

Trajectory Time Series Compression Algorithm Based on Unsupervised Segmentation

Shuang SUN

School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China
E-mail: 470582807@qq.com

Yan CHEN

School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China
E-mail: chenyan@dlmu.edu.cn

Zaiji PIAO*

School of Software, Dalian University of Foreign Languages, Dalian 116044, China
E-mail: pozaiji@dlufl.edu.cn

Abstract Aiming at the problem of ignoring the importance of starting point features of trajectory segmentation in existing trajectory compression algorithms, a study was conducted on the preprocessing process of trajectory time series. Firstly, an algorithm improvement was proposed based on the segmentation algorithm GRASP-UTS (Greedy Randomized Adaptive Search Procedure for Unsupervised Trajectory Segmentation). On the basis of considering trajectory coverage, this algorithm designs an adaptive parameter adjustment to segment long-term trajectory data reasonably and the identification of an optimal starting point for segmentation. Then the compression efficiency of typical offline and online algorithms, such as the Douglas-Peucker algorithm, the Sliding Window algorithm and its enhancements, was compared before and after segmentation. The experimental findings highlight that the Adaptive Parameters GRASP-UTS segmentation approach leads to higher fitting precision in trajectory time series compression and improved algorithm efficiency post-segmentation. Additionally, the compression performance of the Improved Sliding Window algorithm post-segmentation showcases its suitability for trajectories of varying scales, providing reasonable compression accuracy.

Keywords trajectory time series; unsupervised segmentation; trajectory compression; greedy randomized adaptive search

1 Introduction

With the widespread application of mobile devices with locating capabilities in various industries and fields, it is necessary to analyze the historical trajectory from sensors of vehicles, vessels, mobile phones, and so on^[1]. Using the big data mining method and the data-driven research mode^[2–5], researchers can monitor the potential traffic flow in specific areas and identify abnormal behaviors of moving objects. Common trajectory time series^[6, 7] contains not only

Received November 8, 2023, accepted April 28, 2024

Supported by the Basic Research Projects of Liaoning Provincial Department of Education (LJKQZ20222459)

* Corresponding author

the moving route of objects like people or vehicles collected by GPS equipment, but also the change of values collected by sensors with time. Therefore, it can be said that trajectory is everywhere. These rich resources bring a huge demand for trajectory time series research including not only the increase of data transmission load and storage pressure, but also the decrease of data query volume, efficiency and analysis performance. So, the research on the compression algorithm of trajectory time series of moving objects is an urgent problem to be addressed. Due to the limitations of existing storage and computing capabilities, the purpose of moving object trajectory compression^[8] is to reduce the amount of data and the storage space occupied by the data as much as possible on the presumption of retaining the information contained in the data. The achievement of a low compression ratio by retaining only turning points is often accompanied by an increase in compression errors, particularly in preserving the details and overall fidelity of the original data. This trade-off between compression ratio and compression errors is a common challenge in trajectory time series compression algorithms. However, for trajectory data mining higher accuracy according to different requirements should be achieved, especially for short-term trajectory time series prediction^[9–11].

In recent years, numerous studies on the offline and online compression of moving object trajectory have been conducted^[12]. Offline compression mainly preserves the most representative trajectory points in historical or static trajectory time series from a global perspective, such as the Douglas-Peucker (DP) algorithm proposed by Douglas in 1973^[13]. After that, a large number of researchers have studied and improved DP algorithm. In 2004, Meratnia and de By^[14] proposed the top-down algorithm based on speed and time ratio, which is a variant of the DP algorithm. In 2017, Lin, et al.^[15] proposed the ATS algorithm of velocity reservation considering the velocity characteristics of the trajectory based on the DP algorithm. Zhao and Shi^[16] proposed a method for simplifying ship trajectory based on improved DP algorithm, which significantly reduced the compression time under high compression strength. Singh, et al.^[17] put forward a kind of SPM (scan-pick-move) algorithm inspired by the idea of the DP algorithm to compare the error of each trajectory point in turn and modify the baseline selection technique of error assessment. However, current trajectory big data compression mainly preserves the points with the largest cumulative error in the trajectory from the local perspective such as the AIS (automatic identification system) data, so offline compression is not applicable to all scenes.

At present, most of the mainstream online compression algorithms are based on the Sliding Window (SW) and the Open Window (OPW)^[18, 19]. Unlike the SW algorithm, the OPW algorithm calculates the vertical Euclidean distance and time synchronization Euclidean distance of each point in the window. The SW algorithm and the OPW algorithm only consider the position of trajectory points in the current window, so they can be optimized in each local part, but they can't consider the overall trend. Later, Meratnia and de By^[14] also proposed the Opening Window Time Ratio (OPW-TR) algorithm using a synchronized Euclidean distance (SED) instead of vertical distance to consider time characteristics. Potamias, et al.^[20] proposed a STTrace algorithm oriented to small memory devices, which ensures the compression of long trajectory under the premise of limited memory consumption. Muckell, et al.^[21] proposed an online compression algorithm of Spatial Quality Simplification Heuristic (SQUISH). Similar

to the SW algorithm, SQUISH initializes a priority queue and adds trajectory points to the queue, when the queue is full, delete the points that cause the minimum error, and recalculate the priority of each point. SQUISH algorithm has low time complexity, low compression ratio and low trajectory error, but the error of compressed trajectory has no upper bound. Recently, researchers have begun studying the application of compression algorithms to improve the processing time of clustering analysis. Alam and Torgo^[22] analyzed the impact of compression technique Ramer Douglas Peucker (RDP) on the clustering analysis process, and the results showed that compression had no significant impact on the clustering results. Carteron, et al.^[23] quantitatively analyzed the effectiveness of applying compression algorithms to improve the clustering efficiency of ship traffic anomaly detection. The offline compression considers the whole trajectory and obtains a globally better compression. However, online compression only considers local trajectories, so the compression error is relatively high. The average time complexity of offline compression is generally higher than that of online compression^[24–26].

Although the improvement of these typical algorithms can achieve high computational performance or low compression error, the trajectory sequence of moving objects has the characteristics of high dimension, complexity, dynamics and high noise. If the original data is compressed directly, it will lead to the problems of inaccurate mining results, low time efficiency and low reliability of research conclusions. At the same time, the existing compression algorithms are based on a fixed trajectory starting point, and the change of the starting point will have great impacts on the compression results. In summary, the trajectory time series compression method has the following problems: during the compression process, some accuracy information of the original trajectory points will be lost, resulting in a certain error between the compressed trajectory and the original trajectory; compression methods based on trajectory segmentation typically require multiple scans, segments, and calculations of trajectory data, resulting in high time complexity; compression methods based on trajectory segmentation often require setting some parameters, such as segment length, error threshold, etc. The selection of different parameters may lead to different compression effects. Our study focuses on trajectory compression after appropriate segmentation in which setting of the starting point needs to rely on a lot of experience of the user^[27, 28]. This process has a high degree of uncertainty and great dependence on the users. It needs repeated experimental verification to determine the compression effect that meets the user's requirements^[29]. Therefore, before trajectory compression, it is necessary to segment the multidimensional trajectory sequence reasonably, which is conducive to the semantic enhancement of trajectory time series and subsequent series calculation. Segment and mark the long-time trajectory to make the segmented trajectory have specific significance, such as representing a certain trip^[30, 31]. Such segmentation operation not only retains the characteristics and knowledge of trajectory time series, but also reduces the complexity of subsequent data mining calculation. Therefore, the trajectory compression method based on segmentation proposed in this article plays an important role in the field of trajectory data processing. Trajectory data is one of the important forms of spatial data in geographic information systems. The trajectory segmentation based compression method helps to reduce the cost of data storage and processing, and promotes the development of the field of geographic information systems; provide an effective solution for trajectory data processing in the field of mobile computing;

provide an effective means for processing large-scale trajectory data in the fields of data mining and machine learning.

Considering the space-time characteristics of trajectory and the influence of starting point on trajectory compression, this paper proposes a trajectory compression algorithm of moving objects based on unsupervised segmentation. First combine greed and random construction to construct a feasible solution. Secondly, search the neighborhood of the feasible solution and attempts to improve it through local search. The complexity of the segmented trajectory is reduced in the subsequent trajectory mining calculation process, while also providing richer rules and knowledge than before segmentation, improving the effectiveness of storage. Then, the maximum offset distance is introduced to add a sliding window to the segmented trajectory segment for trajectory compression, in order to process anomaly detection and trajectory prediction in the future. In summary, the contributions of this paper are as follows:

1. The idea of adaptive parameters unsupervised segmentation is introduced into the process of trajectory compression.
2. The maximum offset distance reference trajectory point and scan-pick-move are used to choose whether the current trajectory point can be compressed.
3. For the same trajectory time series in the Brazilian coastal fishing vessel data sets, the compressed trajectory curve after segmentation has a high degree of fit with the original trajectory.
4. An extensive experimental study provides concrete evidence that after segmentation by PGRASP-UTS, appropriate trajectory purity may still be attained. Compression time of multiple trajectories compression in different number of trajectories are also compared in the DP, SW, and ISW algorithms.

The remainder of this paper proceeds as follows. In Section 2, the basic definitions of spatiotemporal trajectory is introduced and overall idea of method for compression trajectory based on PGRASP-UTS is proposed. In Section 3, basic information of trajectory segmentation and compression are introduced. Section 4 gives the results and discussion about the experimental results of three compression algorithms, and the conclusions are discussed in Section 5.

2 Problem Formulation and Preliminaries

2.1 Basic Definitions

The spatiotemporal trajectory of a moving object is an ordered collection of a series of location points. As a spatial entity, each location point has multiple attributes such as time, space and semantics^[32, 33]. According to the moving trajectory of the object, the basic characteristic information of the object movement can be obtained, and the characteristic information of the object group and its region can be further obtained. We define spatiotemporal trajectory time series as follows:

Definition 1 *A spatiotemporal trajectory time series T is a collection of several spatiotemporal trajectory points which can be expressed as $T = \{p_1, p_2, \dots, p_n\}$. For different trajectories in a data set, the values of n may be different.*

Definition 2 *A spatiotemporal trajectory point p_i is a tuple $\langle x_i, y_i, t_i, o_i \rangle$, where x_i, y_i represent spatial information, t_i represents time information, o_i represents speed, direction,*

destination and other information.

Definition 3 A trajectory segment $s = \{p_u, \dots, p_v\}$ is a sub trajectory ($s \in T$) of a spatiotemporal trajectory T , representing a trajectory segment from p_u to p_v , $0 \leq u \leq v \leq n$.

2.2 Overall Idea

The problem of unsupervised segmented trajectory compression is defined as follows: The original trajectory time series set and segment loss function are given, the trajectory segment set $\theta_T = (s_1, s_2, \dots, s_T)$ is obtained, and the compressed trajectory segment set $\theta_{T'} = (s_1', s_2', \dots, s_{T'})$ is derived based on off-line and on-line trajectory compression algorithms. The specific framework of three compression algorithms based on segmentation is as follows:

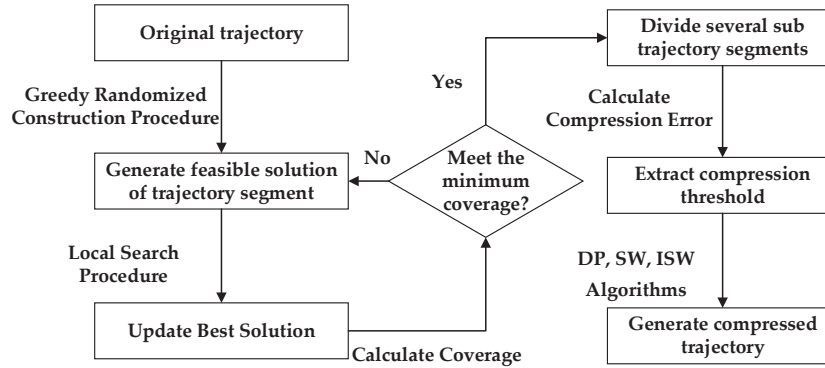


Figure 1 Algorithm framework

Figure 1 shows the framework of this method, which was divided into four steps:

1. Greedy randomized construction procedure. Generate feasible solution of trajectory segment based on dynamic and static information extracted from trajectory time series set.
2. Local search procedure. Update the best solution of trajectory segments to improve the segment results.
3. Calculate coverage between trajectories. We assigned corresponding trajectory purity to different trajectories to get reasonable coverage and divide the sub trajectory segments.
4. Online and offline compression procedure. Extract compression threshold to generate compressed trajectory by the DP, SW, ISW algorithms in the high trajectory accuracy.

3 Methodology

According to the existing relevant research, the trajectory segmentation methods can be divided into two categories, supervised methods and unsupervised methods^[34]. The supervised method can be understood as the need for human participation to set segmentation rules, such as setting thresholds based on human experience or subjective prior knowledge. Common methods can set segmentation rules based on the time, speed and space thresholds of moving objects, and combine these threshold methods to segment trajectories^[35, 36]. Generally, the supervised method performs well in specific applications, but for the universality of other applications, the segmentation rules and thresholds should be reset according to the relevant domain knowledge, and the portability is poor. However, the unsupervised method can be understood as setting

segmentation rules without human participation. The unsupervised algorithm can adaptively complete the trajectory segmentation task according to the designed loss function and other methods. Therefore, compared with the supervised method, the unsupervised method can have better universality in a variety of application scenarios.

3.1 Trajectory Segmentation

3.1.1 Basic Definitions

According to Definitions 1~3, for the algorithm GRASP-UTS^[37], we give the definition of segmented landmarks as follows:

Definition 4 A segmented landmark sm_k is a representative trajectory point of a space-time trajectory T , which is used to divide a space-time trajectory $sm_k = p_l$, $1 \leq k \leq l \leq N$. Then a trajectory segment can be expressed as $s = (sm_k, p_u, \dots, p_v)$.

The spatiotemporal trajectory, and the relationship between trajectory segments and segment landmarks are shown in Figure 2.

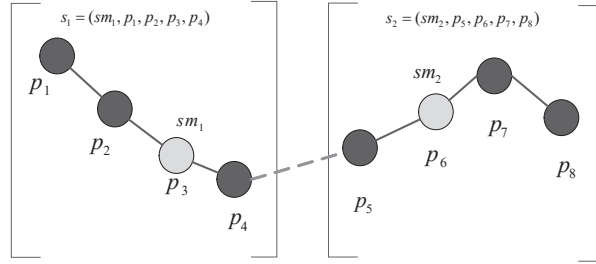


Figure 2 Relationship among spatiotemporal trajectory, trajectory segments and segment landmarks

The trajectory in Figure 2 contains eight trajectory points ($N = 8$) and two trajectory segments $s_1 = (sm_1, p_1, p_2, p_3, p_4)$ and $s_2 = (sm_2, p_5, p_6, p_7, p_8)$. Segment landmarks $sm_1 = p_3$, $sm_2 = p_6$ are used as the representative trajectory points of the two trajectory segments.

3.1.2 Cost Function

The definition of cost function can better help segmentation. In order to measure the similarity between segment landmarks and each trajectory point in the trajectory segment, considering all the features o_M of the trajectory point, the algorithm GRASP-UTS defines the expression of feature distance based on the Euclidean distance, as shown in Equation (1), in which p_{1_i} represents the i th feature of point p_1 . The segment cohesion formula as shown in Equation (2) is defined to calculate the similarity between segment landmarks and all points in the trajectory segment:

$$dist(p_1, p_2) = \sqrt{\sum_{i=1}^M (p_{1_i} - p_{2_i})^2}, \quad (1)$$

$$coh(sm, p_u, \dots, p_v) = \sum_{j=u}^v dist(sm, p_j). \quad (2)$$

In the process of establishing the cost function, two parts need to be built based on the MDL principle^[28]: $L(H)$ and $L(D|H)$. Finding the best landmark set corresponding to the loss function means finding the best hypothesis according to the MDL principle. $L(H)$ measures the maximum compression and $L(D|H)$ measures the mini-mum distortion, as shown in Equation (3) and Equation (4). These two parts constitute the cost function that conforms to the expected principle.

$$MDL = L(H) + L(D|H), \quad (3)$$

$$cost(\alpha_T, \beta_T) = f(\alpha_T) + g(\beta_T). \quad (4)$$

In (3),

$$L(H) = f(\alpha_T) = \log_2 \left(1 + \sum_{i=1}^{T-1} \max_dist - dist(sm_i, sm_{i+1}) \right), \quad (5)$$

$$L(D|H) = g(\beta_T) = \log_2 \left(1 + \sum_{j=1}^T coh(sm_j, [p_u \cdots p_v]_j) \right), \quad (6)$$

$\max_dist$ represents the maximum value of the two feature distances in the trajectory point.

3.1.3 The PGRASP-UTS Algorithm

GRASP-UTS algorithm is mainly divided into two stages: Greedy randomized construction and local search. In the first stage, a feasible solution is constructed based on greed and randomness. In the second stage, the feasible solution is improved by local search. In the whole process of unsupervised segmentation, the maximum number of iterations and greedy degree need to be provided manually, and it needs many times of debugging to achieve the desired segmentation results. The framework of GRASP-UTS is shown in Algorithm 1.

Therefore, this paper proposes an adaptive parameters unsupervised trajectory segmentation algorithm (Adaptive Parameters GRASP-UTS) on the basis of fully considering the spatiotemporal characteristics of the trajectory. The minimum coverage ($\min Cov$) of the trajectory segment is introduced into the unsupervised segmentation algorithm. The coverage of the trajectory segment represents the coverage of the result of the algorithm segmentation in the original annotation segment. The coverage threshold of the trajectory segment is set in advance by the desired segmentation result, which realizes the flexible adjustment in the segmentation process of the algorithm. The algorithm idea of the improved PGRASP-UTS is as follows:

1. The greedy coefficient α and the minimum length $\min Time$ of the trajectory segment are extracted;
2. The maximum number of trajectory segments that can be constructed is calculated according to $\min Time$, and the subsection feasible solution is constructed greedily and randomly;
3. The optimal solution of the segment is updated by local search;

4. After calculating the coverage of the trajectory segments, the greedy coefficient and the minimum length of the trajectory segment are extracted from the sets until the minimum coverage condition is satisfied.

Algorithm 1 GRASP-UTS

Input: A set of trajectory points ordered by time $T_N = (p_1, p_2, \dots, p_N)$

A minimum time for constructing trajectory segments $\min Time$

A segmentation factor p_f

A Greedy coefficient threshold α

Output: A set of trajectory segments $\theta_T = (s_1, s_2, \dots, s_T)$

```

01:  $\max Num \leftarrow N * val / \min Time$ 
02:  $c \leftarrow T_N = (p_1, p_2, \dots, p_N)$ 
03: repeat
04:      $sm \leftarrow random(c)$ 
05:     Add  $sm$  into  $\varphi_T$ 
06:      $\theta_T \leftarrow BuildSegments(T_N, \varphi_T, p_f)$ 
07:     if  $(\theta_T \cdot time < \min Time)$ 
08:         Del  $sm$  from  $\varphi_T$ 
09:     end if
10: until  $c == null$  or  $\theta_T \cdot size == \max Num$ 
11:  $step \leftarrow p_f * s_i \cdot size$ 
12: for  $(i = 1 \rightarrow T)$ 
13:     for  $(j = u \rightarrow v ; j += step)$ 
14:          $\theta_T \leftarrow BuildSegments(T_N, \varphi_T, p_f)$ 
15:         if  $(\theta_T \cdot time \geq \min Time)$ 
16:              $list \leftarrow \langle p_j, cost \rangle$ 
17:         end if
18:     end for
19:     Arrange  $list$  in ascending order of  $cost$ 
20:      $RCL = (p_1, \dots, p_l), l = \alpha * list \cdot size$ 
21: end for
22: return  $Best\theta_T$ 

```

The framework of PGRASP-UTS is shown in Algorithm 2.

Algorithm 2 PGRASP-UTS

Input: A set of trajectory points ordered by time $T_N = (p_1, p_2, \dots, p_N)$

A minimum coverage ($\min Cov$) of trajectory segment

A minimum time set ($\min Time_{list}$) to create a segment

A Greedy coefficient set α_{list}

Output: A set of trajectory segments $\theta_T = (s_1, s_2, \dots, s_T)$

01: for ($i = 1 \rightarrow T$)

02: $\min Time \leftarrow random(\min Time_{list})$

03: $\alpha \leftarrow random(\alpha_{list})$

04: $\theta_T \leftarrow \text{Greedy Randomized Construction Procedure}(T_N, \min Time, \alpha)$

05: $\theta_T \leftarrow \text{Local Search Procedure}(T_N, \min Time, \alpha)$

06: Update Best Solution ($\theta_T, Best\theta_T$)

07: if $Cov(Best\theta_T, \theta_T) < \min Cov$ then

08: Update ($\min Time_{list}, \alpha_{list}$)

09: end if

10: end for

11: return $Best\theta_T$

3.1.4 Segment Evaluation

Reference [39] uses trajectory segment coverage and trajectory segment purity to evaluate the effectiveness of segmentation. Trajectory segment coverage represents the degree of coverage of the algorithm segmentation results in the original labeled segmentation. For example, Suppose that segment $s_1 = (p_5, p_6, \dots, p_{11})$ and segment $s_2 = (p_{12}, p_{13}, \dots, p_{16})$ are obtained through an algorithm, and $s = (p_5, p_6, \dots, p_{13})$ in the original trajectory segment is marked as a non-fishing vessel, then s_1 is the trajectory segment with the highest coverage, with a coverage of $7/9 = 0.78$. That is to say, 78% of the trajectory points in segment s are covered by s_1 . The formula for calculating average coverage is as follows:

$$AvgCov(\psi_V, \theta_T) = \frac{100}{V} \left(\sum_{i=1}^V \arg \max_{j \in [1, T]} \frac{N_c(\psi_V, \theta_j)}{N_i} \right). \quad (7)$$

Among them, ψ_V and θ_T represent the annotated set of segments and the segmented set after algorithm segmentation, represents the number of trajectory points for segment ψ_i , and $N_c(\psi_V, \theta_T)$ represents the number of trajectory points for segment θ_j obtained through algorithm segmentation.

The purity of trajectory segments represents the proportion of trajectory points labeled with the most frequent behavior in the segmentation results of the algorithm. For example, if segment s_1 obtained by the algorithm contains 25 points, and the ship has no fishing operation (NOT-FISHING) among the 20 points, and is fishing operation (FISHING) among the 5 points, the purity of trajectory segments is $20/25 = 0.8$, because the most frequent behavior label is non fishing operation, which means that 80% of trajectory points belong to the same label (NOT- FISHING). The formula for calculating average purity is as follows:

$$AvgPur(\theta_T, \eta_L) = \frac{100}{T} \left(\sum_{i=1}^T \arg \max_{j \in [1, L]} \left(\frac{N_{ij}}{N_i} \right) \right). \quad (8)$$

Among them, η_L represents the set of labels in the dataset, N_i represents the number of trajectory points in segment θ_i , and N_{ij} represents the number of trajectory points with label η_j in segment θ_i . The greater the probability that the trajectory points with the same behavior will be divided into the same segment.

3.2 Trajectory Compression

Three common segmentation linear approximation trajectory compression algorithms are compared and analyzed in segmentation and compression, including the offline compression Douglas-Peucker algorithm, the online compression Sliding Window algorithm and the improved Sliding Window algorithm^[12]. The off-line compression algorithm needs to compress the complete trajectory time series. The online compression algorithm can compress the trajectory while obtaining the trajectory points.

3.2.1 Douglas-Peucker Algorithm

Trajectory compression algorithm should retain more appropriate trajectory points, rather than randomly retain trajectory points. The DP Algorithm^[13, 40] is a compression algorithm that can retain more appropriate points. Its main idea is to replace the original trajectory with several line segments. The basic steps of the DP algorithm are shown in Figure 3. First, take the straight-line segment between the first point and the last point on the trajectory as the approximate trajectory, calculate the vertical Euclidean distance from the point to the segment for the middle point, and select the point with the maximum distance; then add this point as a splitting point to the approximate trajectory if the maximum distance exceeds the preset distance threshold and the whole trajectory is divided into two sub trajectories; repeat the above steps for the two sub tracks respectively until the maximum vertical distance in the sub trajectory is less than the distance threshold or there are only two points in the trajectory (start point and end point).

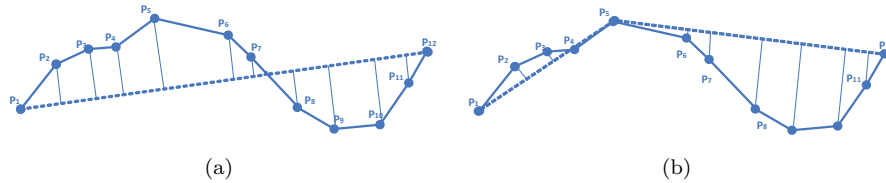


Figure 3 Douglas-Peucker compression algorithm

3.2.2 Sliding Window Algorithm

The core of the typical Sliding Window compression algorithm is to process only three points all the time, and use the idea of gradual compression to transmit the data in the form of a stream for online compression^[41].

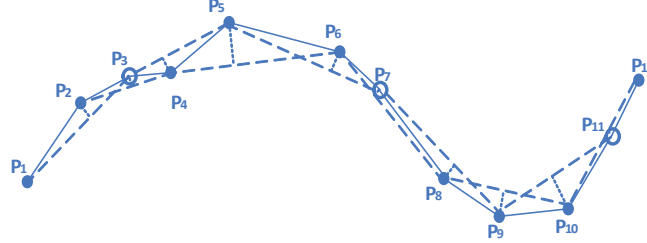


Figure 4 Sliding Window compression algorithm

However, once some changes occur at the start or end of the trajectory, it is necessary to recalculate the vertical distance from all trajectory points to the start and end points, which increases the complexity of the whole algorithm and reduces the efficiency of the algorithm. To solve this problem, the maximum offset distance reference trajectory point in the sliding window is proposed as the criterion of whether the current trajectory point to be compressed can be compressed^[42]. In this way, when there is a change in the sliding window, only the vertical distance between the reference trajectory point with the maximum offset distance and the current trajectory point to be compressed and the straight line between the start trajectory point and the end trajectory point need to be calculated, so as to significantly reduce the complexity of the algorithm and the compression processing time. According to the principle of plane analytic geometry, the calculation formula of vertical distance d_{cur} from the point to the line is as follows:

$$d_{cur} = \sqrt{(x_{P_{cur}} - x_{P'_{cur}})^2 + (y_{P_{cur}} - y_{P'_{cur}})^2} \quad (9)$$

$$x_{P'_{cur}} = \frac{x_{P_{cur}} + K \cdot (y_{P_{cur}} - y_{P_{start}}) + K^2 \cdot x_{P_{start}}}{1 + K^2} \quad (10)$$

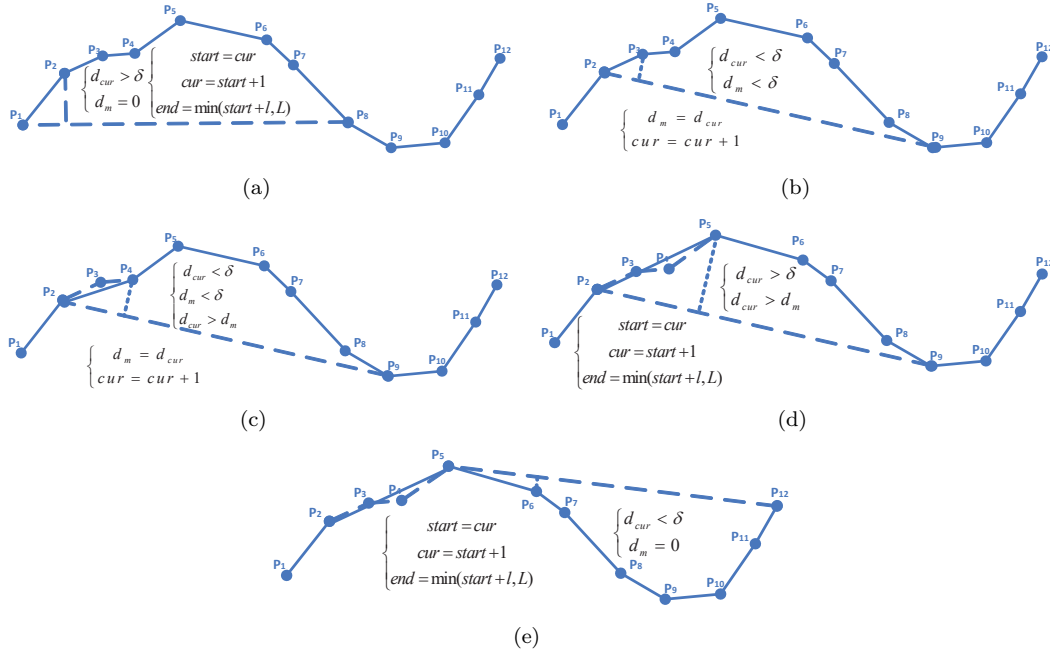
$$y_{P'_{cur}} = x_{P_{cur}} + K \cdot (x_{P'_{cur}} - x_{P_{start}}) \quad (11)$$

where, $K = \frac{y_{P_{end}} - y_{P_{start}}}{x_{P_{end}} - x_{P_{start}}}$, similarly, maximum offset distance d_m can be calculated. The basic steps of the Improved Sliding Window algorithm are shown in Figure 5.

The comparison of the characteristics of the three algorithms is shown in Table 1.

Table 1 Comparison of three compression algorithms.

Algorithm	Distance	On/Off line	Complexity	Error
Douglas-Peucker	vertical	off	$O(n \log n)$	spatial distance
Sliding Window	vertical	on	$O(n^2)$	spatial distance
Improved Sliding Window	vertical	on	$O(n)$	the maximum offset distance

**Figure 5** Improved Sliding Window compression algorithm

4 Results

4.1 Experimental Setting

The compression results of three compression algorithms before and after segmentation are compared and analyzed by a large number of experiments. The specific configuration is as follows: Windows10, Intel Core-i5-8259U, CPU 2.30 GHz 8 GB RAM, python3.7, and IDE is PyCharm. The open data set of trajectories of Brazilian coastal fishing vessels over a period of time is adopted, including ship longitude and latitude, operation start and end time, average speed (speed_inference_m_s), direction change (direction_inference) and other information. Partial information of one of the trajectories is intercepted as shown in Table 2.

Table 2 Brazilian coastal fishing vessel data sets

tid	sid	latitude	longitude	time	d_i	s_i_m_s	label
1	1	-5.778	-35.206	2004/4/14 3:16	0	0	NOT_FISHING
1	1	-5.778	-35.206	2004/4/14 4:08	0	0	NOT_FISHING
1	1	-5.778	-35.206	2004/4/14 5:52	4.09	0	NOT_FISHING
1	1	-5.764	-35.205	2004/4/14 20:06	56.56	0.01	NOT_FISHING
1	1	-5.701	-35.093	2004/4/14 21:00	0.5	4.39	NOT_FISHING

The average velocity and direction change are selected as the point features of the method. In order to determine the impact of segmentation algorithm on trajectory compression performance, a trajectory shown in Figure 6(a) is selected (tid = 4). Based on PGRASP-UTS

trajectory segmentation algorithm, we divide the trajectory into 5 sub trajectory segments, as shown in Figure 6(b).

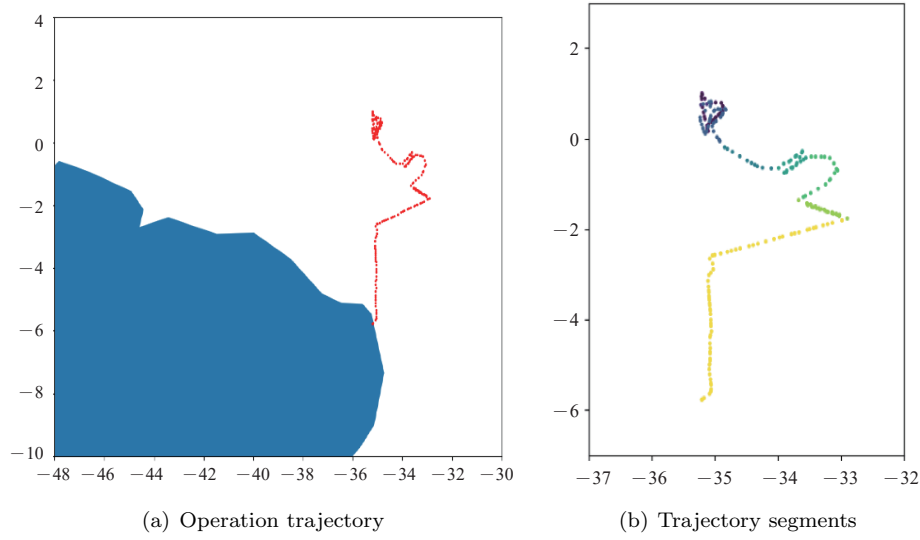


Figure 6 One operation trajectory and segments of Brazilian coastal fishing vessel (tid = 4)

4.2 Compression Evaluation and Algorithm Comparison

4.2.1 Evaluation Criteria

In order to evaluate the compression effect, this paper adopts the most common performance evaluation methods^[43, 44]: compression time T , compression ratio R and compression error E . They are defined as follows:

Compression time T : Time before compression T_1 and time after compression T_2 .

$$T = T_2 - T_1. \quad (12)$$

Compression ratio R : The number of trajectory points before compression M and after compression N .

$$R = \frac{N}{M} \times 100\%. \quad (13)$$

Distance error E : The average distance between the non-feature point and its two adjacent feature points, l_k is the distance between the non-feature point k and the adjacent feature points.

$$E = \frac{1}{M - N} \sum_{k=1}^{k=M-N} l_k. \quad (14)$$

On the premise of satisfying the compression error ($E \leq 5\%$), this paper carries out three algorithms for segmented compression based on the trajectories before and after segmentation, and the threshold value obtained takes the sliding window as an example, as shown in Figure 7.

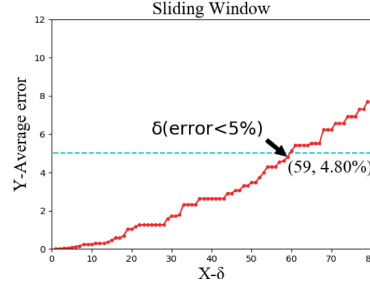


Figure 7 Relationship between compression threshold and error of the Sliding Window algorithm (tid = 4)

4.2.2 Algorithm Comparison of Segmentation

In this section, we compare PGRASP-UTS with other existing trajectory data segmentation algorithms, including Warped K-Means (W-Kmeans), as they are all unsupervised methods and use different cost functions to generate trajectory segments; The CB-SmoT algorithm belongs to the supervised method, which compares explicit standards with SGRASP-UTS through application; GRASP-UTS algorithm and RGRASP SemTS algorithm.

Table 3 Comparison of segmentation results of several trajectory segmentation algorithms

Algorithm	Average purity (%)	Total number of segments
CB-SmoT	91.13	189
W-Kmeans	91.16	122
GRASP-UTS	91.37	96
RGRASP-SemTS	88.50	56
PGRASP-UTS	91.34	75

In terms of trajectory average purity, PSGRASP-UTS is similar to the GRASP-UTS algorithm and has a certain improvement compared to the other three algorithms. This indicates that the proportion of trajectory points with the same label in the segmented segment of the algorithm is relatively small compared to the other three algorithms. This may be due to the fact that the average number of trajectory segments in the algorithm in the article is relatively small, resulting in a higher number of trajectory points in the trajectory segments, which increases the number of trajectory points with different labels and reduces the purity of the trajectory segments.

4.2.3 Algorithm Comparison of Single Trajectory

Take the maximum thresholds of three algorithms DP (Douglas-Peucker), SW (Sliding Window), ISW (Improved Sliding Window) to compress the trajectory of ship tid = 4. The compression results before and after segmentation of single trajectory are shown in the Figure 8~10.

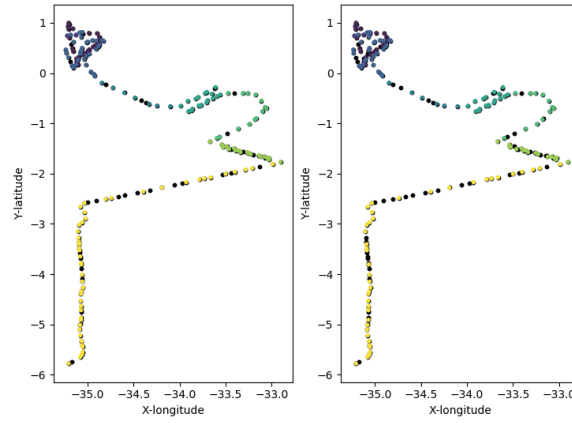


Figure 8 Fitting diagram of DP algorithm before and after segmentation (trajectory points: 173 / 205 before segmentation, 158 / 205 after segmentation)

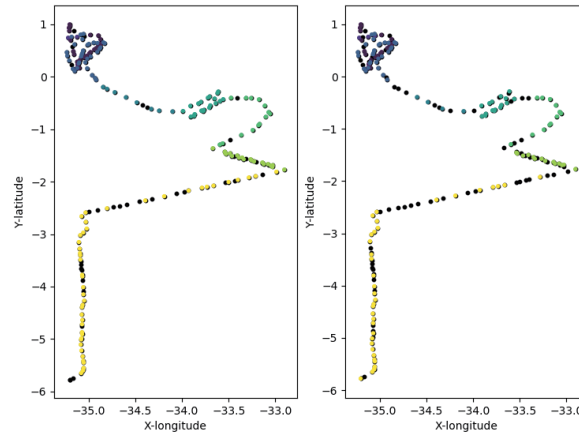


Figure 9 Fitting diagram of SW algorithm before and after segmentation (trajectory points: 171 / 205 before segmentation, 157 / 205 after segmentation)

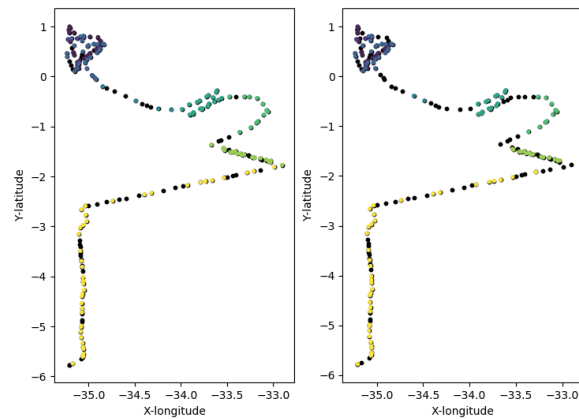


Figure 10 Fitting diagram of ISW algorithm before and after segmentation (trajectory points: 154 / 205 before segmentation, 95 / 205 after segmentation)

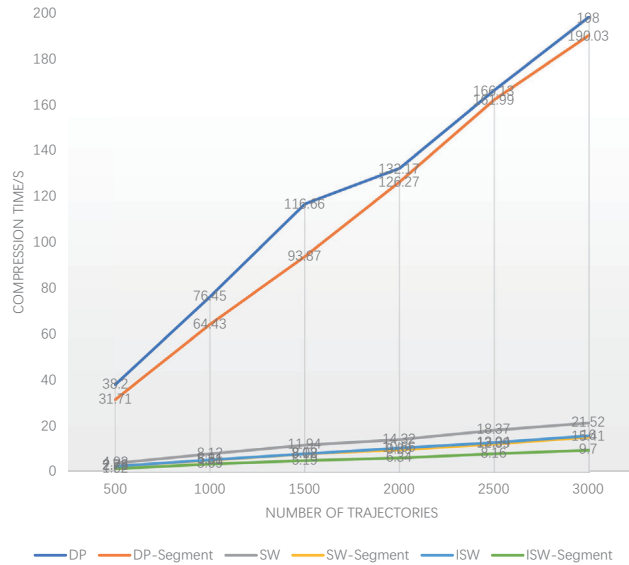
Table 4 Comparison of compression ratios before and after segmentation

Algorithm	compression ratio (%) before segmentation	compression ratio (%) after segmentation
DP	84.39	77.07
SW	83.41	76.59
ISW	75.12	46.34

Figure 8 shows that the compression ratio of the DP algorithm decreased by 6.73% after segmentation, which means that the segmentation algorithm is also effective in improving of-line compression ratio. From the comparison results of Figures 9 and 10, it can be seen that the improved sliding window algorithm can achieve lower compression ratio after segmentation, and the visually distinguished feature points can be preserved. Therefore, it can be seen that the three compression algorithms compress the same trajectory time series, and the compressed trajectory curve has a high degree of fit with the original trajectory, and the segmented trajectory time series has a low compression ratio under the condition of more retaining the start and end points of trajectory segmentation, which improves the work efficiency for the follow-up trajectory mining to a certain extent.

4.2.4 Algorithm Comparison of Multiple Trajectories

In order to compare the errors of the three algorithms in multi trajectory compression, we selected 3000 trajectory time series in a certain period of time for compression. Figure 11 shows the execution time required for the three algorithms to meet similar compression errors by compressing different amounts of trajectories in the data set before and after segmentation.

**Figure 11** Comparison of compression time before and after segmentation of three algorithms

5 Discussion

It can be seen from the results that the overall compression efficiency of the DP algorithm is higher than that of the SW algorithm and its improved algorithm, and the compression efficiency difference is becoming larger and larger with the increase of data volume. When the data volume is small, the three algorithms have similar compression efficiency before and after segmentation. However, when the amount of data is large, the compression efficiency of the three algorithms after segmentation is higher than that before compression. What's more, with the increase of data volume, the compression time of DP algorithm before and after segmentation increases significantly, while the compression time of ISW algorithm remains at a low level. From this we can find that:

1. The unsupervised segmented data not only retains the initial feature points of segmentation, but also improves the compression efficiency to a certain extent;
2. The segmented DP segment algorithm has lower compression efficiency and is only suitable for small-scale trajectory time series compression in the offline state;
3. The segmented SW segment algorithm and the improved ISW segment algorithm always have high compression efficiency for any scale of trajectory time series compression;
4. The improved algorithm shows better performance as the amount of data increases so it is more suitable for any scale of trajectory time series online compression scenarios. In summary, the segmented compression algorithm improves the compression ratio while ensuring compression error, and the compression efficiency also improves with the increase of the number of trajectories.

6 Conclusions

Most of the existing trajectory time series compression algorithms are based on the original trajectory segments of the data set. The idea of adaptive parameter unsupervised trajectory segmentation proposed can not only fully retain the semantics and characteristics of the starting point of each trajectory segment, but also adaptively adjust the segmentation parameters, which also improves the compression efficiency of offline and online trajectory compression algorithms. And the segmentation method PGRASP-UTS has little effect on the parameters of the subsequent compression algorithm, and has good applicability to both the basic algorithm and the improved algorithm. The segmentation algorithm proposed is applied to ship trajectory data compression, which improves the efficiency of computer operation and provides a theoretical basis for future research on ship trajectory big data processing, ship motion pattern recognition, ship motion feature extraction, and single ship behavior. However, the current unsupervised trajectory segmentation trajectory compression algorithm still has some limitations, that is, when the amount of data is huge, parallel computing is required. It is hoped that the unsupervised trajectory segmentation trajectory compression algorithm proposed can be applied to more trajectory mining scenarios based on the support of big data framework, in particular, the research on multi-dimensional attributes and complex data types of trajectory time series is the focus of our next research. We also hope to optimize the loss function, especially when dealing with non numerical attributes, how to consider them will be our next research topic.

References

- [1] Sadoun B, Al-Bayari O. LBS and GIS technology combination and applications. Proceedings of the 2007 IEEE/ACS International Conference on Computer Systems and Applications, Amman, Jordan, May 2007: 578–583.
- [2] Li M T, Li X, Yin J. TORD problem and its solution based on big trajectories data. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(6): 1666–1677.
- [3] Xia D W, Jiang S Y, Yang N, et al. Discovering spatiotemporal characteristics of passenger travel with mobile trajectory big data. Physica A: Statistical Mechanics and Its Applications, 2021, 578(4): 126056.
- [4] Wei Z H, Dong J C, Dong Z. Research on prediction and early warning of A-share market volatility based on HAR-type models. Journal of Systems Science and Information, 2023, 11(6): 671–690.
- [5] Yang X K, Wang G L, Gao J. MaritimeDS: A data service framework for unsupervised maritime traffic monitoring based on trajectory big data. Journal of Reliable Intelligent Environments, 2022, 8(1): 1–17.
- [6] Xiao Y, Liu J J, Hu Y, et al. Time series forecasting using a hybrid adaptive particle swarm optimization and neural network model. Journal of Systems Science and Information, 2014, 2(4): 335–344.
- [7] Yu Z. Trajectory data Mining: An overview. ACM Transactions on Intelligent Systems and Technology, 2015, 6(3): 1–41.
- [8] Song R C, Sun W W, Zheng B H, et al. PRESS: A novel framework of trajectory compression in road networks. Proceedings of the VLDB Endowment, 2014, 7(9): 661–672.
- [9] Chen M J, Xu M T, Franti P. A fast $O(N)$ multiresolution polygonal approximation algorithm for GPS trajectory simplification. IEEE Transactions on Image Processing, 2012, 21(5): 2770–2785.
- [10] Popa I S, Zeitouni K, Oria V, et al. Spatio-temporal compression of trajectories in road networks. Geoinformatica, 2015, 19(1): 117–145.
- [11] Chen C, Ding Y, Xie X F, et al. TrajCompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(5): 2012–2028.
- [12] Sun S, Chen Y, Piao Z J, et al. Vessel AIS trajectory online compression based on scan-pick-move algorithm added sliding window. IEEE Access, 2020(8): 109350–109359.
- [13] Douglas D, Peucker T. 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, 1973, 10(2): 112–122.
- [14] Meratnia N, de By R A. Spatiotemporal compression techniques for moving point objects. Proceedings of the Advances in Database Technology-EDBT 2004, Springer, Berlin, Heidelberg, 2004: 765–782.
- [15] Lin C Y, Hung C C, Lei P R. A velocity-preserving trajectory simplification approach. Proceedings of the 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI), IEEE: Hsinchu, Taiwan, China, November 2016: 58–65.
- [16] Zhao L B, Shi G Y. A method for simplifying ship trajectory based on improved Douglas-Peucker algorithm. Ocean Engineering, 2018, 166(15): 37–46.
- [17] Singh A K, Aggarwal V, Saxena P, et al. Performance analysis of trajectory compression algorithms on marine surveillance data. Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE: Udupi, September 2017: 1074–1079.
- [18] Keogh E J, Chu S, Hart D, et al. An online algorithm for segmenting time series. Proceedings of the 2001 IEEE International Conference on Data Mining, IEEE Comput Soc, San Jose, CA, USA, 2001: 289–296.
- [19] Liu J, Zhao K, Sommer P, et al. A novel framework for online amnesic trajectory compression in resource-constrained environments. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(11): 2827–2841.
- [20] Potamias M, Patroumpas K, Sellis T. Sampling trajectory streams with spatiotemporal criteria. Proceedings of the 18th International Conference on Scientific and Statistical Database Management (SSDBM'06), IEEE: Vienna, Austria, 2006: 275–284.
- [21] Muckell J, Hwang J H, Patil V, et al. SQUISH: An online approach for GPS trajectory compression. Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications - COM.Geo'11, ACM Press: Washington, DC, 2011, 13: 1–8.
- [22] Alam M M, Torgo L. A clustering-based approach for predicting the future location of a vessel. Proceedings of the Canadian Conference on Artificial Intelligence, May 27, 2022: 1–12.

- [23] Carteron A, Jeanmougin M, Leprieux F, et al. Assessing the efficiency of clustering algorithms and goodness-of-fit measures using phytoplankton field data. *Ecological Informatics*, 2012, 9: 64–68.
- [24] Liu J, Zhao K, Sommer P, et al. Bounded quadrant system: Error-bounded trajectory compression on the go. *Proceedings of the 2015 IEEE 31st International Conference on Data Engineering*, IEEE: Seoul, South Korea, April 2015: 987–998.
- [25] Han Y, Sun W, Zheng B. COMPRESS: A comprehensive framework of trajectory compression in road networks. *ACM Transactions on Database Systems*, 2017, 42: 1–49.
- [26] Chen H, Chen X. A trajectory ensemble-compression algorithm based on finite element method. *ISPRS International Journal of Geo-Information*, 2021, 10(5): 334–351.
- [27] Ouyang Z, Xue L, Ding F, et al. PSOTSC: A global-oriented trajectory segmentation and compression algorithm based on swarm intelligence. *ISPRS International Journal of Geo-Information*, 2021, 10(12): 817–838.
- [28] Kang H, Liu Y, Zhang W. Cloud-based framework for spatio-temporal trajectory time series segmentation and query. *IEEE Transactions on Cloud Computing*, 2022, 10: 258–275.
- [29] Panagiotakis C, Pelekis N, Kopanakis I, et al. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 2011, 24(7): 1328–1343.
- [30] Junior A S, Times V C, Renso C, et al. A semi-supervised approach for the semantic segmentation of trajectories. *Proceedings of the 2018 19th IEEE International Conference on Mobile Data Management (MDM)*, IEEE: Aalborg, Denmark, June 2018: 145–154.
- [31] Despinoy F, Bouget D, Forestier G, et al. Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering*, 2015, 63(6): 1280–1291.
- [32] Hong Z, Chen Y, Mahmassani H S. Recognizing network trip patterns using a spatio-temporal vehicle trajectory clustering algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 2017, 19(8): 2548–2557.
- [33] Nie P, Chen Z, Xia N, et al. Trajectory similarity analysis with the weight of direction and K -neighborhood for AIS data. *ISPRS International Journal of Geo-Information*, 2021, 10: 757–776.
- [34] Etemad M, Soares A, Etemad E, et al. SWS: An unsupervised trajectory segmentation algorithm based on change detection with interpolation kernels. *Geoinformatica*, 2021, 25: 269–289.
- [35] Lee J G, Han J, Whang K Y. Trajectory clustering: A partition-and-group framework. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data - SIGMOD'07*, ACM Press: Beijing, China, 2007: 593–604.
- [36] Alewijnse S, Buchin K, Buchin M, et al. A framework for trajectory segmentation by stable criteria. *Proceedings of the 22nd ACM Sigspatial International Conference on Advances in Geographic Information Systems*, ACM: Dallas Texas, November 4, 2014: 351–360.
- [37] Buchin M, Driemel A, Van Kreveld M, et al. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Science*, 2011: 33–63.
- [38] Soares Júnior A, Moreno B N, Times V C, et al. GRASP-UTS: An algorithm for unsupervised trajectory segmentation. *International Journal of Geographical Information Science*, 2015, 29(1): 46–68.
- [39] Guo W Q, Zhao Z F, Li Z Y. REGRASP: A reactive GRASP based unsupervised trajectory segmentation method. *Journal of Northwest Normal University (Natural Science)*, 2020, 56: 44–52.
- [40] Tang C, Wang H, Zhao J, et al. A method for compressing ais trajectory time series based on the adaptive-threshold Douglas-Peucker algorithm. *Ocean Engineering*, 2021, 232: 109041.
- [41] Gao M, Shi G Y. Ship spatiotemporal key feature point online extraction based on AIS multi-sensor data using an improved sliding window algorithm. *Sensors*, 2019, 19(12): 2706.
- [42] Yang L, Yang G, Yin Y, et al. Sliding window-based region of interest extraction for finger vein images. *Sensors*, 2013, 13: 3799–3815.
- [43] Muckell J, Olsen P W, Hwang J H, et al. Compression of trajectory time series: A comprehensive evaluation and new approach. *Geoinformatica*, 2014, 18: 435–460.
- [44] Makris A, Silva C L, Bogorny V, et al. Evaluating the effect of compressing algorithms for trajectory similarity and classification problems. *Geoinformatica*, 2021, 25(4): 679–711.