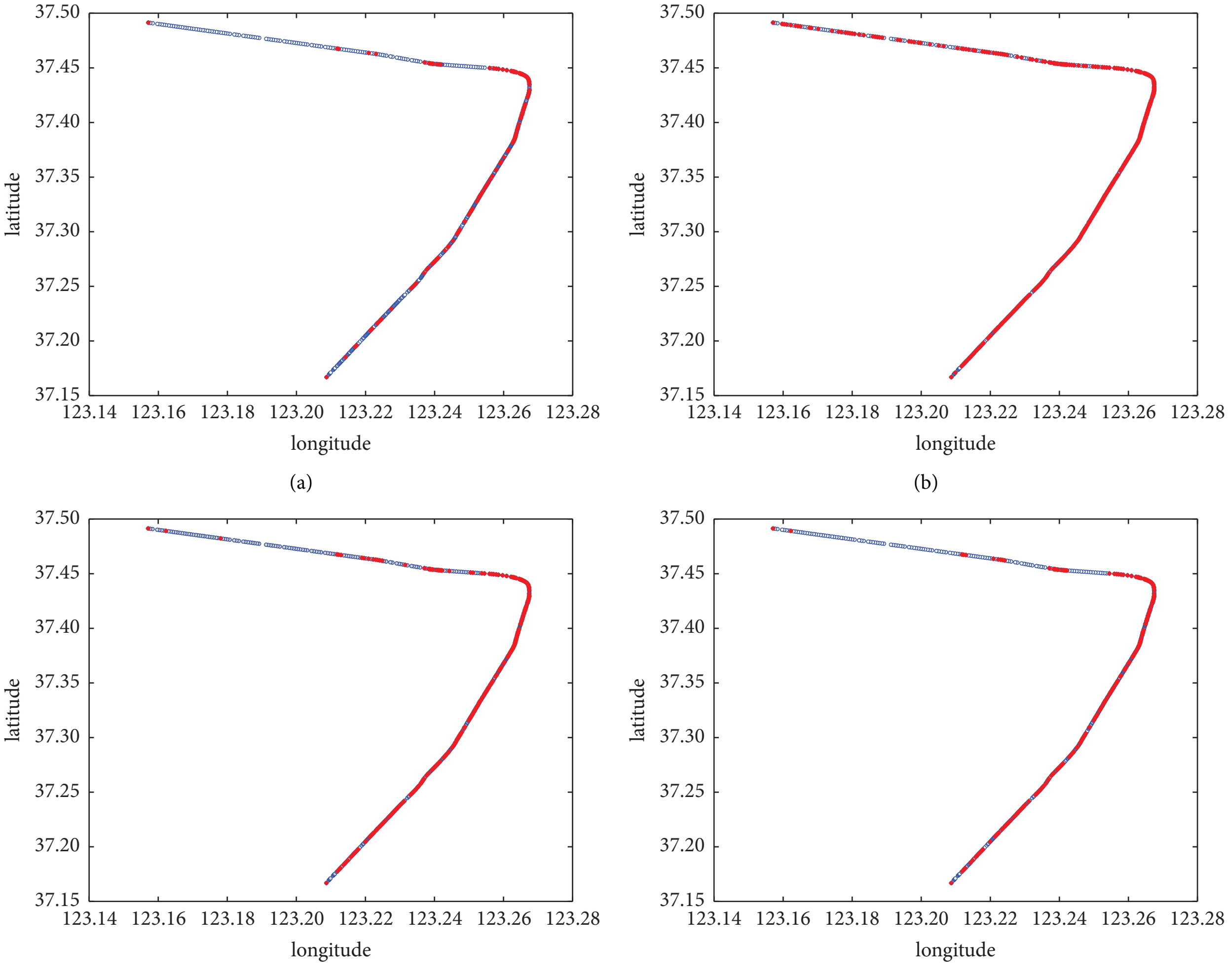
3.2.2. Extracting Trajectory Points with Time Characteristics in Vessel AIS Data. In this paper, a sliding window algorithm with low time complexity is used to compress the raw trajectories in order to preserve the temporal characteristics of the AIS trajectories. The two metrics proposed by Zhang et al. [26], namely vessel heading keeping and speed keeping,



(a)

(b)

Figure 6: Evaluation results of trajectory node change degree.



(a)

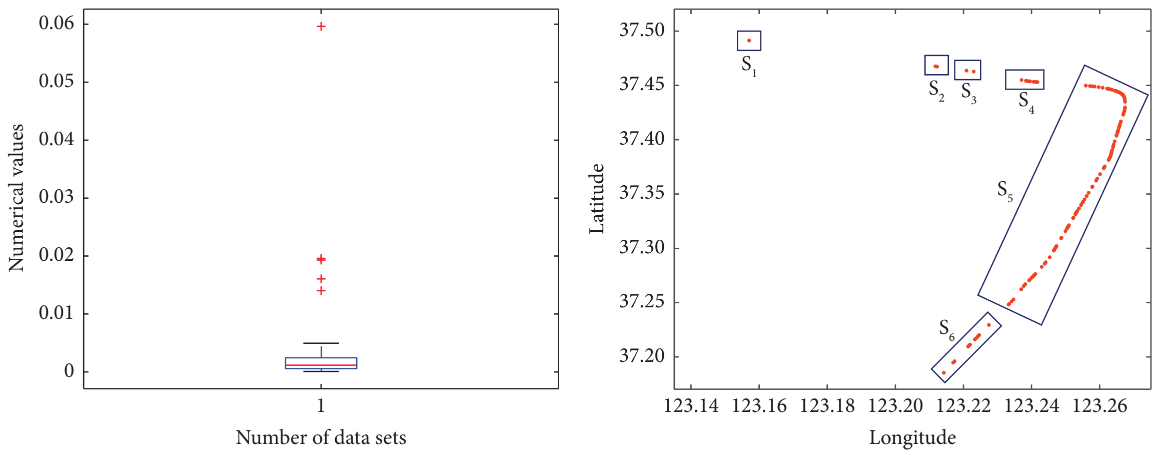
(b)

(c)

(d)

Figure 7: Screening results of critical trajectory points. (a) Upper quartile. (b) Lower quartile. (c) Median. (d) Mean.

are mainly used in the study as the main factors to measure the temporal characteristics. However, the window size and input threshold set in the sliding window algorithm also have an impact on the final compression effect [36, 37]. When the number of data points within the window increases to *n*, n calculations need to be performed, which will increase the time complexity of the algorithm to some extent. In contrast, when the number of trajectory points



(a)

(b)

Figure 8: Distribution of critical nodes and regions.

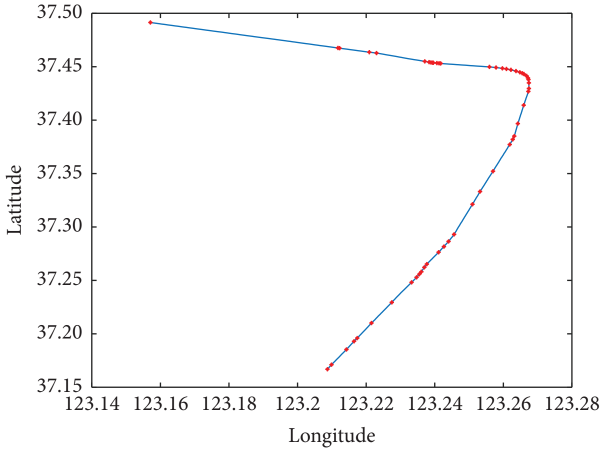


Figure 9: Plot of critical regions of AIS trajectories based on DP algorithm.

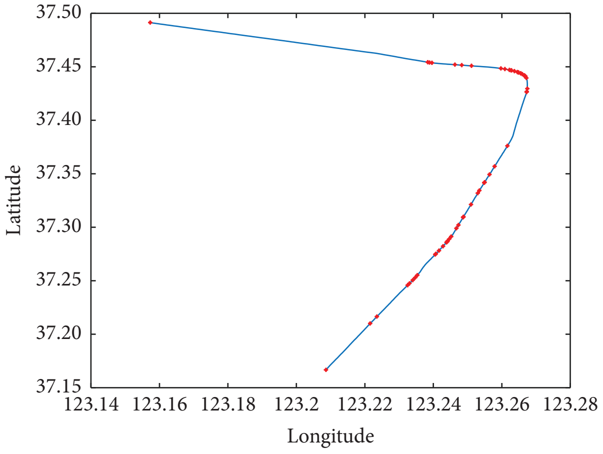


Figure 10: Graph of compression results for the sliding window algorithm.

within the window decreases, the compression effect is highly dependent on the setting of the input threshold. In the field of trajectory compression, sliding window algorithms most commonly use three points as a single window size, and this classical approach is also used in this paper. Moreover, it cites the 1.6 times standard deviation model analyzed by Wei et al. [21] using statistical methods as the input threshold. The trajectory points obtained by the sliding window algorithm according to the temporal characteristics are shown in Figure 10.

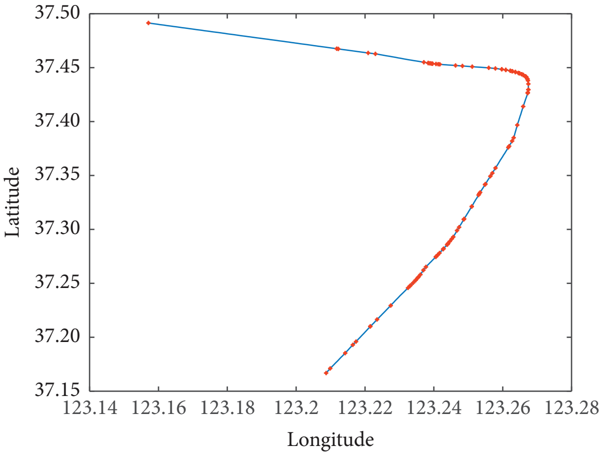


Figure 11: Compression effect of VATDC_CCRI algorithm.

3.2.3. VATDC_CCRI Algorithm Compression Results Presentation. The compression result combining the spatial characteristics and time characteristics is shown in Figure 11. The red dots in the figure represent the compressed trajectory points, and the blue line represents the original trajectory. The more obvious trajectory turning points in the figure have been preserved, and more nodes are reserved in the region with significant trajectory turning. The advantages and disadvantages of the compression algorithm mainly depend on the algorithm's running speed, compression rate, and length loss rate. Among them, the running speed of the algorithm is significantly improved after the region division, and the compression rate of the algorithm can be adjusted by parameters. In general, the length loss of the compressed trajectory is mainly concentrated in the regions where the ship turns frequently or the waters are complex [28], and the proposed algorithm also extracts iteratively for these regions. The length loss is mainly concentrated in the regions where the vessel turns frequently, or the waters are complex [28], and the target algorithm also repeatedly extracts these regions. Therefore, the VATDC_CCRI algorithm proposed in this paper inherits better the ability of the DP algorithm for regional trajectory shape retention and the sliding window algorithm for examining the change of AIS metrics within the time stamp. In addition, the result of identifying significant regions enhances the flexibility of the compression process while improving the efficiency of the algorithm operation.

3.3. VATDC_CCAI Complexity Analysis. This paper proposes a vessel trajectory data compression algorithm (VATDC_CCRI) that takes into account the identification of important regions, which extracts the important regions through the node variability evaluation algorithm and then compresses them through the DP algorithm, while, at the same time, executes the sliding window algorithm to compress the trajectory as a whole and finally takes the two parts of the result as the final compression result. In order to better evaluate the execution efficiency of the VATDC_CCRI algorithm, it is necessary to analyze its complexity, which includes both space complexity and time complexity. For the spatial complexity of the algorithm, assuming that the number of samples in the current trajectory dataset is n, the first thing that needs to be carried out is the evaluation of node variability and the identification of important regions; the space complexity of node variability evaluation is $O(n)$. Moreover, the identification of important regions is obtained through the trajectory points peeled off from the upper quarter guard point, in which the distance between the peeled-off trajectory points needs to be calculated and analyzed, and this part of the space complexity is $O(0.25n)$. Therefore, the space complexity of the comprehensive identification of the important region is $O(1.25n)$. Then, the trajectory of the important region is compressed by the DP algorithm, under the assumption that the number of trajectory points in the important region is m, and the increased space complexity is $O(m)\{m|m \in [10, n]\}$. Finally, the space complexity of the sliding window algorithm is also included $O(n)$. Therefore, the final space complexity of the VATDC_CCRI algorithm is $O(2.25n + m)$.

For the time complexity of the VATDC_CCRI algorithm, under the condition that the number of samples in the current trajectory dataset is n, the node variability needs to be calculated once for each trajectory point, and the time complexity of this part is $O(n)$. Then, the important regions will be extracted. Since the extraction of the important regions is obtained by the screening of the trajectory points by the upper quartile points, it includes the calculation of the distances between these points, and the time complexity of this part is $O(0.25n)$. After screening the important regions, the DP algorithm will be used to compress the key regions, and assuming that the number of key regions is i, the corresponding number of samples is x_i. Therefore, this part of the increased algorithmic time complexity is $O(x_1^2 + x_2^2 + \ldots + x_i^2)$. On the other hand, the classic three-point sliding window algorithm has a time complexity of $O(3n)$. Therefore, the overall time complexity of the VATDC_CCRI algorithm is $O(4.25n + x_1^2 + x_2^2 + \ldots + x_i^2)$, which is between the sliding window algorithm and the DP algorithm. When the number of samples of a single trajectory increases, the overall running efficiency of the algorithm will have a more obvious advantage of the algorithm's running speed compared with the classical DP algorithm.

In summary, the VATDC_CCRI algorithm proposed in this paper is slightly higher than the classical DP algorithm and sliding window algorithm in terms of space complexity. However, the overall space complexity still belongs to the same level. As for the time complexity, the VATDC_CCRI algorithm directly reduces a large number of calculations in the trajectory compression process by identifying the focus region of the trajectory, but, because the final compression result is a combination of the two algorithms, its time complexity is between the sliding window algorithm and the DP algorithm.

4. Vessel Trajectory Compression Experiments

4.1. Compression Evaluation Indicators. The compression evaluation index is an essential basis for evaluating the compression effect of the algorithm [38]. In this paper, the performance of the algorithm is investigated using five metrics: trajectory similarity (TS), trajectory compression rate (CR), length loss rate (LLR), algorithm running time, and algorithm overall efficiency (AOE). It is worth mentioning that the trajectory similarity is calculated using the dynamic time warping (DTW) algorithm, which can describe the relation vessel between two discrete time series using a time warping function and obtain the corresponding warp path distance (WPD) accurately. The algorithm is often used in pattern recognition and information retrieval due to its excellent data-matching effect [39]. COE is a comprehensive metric proposed in this paper to evaluate the compression performance of the algorithm, which combines the characteristics of four metrics: TS, CR, LLR, and algorithm running time. Under standardizing the index data, CR is the positive index, and TS, LLR, and algorithm running time are the negative indexes. The larger the positive indicator, the better the algorithm's effect; the smaller the negative indicator, the better the algorithm's effect. The calculation principles of all indicators are shown in equations (6)–(11):

$$\begin{aligned}\mathrm{WPD}(i, j) = {} & d\left(q_i, c_j\right) \\ & + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\},\end{aligned} \tag{6}$$

$$\begin{aligned}\mathrm{TS} & = \mathrm{DTW}(q, c) \\ & = \sum_{i=1}^{n}\sum_{j=1}^{m}\gamma(i, j),\end{aligned} \tag{7}$$

$$\mathrm{CR} = \frac{q^{'} - c^{'}}{q^{'}} \times 100\%, \tag{8}$$

$$\begin{aligned}\mathrm{LLR} & = \frac{Q - C}{Q} \\ & = \frac{\sum_{i=1}^{q^{'}-1}\left|q_i q_{i+1}\right| - \sum_{\mathrm{j}=1}^{c^{'}-1}\left|c_j c_{j+1}\right|}{\sum_{i=1}^{q^{'}-1}\left|q_i q_{i+1}\right|} \times 100\%,\end{aligned} \tag{9}$$

$$S(X) = \frac{X}{\sum_{i=1}^{n} X_i}, \tag{10}$$

$$\mathrm{COE} = \frac{S(\mathrm{CR})}{S(\mathrm{TS}) * S(\mathrm{LLR}) * S(\mathrm{time})}, \tag{11}$$

where q and c are the Vessel trajectory data before and after compression and equations (6) and (7) introduce the calculation principle of the DTW algorithm. q_i and cj in equation (6) are the elements in q and c and d (q_i, c_j) is the Euclidean distance of q_i and c_j. q' and c' in equation (8) are the number of trajectory points before and after compression, respectively. Q and C in equation (9) are the total lengths of the trajectory before and after compression and X in Equation (10) represents the value of the same metric for different algorithms.

4.2. Trajectory Compression Experiments

4.2.1. Experimental Environment and Comparison Algorithms. In this paper, a vessel trajectory compression algorithm named VATDC_CCRI is proposed, which is implemented by MATLAB using Windows 10 operating system with 64 bit architecture. The hardware environment consists of an Intel Core I5-7200 processor, 4 GB RAM, and a 128 GB hard disk. The implementation of the algorithm is presented in Section 3 and evaluated at the algorithm complexity level. In order to compare the actual effect of the VATDC_CCRI algorithm, three representative algorithms in the field of trajectory compression are selected for comparison, which are the DP algorithm, DP-Slide algorithm, and angle-speed algorithm. The DP algorithm takes 0.8 times the vessel length proposed by Zhang et al. [10] as the data threshold, and the DP-Slide algorithm is an algorithm considering the characteristics of vessel navigation proposed by Wei et al. [21]. It combines the features of the DP algorithm and sliding window algorithm and can retain relatively complete vessel trajectory information. The angle-speed algorithm is an algorithm proposed by Zhu and Ma [7] to compress vessel trajectories considering the processing mode. The algorithm compresses the trajectory points by analyzing the rate of change of steering and speed. The final experiment proves that the angular velocity algorithm can effectively compress vessel trajectory data online with low time complexity. Compared with the VATDC_CCRI algorithm proposed in this paper, the DP algorithm focuses on the overall positional change of the trajectory to judge whether the trajectory points should be compressed or not, while the VATDC_CCRI algorithm is able to comprehensively examine the spatial and temporal dimensions. The DP-Slide algorithm is a simple combination of the compression results of the DP algorithm and the slide algorithm. In contrast, the VATDC_CCRI algorithm greatly improves the algorithm's compression results through the focused region. The angle-speed algorithm mainly considers the rate of change of speed and steering angle in vessel trajectory, and its operating framework is the sliding window algorithm, which is also included in the VATDC_CCRI algorithm proposed in this paper.

4.2.2. Comparison of Single AIS Vessel Trajectory Data Compression. The experiment is divided into two parts. The first part examines the trajectory similarity, length loss rate, and the overall efficiency of the algorithm on a single trajectory. The experimental data are obtained from the AIS trajectory data of vessel number MMSI 210698000 on 13–15 April 2019. It contains 570 trajectory information points, each retaining information such as vessel latitude and longitude and vessel heading speed. The visualization results of the four compression algorithms are shown in Figure 12, where the hollow blue circles are the initial trajectory point markers and the red " $*$ " are the compressed trajectory point markers. The evaluation metrics are shown in Table 1, with the best-performing data marked in bold black font.

In the experiments, the VATDC_CCRI algorithm determines the speed and steering angle changes based on the parameters identified in Section 3, i.e., 0.8 times the vessel's length, the upper quarterback point to identify the focal area, and 1.6 times the standard deviation, and the DP algorithm uses 0.8 times the vessel's length, the DP-Slide algorithm uses 0.8 times the vessel's length and 1.6 times the standard deviation to determine the speed and steering angle changes, and the angle-speed algorithm is controlled by the standard deviation of the rate of change to be consistent with the compression rate of the VATDC_CCRI algorithm. In general, the larger the trajectory compression rate, the lower the trajectory similarity, and the higher the length loss rate. According to the performance of each algorithm in terms of trajectory similarity and length loss rate, the VATDC_CCRI algorithm and DP-Slide algorithm proposed in this paper perform better. Their trajectory similarity and length loss rate are only 0.0306 and 2.96%, respectively, which is not a big difference. In comparison, the VATDC_CCRI algorithm has a difference of 0.09% or more in the similarity of the trajectory compared with the DP algorithm and angle-speed algorithm. Compared with the DP algorithm and

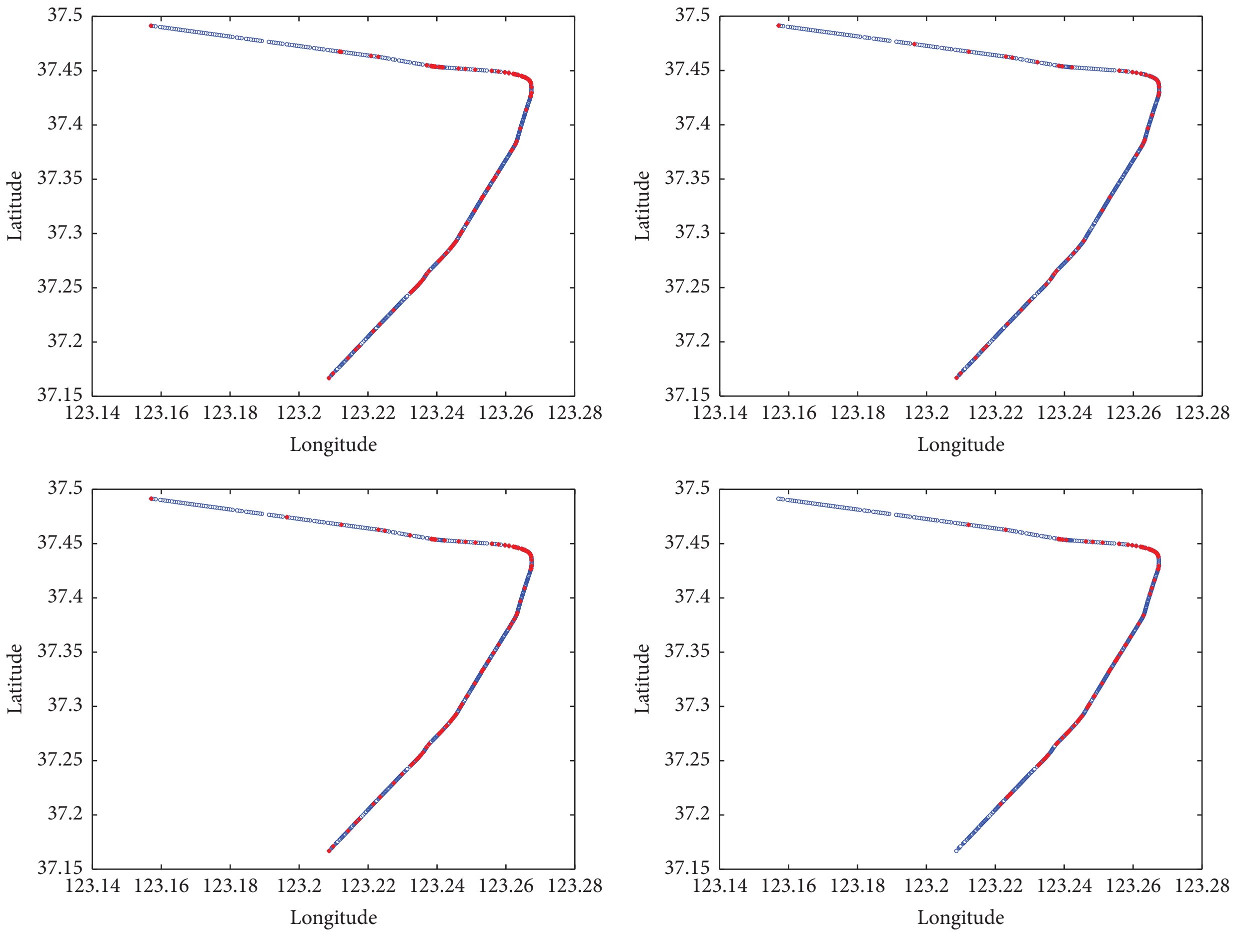


Figure 12: Algorithm compression effect graph.

Table 1: Algorithm performance parameters.

Algorithms	TS	CR (%)	LLR (%)	RT (s)	COE (%)
VATDC_CCRI	0.1571	85.95	13.26	0.2169	**33.94**
DP algorithm	0.2515	**91.57**	21.5	0.2781	10.87
DP-Slide algorithm	**0.1265**	81.9	**10.3**	0.4535	24.73
Angle-speed algorithm	0.3463	85.77	16.54	**0.186**	14.37

Bold values represent best-performing data.

angle-speed algorithm, the VATDC_CCRI algorithm has more than 0.09 similarity and more than 3% reduction in the length loss rate. In terms of trajectory compression rate, the DP algorithm has the best performance, reaching 91.57%, followed by the VATDC_CCRI algorithm, and the DP-Slide algorithm has a lower compression rate because it combines all the trajectory points of the two algorithms. The angle-speed algorithm performs the best in terms of runtime, while the VATDC_CCRI algorithm and DP algorithm perform similarly in terms of runtime due to the reasonable threshold selection of the DP algorithm and the small amount of trajectory data. The DP-Slide algorithm has the worst performance in runtime because it needs to run the DP algorithm and the sliding window algorithm on the whole trajectory data and then integrate them. In terms of comprehensive operation effect (COE), the VATDC_CCRI algorithm has the best effect, followed by the DP-Slide algorithm and finally the DP algorithm with angle-speed algorithm. Therefore, the VATDC_CCRI algorithm proposed in this paper can meet the quality requirements of vessel trajectory compression, and compared with other trajectory compression algorithms, the VATDC_CCRI algorithm has advantages in trajectory similarity, compression rate, length loss rate, and operation time.

4.2.3. Comparison of Compression Effects of Regional AIS Vessel Trajectory Data. The second part examines the compression effect of regional AIS vessel trajectory data, tests the stability of the VATDC_CCRI algorithm, and examines the effect of the time complexity of different algorithms. In this paper, vessel trajectory data from 22 to 27 April 2019 in the Cheng San Jiao region are selected for the experiments. Due to the influence of conditions such as climate and equipment damage during the transmission and storage of AIS data, there are outliers and missing data in the original vessel trajectory [40]. Therefore, before vessel trajectory compression, the original trajectory data must be preprocessed, and this process includes two parts; one part is to eliminate the vessel trajectory with more missing data and

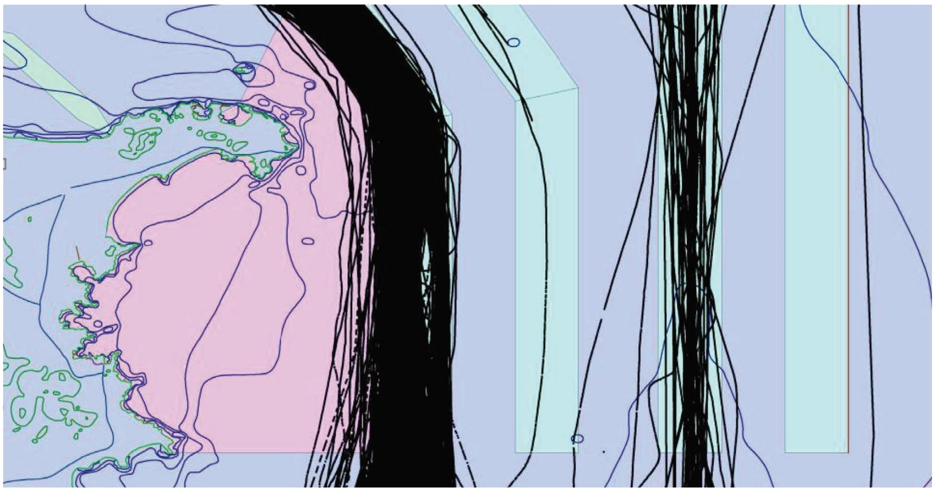
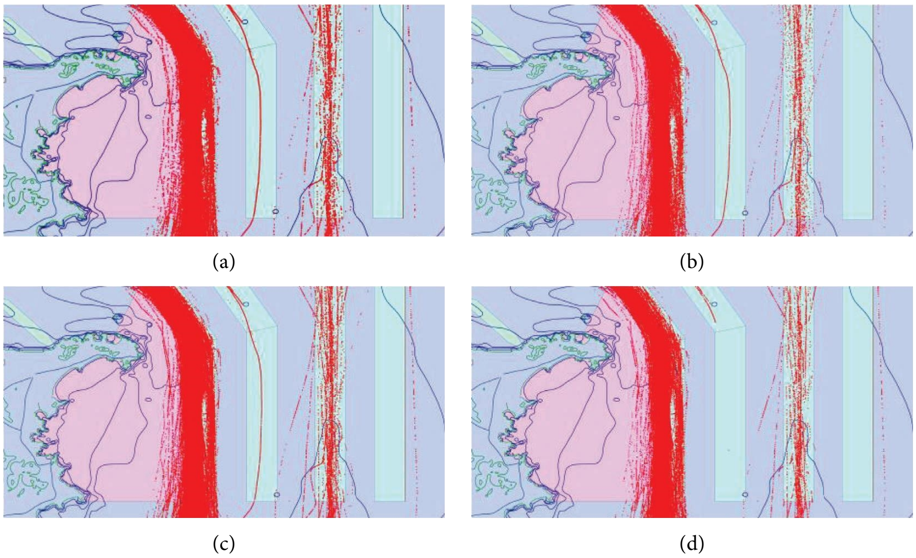


Figure 13: Distribution of original trajectory points.



(a)

(b)

(c)

(d)

Figure 14: Plot of trajectory compression results for different algorithms. (a) VATDC_CCRI algorithm. (b) DP algorithm. (c) DP-slide algorithm. (d) Angle-speed algorithm.

abnormal values, and in this part, there is still no effective algorithm to distinguish the normal data from the incomplete, missing, and abnormal data, so the process can only be filtered by hand. In the other part, for the trajectory data with fewer missing and abnormal values, we use the algorithm based on statistical theory and sliding window proposed by Yan et al. [22]. In the other part, for the trajectory data with few missing values and outliers, we use an algorithm proposed by Ran et al. [22] based on statistical theory and sliding window for identification, supplemented and corrected by linear interpolation, and then data compression by four algorithms covered in Section 4.2.2. There are 779 vessel trajectories after cleaning, involving 576819 trajectory points. The original trajectory distribution is shown in Figure 13, and the comprehensive trajectory compression effect of different algorithms is shown in Figure 14.

The trajectory data compression results of the four algorithms in Figure 14 show that the DP algorithm has a higher compression rate. In contrast, the angle-speed algorithm is the result of adjusting the amount of data compressed according to the algorithm in this paper, which cannot guarantee the integrity of the vessel's trajectory shape features and therefore is prone to having too many missing parts. The experimental conditions of the comprehensive compression rate, length loss rate, and compression time of all algorithms are shown in Figures 15 and 16. Similar to the experimental results for individual trajectories, the length loss rates of the VATDC_CCRI and DP-Slide algorithms are 14.25% and 11.24%, respectively, for a large number of vessel trajectory datasets, which is a significant advantage compared to the other algorithms. Compared with the compression rate and running time of the DP-Slide algorithm, the compression rate and compression time of the

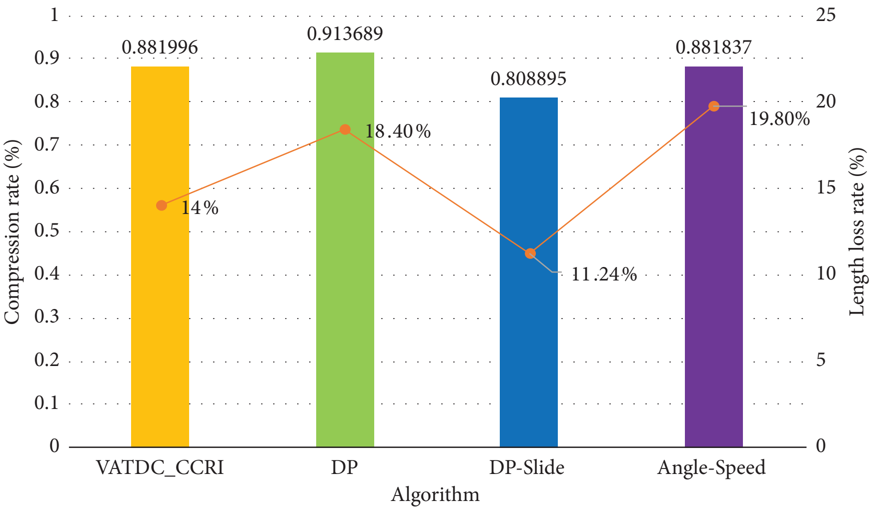


Figure 15: Compression rate and average length loss rate of different algorithms.

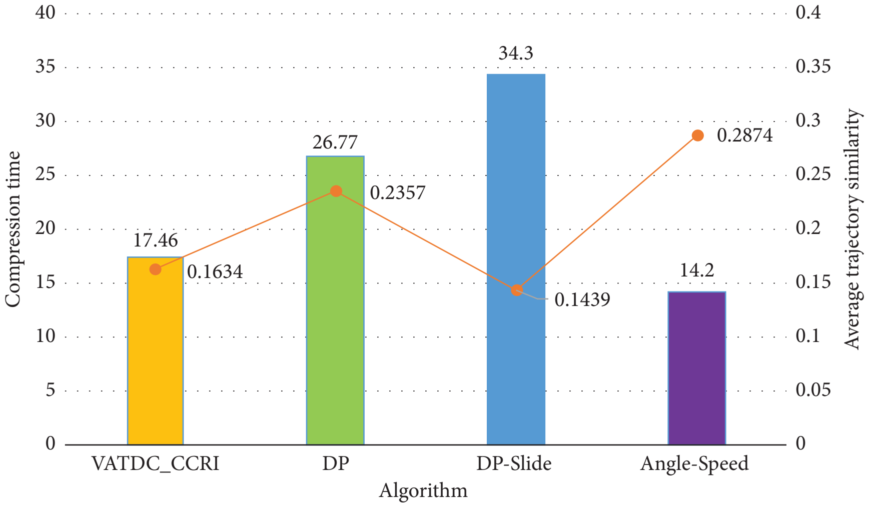


Figure 16: Compression time and average trajectory similarity for different algorithms.

VATDC_CCRI algorithm are, respectively, increased by 7.31% and 16.84 s. In addition, under the same threshold as the DP-Slide index, the VATDC_CCRI algorithm has a higher compression ratio. Usually, the compression rate of the vessel's trajectory data compression algorithm is directly proportional to the length loss rate. That is, the increase in the compression rate will also cause an increase in the length loss rate. However, the VATDC_CCRI algorithm proposed in this paper can significantly improve the algorithm's performance in terms of compression rate and compression time while sacrificing less length loss rate and has more obvious advantages compared with traditional trajectory compression algorithms.

5. Conclusions

Vessel trajectory data compression is crucial for mining maritime traffic information and improving the efficiency of maritime data processing. In this paper, a vessel trajectory data compression algorithm (VATDC_CCRI) considering critical region identification is proposed, and experiments on single and multiple vessel trajectories prove the stability of the algorithm. Compared with the current major similar research results, the VATDC_CCRI algorithm has three innovative points. First, the algorithm divides the critical regions of vessel trajectory operation according to the degree of change of trajectory points, takes the critical regions as the key to keeping the shape of vessel trajectories, and preserves the trajectory shapes in these regions by the DP algorithm. Second, the VATDC_CCRI algorithm combines the features of the DP algorithm and sliding window algorithm to improve the utilization of vessel trajectory information, and the compressed data better retain the spatiotemporal and shape features of vessel trajectory. In addition, compared with other compression algorithms, the compression ratio is higher, and the compression time is shorter under the same conditions. Third, in most cases, a single vessel trajectory has a large span, so a single trajectory threshold only guarantees the effective extraction of the whole trajectory information. However, the VATDC_CCRI algorithm can identify critical regions where vessel trajectories vary, which means that more critical regions can be extracted when faced with trajectories with a large amount of data. These regions will have the opportunity to be compressed using different thresholds, further improving the quality of compression, which is an important guide for the study of navigational regions and routes. However, the research in this paper also has some things that could be improved. For example, in the experiments, due to the need to control the compression rate to compare the performance of different algorithms in terms of trajectory similarity and length loss rate, this paper controls the compression rate of the angle-speed algorithm through the standard deviation, which may lead to the compression effect of this algorithm that is not optimal. In addition, more parameters are introduced in the VATDC_CCRI algorithm.

Data Availability

All data or codes used to support the findings of this study are available from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

S.Z. proposed the methodology, supervised the study, performed funding acquisition, and reviewed and edited the manuscript; X.Z. wrote the original draft and provided the software.

Acknowledgments

This work was supported by Natural Science Foundation of Fujian Province (Grant no. 2020J01658), Open Project Fund of National Local Joint Engineering Research Center for Ship Assisted Navigation Technology (Grant no. HHXY2020002), and Doctoral Start-up Fund of Jimei University (Grant no. ZQ2019012).

References

[1] H. L. Zheng, J. Yang, and Y. Zhao, "Reliability demonstration test plan for degraded products subject to Gamma process with unit heterogeneity," *Reliability Engineering & System Safety*, vol. 240, Article ID 109617, 2023.

[2] H.-S. Kim, E. Lee, E.-J. Lee et al., "A study on grid-cell-type maritime traffic distribution analysis based on AIS data for establishing a coastal maritime transportation network," *Journal of Marine Science and Engineering*, vol. 11, no. 2, p. 354, 2023.

[3] J. Chen, J. X. Zhang, H. Chen, Y. Zhao, and H. D. Wang, "A TDV attention-based BiGRU network for AIS-based vessel trajectory prediction," *iScience*, vol. 26, no. 4, Article ID 106383, 2023.

[4] L. Huang, "A mathematical modeling and an optimization algorithm for marine ship route planning," *Journal of Mathematics*, vol. 2023, Article ID 5671089, 8 pages, 2023.

[5] H. Liu, Y. Liu, B. Li, and Z. Qi, "Ship abnormal behavior detection method based on optimized GRU network," *Journal of Marine Science and Engineering*, vol. 10, no. 2, p. 249, 2022.

[6] R. W. Liu, J. Nie, S. Garg et al., "Data-driven trajectory quality improvement for promoting intelligent vessel traffic services in 6G-enabled maritime IoT systems," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5374–5385, 2021.

[7] F. Zhu and Z. Ma, "Ship trajectory online compression algorithm considering handling patterns," *IEEE Access*, vol. 9, pp. 70182–70191, 2021.

[8] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[9] A. K. Singh, V. Aggarwal, P. Saxena, and O. Prakash, "Performance analysis of trajectory compression algorithms on marine surveillance data," *International Conference on Advances in Computing Mangalore, India*, vol. 30, pp. 1074–1079, 2017.

[10] S. K. Zhang, Z. J. Liu, Y. Cai, Z. L. Wu, and G. Y. Shi, "AIS trajectories simplification and threshold determination," *Journal of Navigation*, vol. 69, no. 4, pp. 729–744, 2016.

[11] J. X. Liu, H. H. Li, Z. L. Yang, K. F. Wu, Y. Liu, and R. W. Liu, "Adaptive Douglas- peucker algorithm with automatic thresholding for AIS-based vessel trajectory compression," *IEEE Access*, vol. 7, pp. 150677–150692, 2019.

[12] C. H. Tang, H. Wang, J. H. Zhao, Y. Q. Tang, H. R. Yan, and Y. J. Xiao, "A method for compressing AIS trajectory data based on the adaptive-threshold Douglas-Peucker algorithm," *Ocean Engineering*, vol. 232, Article ID 109041, 2021.

[13] J. Gao, Z. Cai, W. Yu, and W. Sun, "Trajectory data compression algorithm based on ship navigation state and acceleration variation," *Journal of Marine Science and Engineering*, vol. 11, no. 1, p. 216, 2023.

[14] L. B. Zhao and G. Y. Shi, "A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm," *Ocean Engineering*, vol. 166, pp. 37–46, 2018.

[15] X. F. Wang, W. Yang, Y. Liu et al., "Segmented douglas-peucker algorithm based on the node importance," *Ksii T Internet Inf*, vol. 14, pp. 1562–1578, 2020.

[16] J. S. Lee, H. T. Lee, and I. S. Cho, "Maritime traffic route detection framework based on statistical density analysis from AIS data using a clustering algorithm," *IEEE Access*, vol. 10, pp. 23355–23366, 2022.

[17] Y. Huang, Y. Li, Z. F. Zhang, and R. W. Liu, "GPU-accelerated compression and visualization of large-scale vessel trajectories in maritime IoT industries," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10794–10812, 2020.

[18] X. M. Song, C. X. Cheng, C. H. Zhou, and D. H. Zhu, "Gestalt-based douglas-peucker algorithm to keep shape similarity and area consistency of polygons," *Sensor Letters*, vol. 11, no. 6, pp. 1015–1021, 2013.

[19] J. H. Shi and Z. J. Liu, "Track pairs collision detection with applications to ship collision risk assessment," *Journal of Marine Science and Engineering*, vol. 10, no. 2, p. 216, 2022.

[20] Z. Zhou, Y. Zhang, X. Yuan, and H. Wang, "Compressing AIS trajectory data based on the multi-objective peak douglas–peucker algorithm," *IEEE Access*, vol. 11, pp. 6802–6821, 2023.

[21] Z. K. Wei, X. L. Xie, and X. J. Zhang, "AIS trajectory simplification algorithm considering ship behaviours," *Ocean Engineering*, vol. 216, Article ID 108086, 2020.

[22] R. Yan, H. Y. Mo, D. Yang, and S. A. Wang, "Development of denoising and compression algorithms for AIS-based vessel trajectories," *Ocean Engineering*, vol. 252, Article ID 111207, 2022.

[23] M. Gao, G. Y. Shi, and W. F. Li, "Online compression algorithm of AIS trajectory data based on improved sliding window," *Journal of Traffic and Transportation Engineering*, vol. 18, pp. 222–231, 2018.

[24] M. Gao and G. Y. Shi, "Ship spatiotemporal key feature point online extraction based on AIS multi-sensor data using an improved sliding window algorithm," *Sensors*, vol. 19, no. 12, p. 2706, 2019.

[25] H. L. Shen, H. Tang, and Y. Yin, "A novel method for ship trajectory clustering," *International Journal of Naval Architecture and Ocean Engineering*, vol. 14, Article ID 100474, 2022.

[26] Y. Q. Zhang, G. Y. Shi, S. Li, and S. K. Zhang, "Vessel trajectory online multi- dimensional simplification algorithm," *Journal of Navigation*, vol. 73, no. 2, pp. 342–363, 2020.

[27] X. S. Chen, Z. Y. Yang, and Y. J. Liu, "Simplified algorithm of moving object trajectory based on interval floating," *Mobile Information Systems*, vol. 2022, Article ID 5032375, 10 pages, 2022.

[28] L. F. Sánchez-Heres and H. Sánchez, "Simplification and event identification for AIS trajectories: the equivalent passage plan method," *Journal of Navigation*, vol. 72, no. 2, pp. 307–320, 2018.

[29] S. Sun, Y. Chen, Z. J. Piao, and J. S. Zhang, "Vessel AIS trajectory online compression based on scan-pick-move algorithm added sliding window," *IEEE Access*, vol. 8, pp. 109350–109359, 2020.

[30] Y. L. Zhong, J. L. Kong, J. Q. Zhang, Y. Z. Jiang, X. Fan, and Z. Y. Wang, "A trajectory data compression algorithm based on spatio-temporal characteristics," *PeerJ Computer Science*, vol. 8, p. e1112, 2022.

[31] J. Y. Han, W. Lu, K. Ge et al., "A pattern accumulated compression method for trajectories constrained by urban road networks," *Data & Knowledge Engineering*, vol. 145, Article ID 102143, 2023.

[32] D. Tomar, P. Tomar, A. Bhardwaj, and G. R. Sinha, "Deep learning neural network prediction system enhanced with best window size in sliding window algorithm for predicting domestic power consumption in a residential building," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7216959, 14 pages, 2022.

[33] R. W. Liu, M. Liang, J. Nie, B. Lim, Y. Zhang, and M. Guizani, "Deep learning-powered vessel trajectory prediction for improving smart traffic services in maritime internet of things," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3080–3094, 2022.

[34] N. Khan, K. Iqbal, and M. G. Martini, "Lossless compression of data from static and mobile dynamic vision sensors-performance and trade-offs," *IEEE Access*, vol. 8, pp. 103149–103163, 2020.

[35] S. Blindheim and T. A. Johansen, "Electronic navigational charts for visualization, simulation, and autonomous ship control," *IEEE Access*, vol. 10, pp. 3716–3737, 2022.

[36] Y. Xu, G. Li, Z. Li et al., "Smartphone-based unconstrained step detection fusing a variable sliding window and an adaptive threshold," *Remote Sensing*, vol. 14, no. 12, p. 2926, 2022.

[37] G. Huang, D. Wang, Y. Du, Q. Zhang, Z. Bai, and C. Wang, "Deformation feature extraction for GNSS landslide monitoring series based on robust adaptive sliding window algorithm," *Frontiers of Earth Science*, vol. 10, Article ID 884500, 2022.

[38] A. Afnan, F. Ullah, Y. Yaseen, J. Lee, S. Jamil, and O. J. Kwon, "Subjective assessment of objective image quality metrics range guaranteeing visually lossless compression," *Sensors*, vol. 23, no. 3, p. 1297, 2023.

[39] E. Mizutani and S. Dreyfus, "On using dynamic programming for time warping in pattern recognition," *Information Sciences*, vol. 580, pp. 684–704, 2021.

[40] R. W. Liu, M. H. Liang, J. T. Nie et al., "STMGCN: mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7977–7987, 2022.