



# A Semantics-Based Trajectory Segmentation Simplification Method

Minshi Liu<sup>1</sup> · Guifang He<sup>1</sup> · Yi Long<sup>2</sup>

Accepted: 18 September 2021 / Published online: 27 September 2021  
© The Author(s) 2021

## Abstract

With the development of mobile positioning technology, a large amount of mobile trajectory data has been generated. Therefore, to store, process and mine trajectory data in a better way, trajectory data simplification is imperative. Current trajectory data simplification methods are either based on spatiotemporal features or semantic features, such as road network structure, but they do not consider semantic features of a trajectory stop. To overcome this limitation, this study presents a trajectory segmentation simplification method based on stop features. The proposed method first extracts the stop feature of a trajectory, then divides the trajectory into move segments and stop segments based on the stop features, and finally simplifies the obtained segments. The proposed method is verified by experiments on personal trajectory data and taxi trajectory data. Compared with the classic spatiotemporal simplification method, the proposed method has higher spatiotemporal and semantic accuracy under different simplification scales. The proposed method is especially suitable for trajectory data with more stop features.

**Keywords** Trajectory simplification · Spatial–temporal characteristic · Stop features · Optics method

## Introduction

With the development of mobile positioning technologies, such as the Global Positioning System (GPS), Global System for Mobile Communications (GSM), and Radio Frequency Identification (RFID), a large number of mobile positioning devices with high positioning accuracy and low price have been proposed, including mobile phones, GPS collectors, and personal digital assistants (PDAs). As a result, a large amount of movement trajectory data has been generated, which brings difficulties in data storage and processing. For instance, in the T-Drive data set, there are 10,357 taxis, the sampling frequency is 5 s, and each record occupies 40 b (Yuan et al. 2010), so the amount of trajectory data of all taxi trajectory in Beijing city can reach 4 GB per day. Storing and indexing such massive data can cause high economic costs and low time efficiency, and it is challenging to process

massive trajectory data, mine hidden features, and extract spatiotemporal patterns in the data. Therefore, it is necessary to perform compression and simplification of trajectory data.

Most trajectory data simplification methods are offline or online simplification methods that use compression ratio and geometric feature preservation, including spatial features, spatiotemporal features, and velocity features, as a compression target.

Because trajectory data are commonly collected on the road network, a trajectory simplification method constrained by the road network and a trajectory data simplification method after map matching have been proposed (Kellaris et al. 2009, 2013; Popa et al. 2015). In this way, the trajectory reduction result is more in line with the real situation.

The common disadvantage of these two types of methods is that when the compression ratio is high, data simplification results may lose semantic features of the original data. To overcome this problem, a semantic trajectory simplification method has been proposed (Schmid et al. 2009; Richter and Schmid 2012). This method first extracts stops of a trajectory in geographical context and then abstractly expresses mobile trajectory to achieve the purpose of compression. Although this method has a high compression ratio, it reconstructs the trajectory through stops, so all the movement information between stops in the trajectory is lost.

---

✉ Minshi Liu  
liuminshi@chzu.edu.cn

<sup>1</sup> School of Geographic Information and Tourism, Chuzhou University, Chuzhou 239000, China

<sup>2</sup> Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing 210023, China

To address the abovementioned limitations, this study proposes a semantics-based trajectory segmentation simplification method (STSS). In this method, first the stop features of a trajectory are extracted first, then the trajectory is divided into stop segments and move segments based on the stop feature, and finally stop segment trajectories and move segment trajectories are simplified by their own methods respectively. The proposed method retains more spatiotemporal and semantic information of data while achieving high compression ratio.

The rest of the article is organized as follows. Section 2 reviews the related work on trajectory data simplification. Section 3 describes the proposed semantic-based trajectory segmentation simplification method. Section 4 verifies the proposed method by experimental tests and compares the data simplification result regarding different compression ratio. Finally, Sect. 5 draws conclusions about the applicability of the method.

## Related Works

### Trajectory Simplification Based on Spatiotemporal Features

The trajectory simplification method based on spatiotemporal features improves the general curve simplification method by constructing homomorphic spatial distance (Meratnia and de By 2003, 2004), spatiotemporal three-dimensional space (Trajcevski et al. 2006; Cao et al. 2006), or velocity features to improve the accuracy of simplification (Gudmundsson et al. 2009). These methods are aimed at maintaining high geometric accuracy and controlling the trajectory error (Muckell et al. 2014). They determine whether to retain or delete trajectory points according to the preset distance (position), angle (direction), and velocity (time) thresholds. They can be roughly divided into offline trajectory simplification methods and online trajectory simplification methods (Lee and Krumm 2011).

The main purpose of the offline trajectory simplification method is to compress trajectory data. The main idea is to retain more spatiotemporal information of data while reducing the amount of trajectory data. Meratnia and de By (2004) introduced the classic line feature simplification method, the Douglas-Peucker (D-P) method, into trajectory data compression for the first time. This method improves the D-P method by constructing the homomorphic space distance, proposing the D-P method of homomorphic distance named the top-down time-ratio (TD-DR) method. After that, a variety of methods have been developed on the basis of the TD-DR method and applied to various types of trajectory data simplification tasks (Zhao and Shi 2018).

Because of the dynamic and real-time characteristics of trajectory data, online trajectory simplification has become the focus of the current trajectory compression research. The simplification method based on deduced positioning (Trajcevski et al. 2006; Long et al. 2014) and the simplification method based on region filtering (Potamias et al. 2006; Gudmundsson et al. 2009) have been proposed. Both of these two methods are local optimization methods based on the trajectory data stream. Their main advantage is high efficiency, but their disadvantage is that the simplification accuracy cannot be guaranteed (Muckell et al. 2014). An online trajectory compression with controllable accuracy and compression ratio was proposed by Muckell et al. (2011). In this method, the queue is formed by the current trajectory point series, and the points with the minimum feature value are gradually deleted until the error threshold is exceeded or the compression ratio is reached. In addition, a new online trajectory simplification algorithm based on directed acyclic graph (OLTS) was proposed to apply to online services. This method represents an approximate optimal compression algorithm (Wu et al. 2017).

### Trajectory Simplification Based on Road Network

Considering that the trajectory is constrained by a road network, the road network space is used instead of a two-dimensional space, and the trajectory is simplified by structural characteristics of the road network (Li et al. 2008; Wu et al. 2015; Zhang et al. 2018), or it is simplified after map matching (Kellaris et al. 2013; Liu et al. 2014; Song et al. 2014). Among them, Li et al. (2008) extracted the characteristic trajectory points by combining the speed and direction characteristic information with the road network characteristic information for logical operation so as to simplify the trajectory data; Zhang et al. (2018) proposed an improved spatial-temporal trajectory compression method with constraints of a road network's structural features. The advantage of these methods is that the compressed trajectory can retain the characteristics of the road network, but the data does not match to the road network. Thus, a variety of trajectory compression strategies considering the road network constraints were proposed by Kellaris et al. (2009, 2013), including map matching, and different combinations of map matching and compression. However, the map matching accuracy after compression is low. Therefore, most studies aimed to match the road network first and then compressed the trajectory. The biggest disadvantage of this type of method is the low efficiency of the map matching algorithm.

## Semantic Trajectory Simplification

A concept of semantic trajectory compression was introduced by Schmid et al. (2009), wherein a semantic representation of a trajectory that consists of semantic locations associated with the trajectory stop features replaces the original trajectory points. Although the compression ratio of this method is very high, it only retains the trajectory points expressing the stops and deletes all the trajectory points in the moving state, so its simplification accuracy cannot be guaranteed. An enhanced semantic trajectory compression was proposed by Feng et al. (2013), wherein a semantic of a trajectory was represented by the speed change. Moreover, Yang et al. (2019) added semantic information to a trajectory through velocity clustering and then combined it with the trajectory space–time simplification method so as to effectively maintain the spatiotemporal characteristics and velocity characteristics of the trajectory. Their method is to hierarchical cluster all points on a single trajectory line based on velocity and each point becomes part of the clustering result, and then simplify trajectory according to the results of hierarchical clustering. In addition, Andrienko and Andrienko (2010) propose a spatial generalization and aggregation method of massive movement data for visualization. Their method can greatly compress data and extract features from data; however, their generalization and aggregation is not based on the trajectory line, but on the feature points after the transformation of all trajectory lines. Thus, in this paper, the proposed method takes a trajectory line as a unit to simplify the trajectory data. It extracts the stop features of the trajectory by clustering, and then the whole trajectory line is divided into the stop segments and the move segments for “divide and conquer” simplification.

## Proposed Method

### General Idea

As shown in Fig. 1, the general idea of the proposed method is as follows. Firstly, the multi-level stop features of the trajectory are extracted by improving the OPTICS method (Ankerst et al. 1999). Secondly, the trajectory is divided into stop segments and move segments according to the stop features. Thirdly, the stop segments and move segments of the trajectory are simplified by their own method. Finally, the simplified stop segment trajectories and move segment trajectories are merged into the whole trajectory.

### Stop Feature Extraction

Stop feature extraction is based on the clustering method of trajectory point string, which represents an improvement

of the OPTICS method. Similar to the OPTICS method, the trajectory points clustering algorithm also includes two steps: cluster-ordering of the trajectory points and clustering structure generation from cluster-ordering.

### Cluster-Ordering of Trajectory Points

In the trajectory point string, the distance between two points is no longer a straight-line distance between them but a sum of lengths of straight-line segments composed of a series of points between the two points.

**Definition 1: Distance between trajectory points** Assume  $P$  is a set of trajectory points, and  $p_i$  and  $p_j$  are trajectory points with sequences  $i$  and  $j$  in  $P$ , respectively; then, the distance between trajectory points  $p_i$  and  $p_j$ , can be calculated as follows:

$$td(p_i, p_j) = \sum_{k=1}^{j-1} d(p_k, p_{k+1}) \tag{1}$$

In the OPTICS algorithm, cluster-ordering requires searching for the  $\epsilon$ - neighborhood of the core point and calculating and sorting reachability-distances of all points in the  $\epsilon$ - neighborhood in every iteration. In this algorithm, because the trajectory point string is an ordered set, cluster-ordering of trajectory points does not require sorting reachability-distances of all points in the  $\epsilon$ - neighborhood of the core point but can directly use the original ordering of trajectory points.

Moreover, according to the calculation formula of a distance between trajectory points, the shortest distance between a point and the other points in the set is one of the distances between this point and its two adjacent points. Therefore, the reachability-distance of a point can be calculated only once. In addition, the  $\epsilon$ - neighborhood of the core point can be searched sequentially rather than searching all trajectory points.

The cluster-ordering process of the trajectory points is shown in Algorithm 1.

The proposed method traverses the trajectory point set  $P$  and calculates the reachability-distance of each point in the set  $P$ .

First, the algorithm computes the core-distance of the current point  $p_i$  by running the function  $CalculateCoreDistance(p_i, P, \epsilon, MinPts)$ , which first searches the points set  $PN$  of the  $\epsilon$ - neighborhood of  $p_i$  and then compares the number of points in set  $PN$  with  $MinPts$ ; if the number of points in set  $PN$  is less than  $MinPts$ ,  $c(p_i)$  is infinity; otherwise,  $c(p_i)$  is the maximum distance between  $p_i$  and points in the set  $PN$ .

Second, the reachability-distance of the next point  $p_{i+1}$  is calculated by running the function  $CalculateReachabilityDistance(p_i, c(p_i))$ , which is

dependent on the current point core-distance  $c(p_i)$  and whether the point is in the  $\varepsilon$ -neighborhood of  $p_i$ . If  $c(p_i)$  is not equal to infinity and  $p_j$  is in the  $\varepsilon$ -neighborhood of  $p_i$ ,

then  $r(p_j)$  is the maximum distance between  $c(p_i)$  and the linear distance between  $p_i$  and  $p_j$ ; otherwise,  $r(p_i)$  is infinity.

---

**Algorithm 1. Cluster-ordering algorithm of trajectory points**


---

**Input:** Trajectory points  $P = \{p_i | i = 1, 2, \dots, n\}$ , distance threshold  $\varepsilon$ , minimum number of points  $MinPts$

**Output:** The reachability-distances sequence of trajectory points (cluster-ordering)  
 $R = \{r(p_i) | i = 1, 2, \dots, n\}$

1:  $r(p_i) = \infty$ ; // Initialize  $r(p_0)$  as infinity

2:  $R.add(r(p_i))$ ;

3: **for**  $i = 1$  to  $(n-1)$  **do** // Traverse P until the last point

4:      $c(p_i) = CalculateCoreDistance(p_i, P, \varepsilon, MinPts)$ ; // Calculate core-distance of  $p_i$

5:      $r(p_{i+1}) = CalculateReachabilityDistance(p_{i+1}, c(p_i))$  ; // Calculate the reachability-distance of  $p_{i+1}$

6:      $R.add(r(p_{i+1}))$ ;

7: **end for**

8: **return**  $R$ ;

---

### Cluster-Structure Generation of Trajectory Points

The generation method of trajectory clustering structure is the same as the OPTICS method. In this method, the steepness point is first determined based on the steepness threshold, then the steepness area is extracted, and finally, the clustering structure is generated by matching the steep downward area and steep upward area that meet the clustering conditions. More detailed information on this method can be found in (Ankerst et al. 1999).

### Multi-Level Stop Feature Extraction of Trajectory

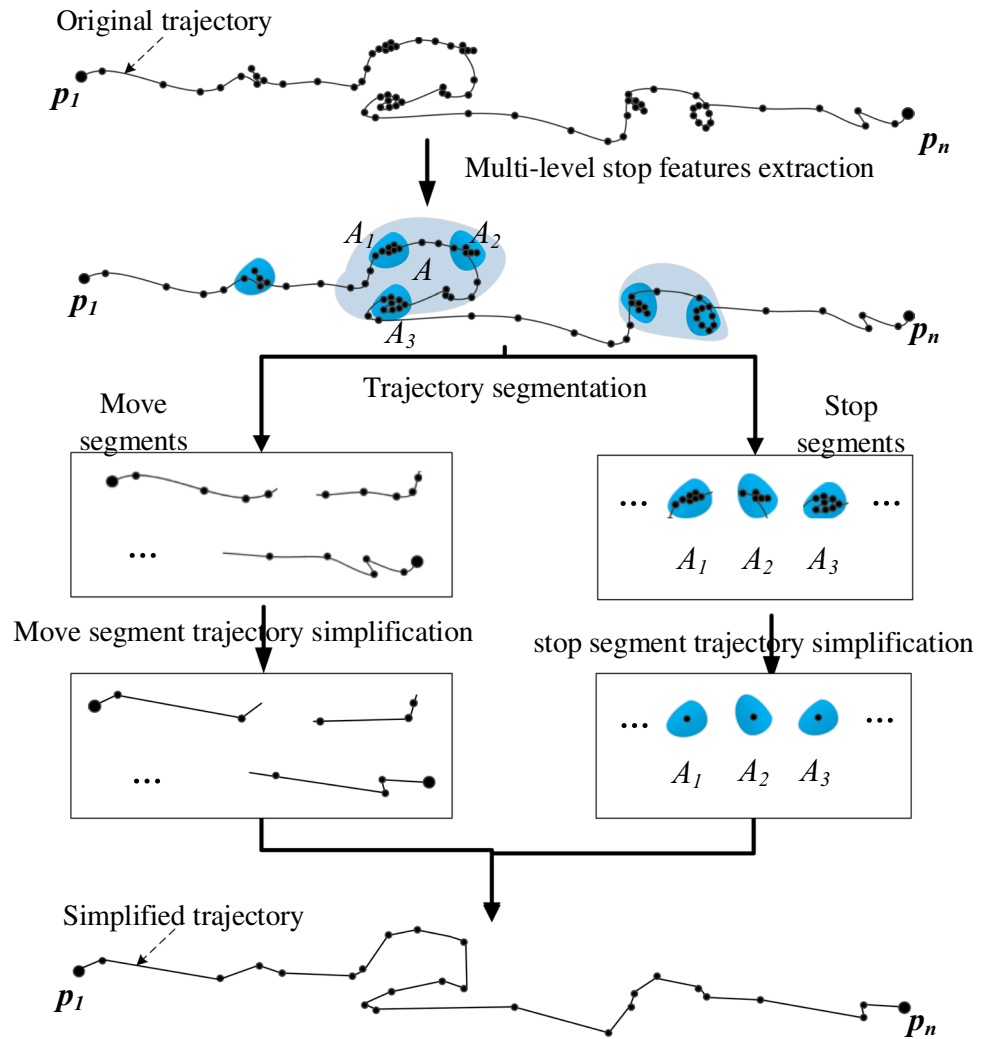
In the trajectory clustering structure, clusters are not completely independent of each other but can contain each other. There are two types of inclusion relationships between clusters: (1) A cluster contains only one cluster, and the two clusters belong to the same stop feature, so one cluster can be deleted. As shown in Fig. 2a,  $C_3$  and  $C_2$  are clustering, where  $C_3$  contains  $C_2$  and they represent the same stop feature, so  $C_2$  is deleted. (2) A cluster contains more than one cluster, and they belong to different stop features. As shown in Fig. 2a,  $C_4$ ,  $C_1$ , and  $C_3$  are clustering, where  $C_4$  includes  $C_1$  and  $C_3$ , and  $C_1$  and  $C_3$  are different stop features, so they should be retained. Thus, the hierarchical

relationship between them can be represented by a tree structure in Fig. 2b.

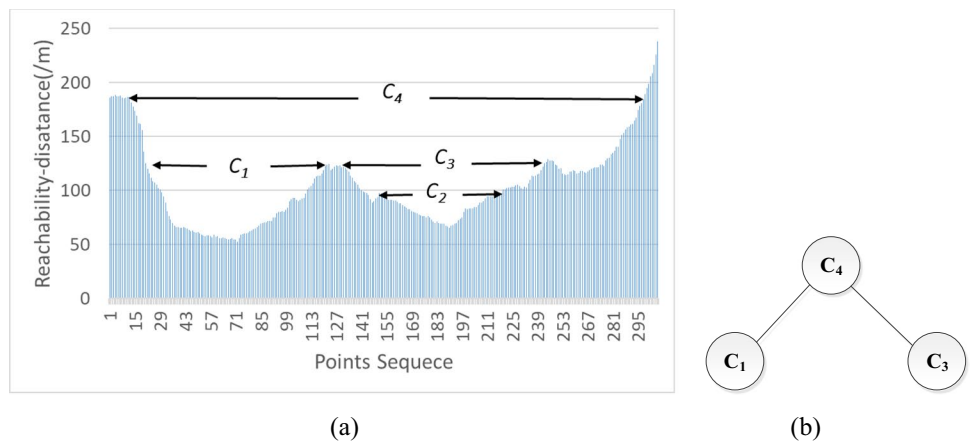
The multi-level stop feature extraction algorithm of the trajectory is shown in Algorithm 2. The algorithm input is a trajectory clustering set  $C$ ;  $c$  is a cluster in set  $C$ , which is represented by  $(P, s, e)$ , where  $P$  denotes the trajectory point string of the cluster, and  $s$  and  $e$  are the positions of the start and end points of the cluster in the original trajectory point string, respectively. The algorithm output is the trajectory stop segment tree set  $N$ ;  $n$  is a stop segment node in the set  $N$ , and it is a tree node represented by  $(c, childNodes)$ , where  $c$  denotes a cluster and  $childNodes$  stands for all child nodes of node  $n$ .

The algorithm first initializes the global clustering range  $(gs, ge)$  as empty and then traverses cluster set  $C$ . Next, it is determined whether the intersection of the global range and the current clustering range and the intersection of the current clustering range and the subsequent clustering range are empty. If both of them are empty, then the current cluster is added to set  $N$  as a tree node, and the global range is the current cluster range. If the former is not empty, but the latter is empty, then the current cluster is added to set  $N$  as a tree node, and the child nodes of the node are found in set  $N$ , then the global range is set to the current cluster range; otherwise, it will not be processed.

**Fig. 1** An overview of the processing workflow



**Fig. 2** The inclusion relationships between clusters and the tree structure of stop features. **a** Two types of inclusion relationships between clusters; **b** the tree structure of corresponding to **a**



**Algorithm 2 Multi-level stop feature extraction algorithm of the trajectory**


---

**Input:** Trajectory cluster-structure set  $C = \{c_i | i = 1, 2, \dots, n\}$   
**Output:** Trajectory stop segment set  $N = \{n_i | i = 1, 2, \dots, n\}$

- 1:  $(gs, ge) = (\infty, -\infty)$ ; // Initialize  $(gs, ge)$  as  $\emptyset$
- 2: **for**  $i = 1$  to  $(n-1)$  **do** // Traverse  $P$  until the last point
- 3:     **if**  $((s_i, e_i) \cap (gs, ge) == \emptyset \ \&\& \ (s_i, e_i) \cap (s_{i+1}, e_{i+1}) == \emptyset)$
- 4:          $n = ConstructTreeNode(c_i)$
- 5:          $(gs, ge) = (s_i, e_i)$
- 6:          $N.add(n)$
- 7:     **else if**  $((s_i, e_i) \cap (gs, ge) != \emptyset \ \&\& \ (s_i, e_i) \cap (s_{i+1}, e_{i+1}) == \emptyset)$
- 8:          $n = ConstructTreeNode(c_i)$
- 9:          $(gs, ge) = (s_i, e_i)$
- 10:          $nlist = FindChildNodesofn(N, n)$
- 11:          $n.childs = nlist$
- 12:          $N.add(n)$
- 13:          $N.delete(nlist)$
- 14:     **End if**
- 15: **end for**
- 16: **return**  $N$ ;

---

### Three Thresholds Setting

In the proposed method, there are three thresholds: distance threshold  $\varepsilon$ , number of points threshold  $MinPts$ , and steepness threshold  $\xi$ . The distance threshold determines the minimum density of a cluster, the number of points threshold determines the minimum number of points in a cluster, and the steepness threshold determines the minimum difference in the density between a cluster and its surrounding scattered points. Therefore, the first two thresholds affect cluster-ordering of a trajectory, whereas the third threshold affects cluster-structure generation. Ankerst et al. (1999) have been suggested that similar results can be obtained using different ranges of  $\varepsilon$  and  $MinPts$ , as long as the value of the two threshold is not too small.

In the proposed method, the distance threshold  $\varepsilon$  represents the minimum moving range of a trajectory stop segment. The trajectory stop feature does not necessarily mean that the moving object stops; it can still move but at a slow speed. Therefore, the distance threshold is expressed as a product of the residence time and moving speed in the trajectory stop feature.

The point number threshold  $MinPts$  denotes the minimum number of points in the stop segment of a trajectory. The point number threshold of a trajectory can be expressed as

a ratio of the residence time to the sampling frequency of trajectory points.

The steepness threshold parameter  $\xi$  represents a difference between the density of the stop segment and the density of the move segment of a trajectory. It is affected by the moving mode of a moving object. Generally, a person's moving mode includes walking, riding, and traveling by car, train, or other means of transportation, so steepness should be set according to the specific mode of transportation.

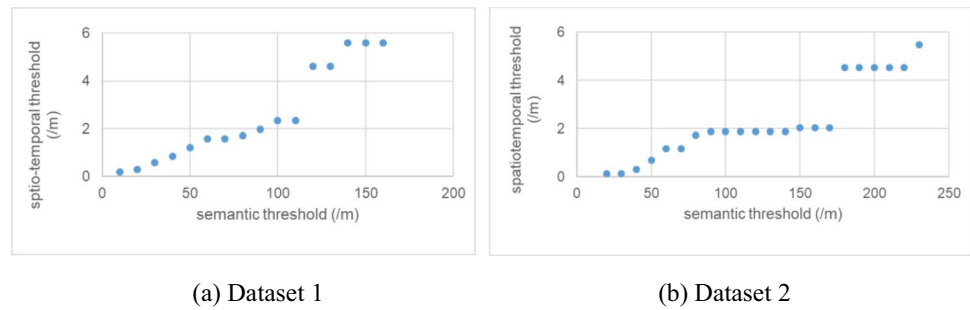
### Simplification of Trajectory Stop Segments

#### Simplification Method of Single Stop Segment of Trajectory

Since the trajectory stop segment is the stop feature of a location, it can be expressed using a point. For this simplified point, two factors need to be considered. First, the point should be as close as possible to the center of the trajectory stop segment, and second, the point should be the original point in the points set of the trajectory stop segment. Therefore, the point is calculated by the following method. First, the center of the point series in the trajectory stop segment is calculated, and then the distances from the point series in the trajectory stop segment to the center point are compared; the point with the smallest distance is taken as a simplified point.



**Fig. 3** The relationship between the semantic threshold and spatiotemporal threshold



**Simplification Method of Multiple Stop Segments of Trajectory**

Owing to the hierarchical relationship between stop segments, it is necessary to merge multiple stop segments when the degree of simplification is increased. The key of merging stop segments is to find stop segments that need to be merged. According to the multi-level stop segments established using the method described in Sect. 3.2.3, the tree structure of stop segments is formed. Based on the hierarchical tree structure of stop segments, the relationship between the reachable distance threshold and stop segments can be established. Therefore, as long as a certain reachable distance is given, the stop segments under the current distance threshold can be obtained.

**Simplification of Trajectory Move Segment**

Simplification method of trajectory move segment adopts the road network constrained moving trajectory simplification method (Zhang et al. 2018), which is to construct binary line generalization (BLG) tree and sort all trajectory points according to the spatial–temporal characteristics of the trajectory and the structure characteristics of the road network. The method can preserve both the spatiotemporal and the road structure characteristic of original trajectory at the same time.

**Trajectory Simplification**

Since the trajectory stop segment simplification and trajectory move segment simplification use their simplification thresholds to quantify the simplification scale, it is necessary to establish a quantitative relationship between the stop segment simplification threshold (semantic threshold) and the move segment simplification threshold (spatiotemporal threshold) based on the same simplification scale.

In this study, the function fitting method is used to establish the relationship between the two thresholds. First, the scatter diagram between the two thresholds is constructed based on the simplification scale, and then the polynomial function model is used for fitting.

The scatter plots of the two thresholds for dataset 1 and dataset 2 are presented in Fig. 3, where it can be seen that there is a linear relationship between the two thresholds. Therefore, the linear functional model is used to fit the relationship between the two thresholds. Let  $y$  be the semantic threshold, and  $x$  be the spatiotemporal threshold; then, the function model fitted by dataset 1 is defined as  $y = 23x + 26.5$ , and the function model fitted by dataset 2 is defined as  $y = 37.3x + 38.8$ .

**Evaluation Method**

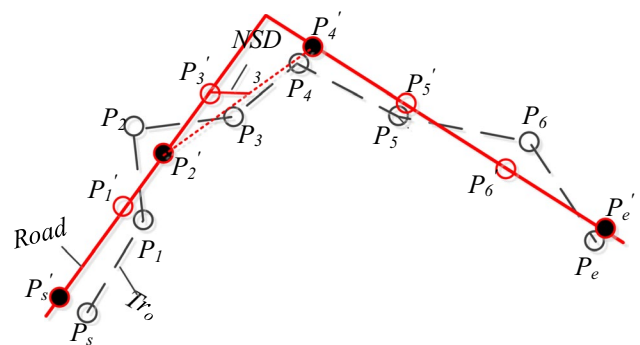
The quality evaluation indexes of trajectory simplification include spatial–temporal accuracy and semantic accuracy.

**Spatial–Temporal Accuracy Evaluation**

Since a trajectory is usually distributed on a road network, spatial–temporal accuracy is evaluated by network homomorphic distance error (Zhang et al. 2018). The network homomorphic distance error is calculated by Eq. (2) and illustrated in Fig. 4.

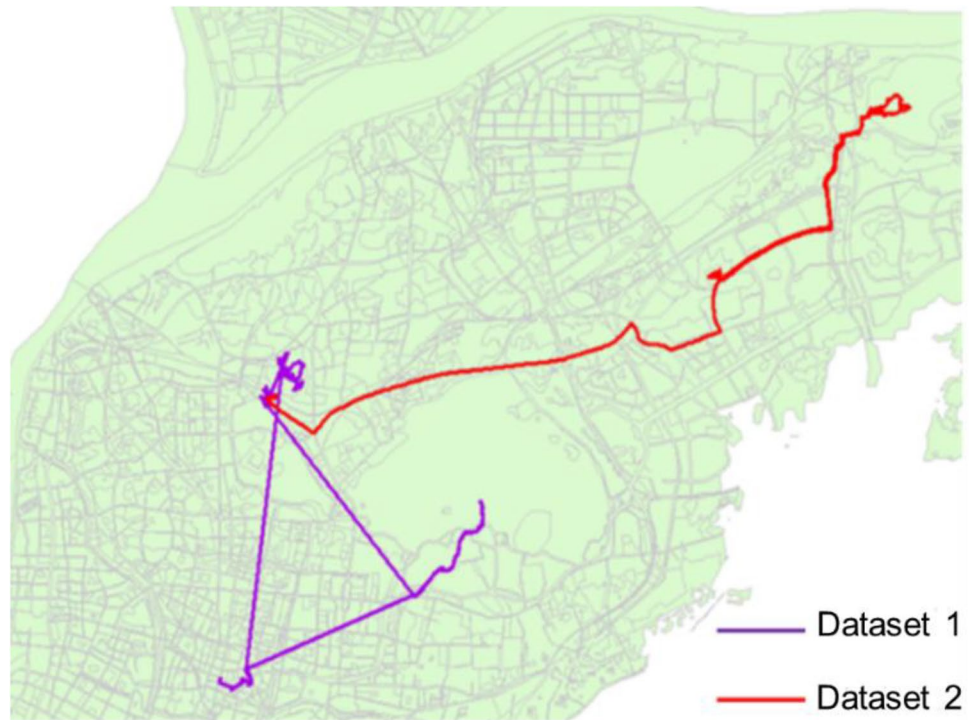
$$TSTA(tra_s, tra_o) = \frac{1}{n} \sum_{i=1}^n nhd_i \tag{2}$$

In Eq. (2),  $tra_o$  denotes the original trajectory,  $tra_s$  denotes the simplified trajectory,  $n$  is the number of points in  $tra_s$ ,



**Fig. 4** Network homomorphic distance error

**Fig. 5** Experimental data; the purple trajectory line denotes dataset 1, and the red trajectory line denotes dataset 2



and  $nhd_i$  represents a distance between trajectory point  $p_i$  and its homomorphic point in the road network.

**Semantic Accuracy Evaluation**

The semantic accuracy evaluation is to extract the stop features of a simplified trajectory and to compare the result of stop features with that of the original trajectory. The semantic accuracy is calculated by:

$$TSEA(tra_s, tra_o) = \frac{NS(tra_s)}{NS(tra_o)}, \tag{3}$$

where  $NS(tra_o)$  and  $NS(tra_s)$  are the numbers of stop features extracted from the original trajectory and from the simplified trajectory, respectively.

**Experiments and Results**

**Experiments on Personal Trajectory Data**

**Experimental Data**

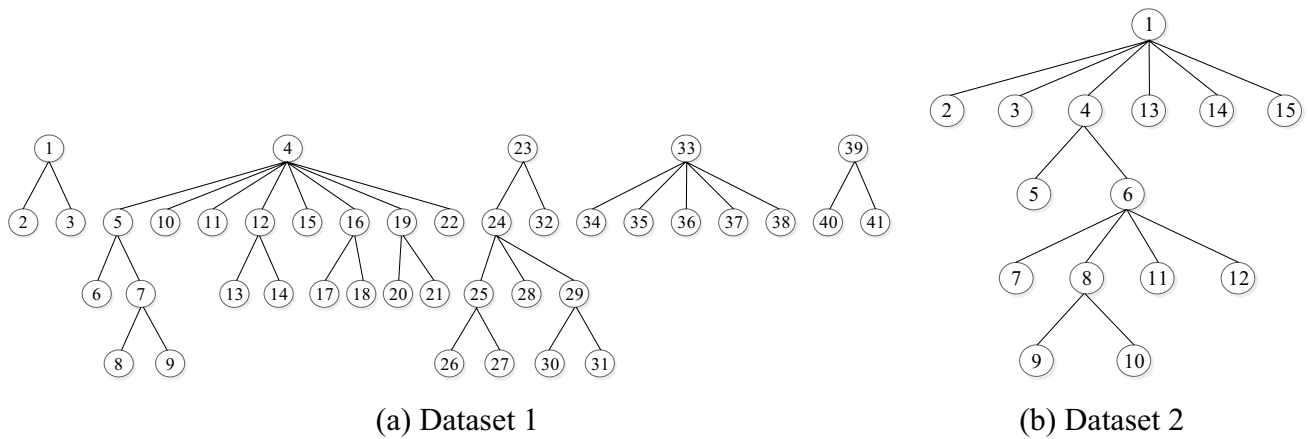
The experimental data were the data of two personal GPS trajectories in the city of Nanjing (Fig. 5), which can be downloaded from the shared database (<https://figshare.com/s/6582b3f6b4906ddc5564>). The details of the data are shown in Table 1. The sampling interval of dataset 1 was 5 s, and the total duration was approximately 17 h; this dataset included 8606 trajectory points with a length of 66,404 m. The sampling interval of dataset 2 was also 5 s, but the total duration was approximately 7 h; the dataset included 3602 trajectory points with a length of 26,553 m.

Stop features should be extracted from trajectory data before simplification of this proposed method. Therefore, for the experimental data, the three thresholds were set as follows: Considering that the minimum length of stop should not be shorter than 5 min, and the trajectory sampling

**Table 1** Details of the experimental data

ID	Sampling interval (s)	Number of trajectories	Number of trajectory points	Length (m)	Total duration
Dataset 1	5	2	8,606	66,404	17 h and 45 min
Dataset 2	5	1	3,602	26,553	7 h and 22 min

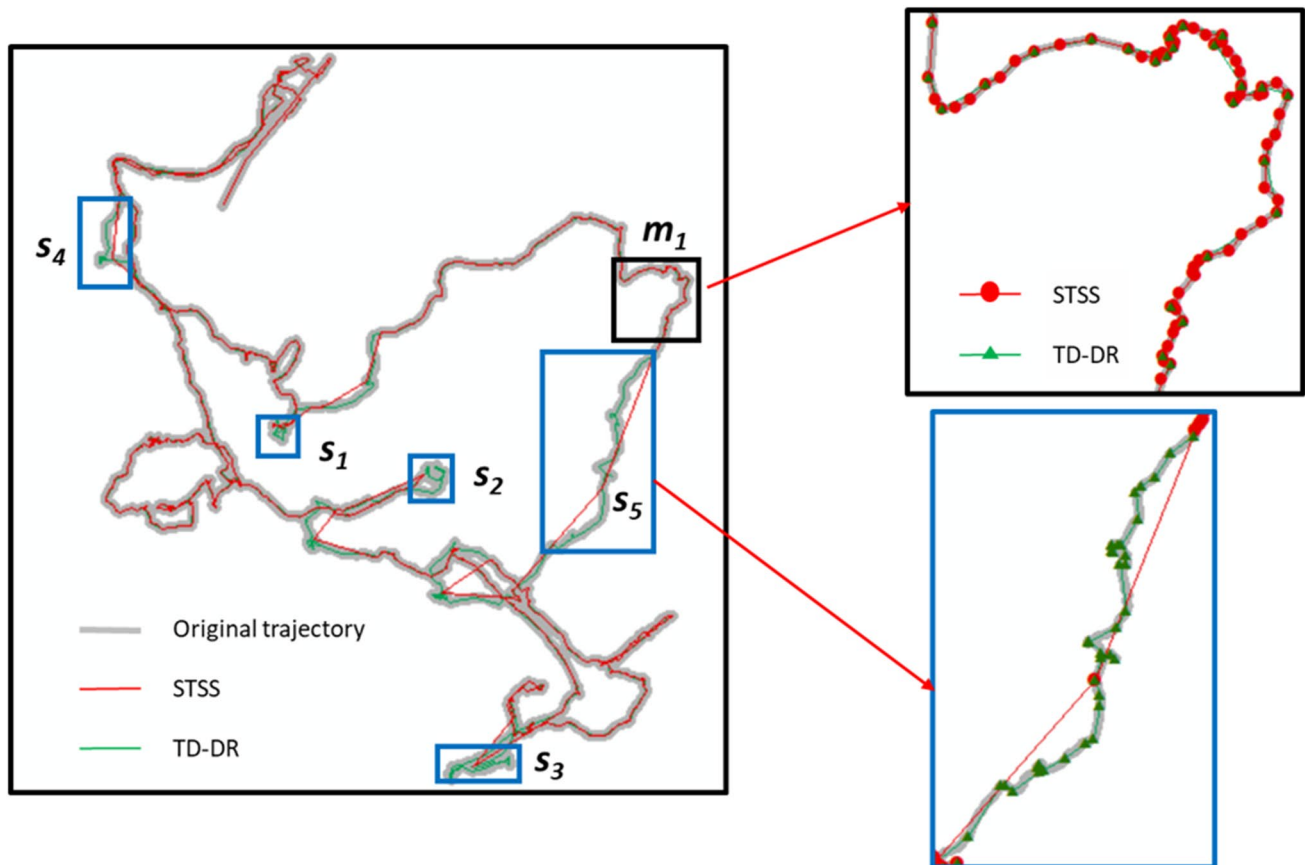




**Fig. 6** The tree structure of stop feature extracted from the experimental data under  $e = 1$  km,  $minpts = 60$ , and  $n = 0.02$

frequency was 5 s; *MinPts* was set to 60. Since it is generally believed that the speed of the stop state should not exceed 1 m/s, the distance threshold  $e$  should be greater than 300 m. Although a large distance threshold could provide better clustering results, an excessive threshold might cause

a calculation burden to the algorithm, so the distance threshold was set to 1000 m based on practical experience. Finally, according to the experiment, the slope threshold  $\xi$  was set to 0.02.



**Fig. 7** Visual analysis results between the TD-DR method and the STSS method in partial dataset 1. Simplification threshold for TD-DR method is 1.8 m, spatiotemporal threshold and semantic threshold for STSS method are 0.5 m and 38 m, respectively

The tree structure of stop feature extracted from the experimental data is shown in Fig. 6. As shown in Fig. 6, 41 stop features were extracted from experimental dataset 1, and they were divided into 4 levels; 15 stop features were extracted from experimental dataset 2, and they were divided into 5 levels.

### Experimental Design

In the experiment, the accuracies of the proposed method and the TD-DR method [7] were compared. The analysis was performed on the series scale of simplification, and the analysis results were compared from two perspectives, the simplification threshold, and the compression ratio.

Simplification threshold is composed of spatiotemporal threshold and semantic threshold. Since the linear relationship between spatiotemporal threshold and semantic threshold is established, the threshold of simplification is represented by spatiotemporal threshold. In the simplification threshold-based analysis, it was necessary to select the appropriate series of simplification thresholds. Seven spatiotemporal threshold values of 0.25 m, 0.5 m, 1 m, 2 m, 4 m, 8 m, and 16 m were selected and used in multiple experiments.

In the compression ratio-based analysis, to obtain the same compression ratio in the two methods, the processing method of “intelligent damping oscillation” was adopted (Liu et al. 2016). The basic idea of this method is to intelligently adjust the threshold value through a variable step size

until the simplification result is consistent with the preset compression ratio.

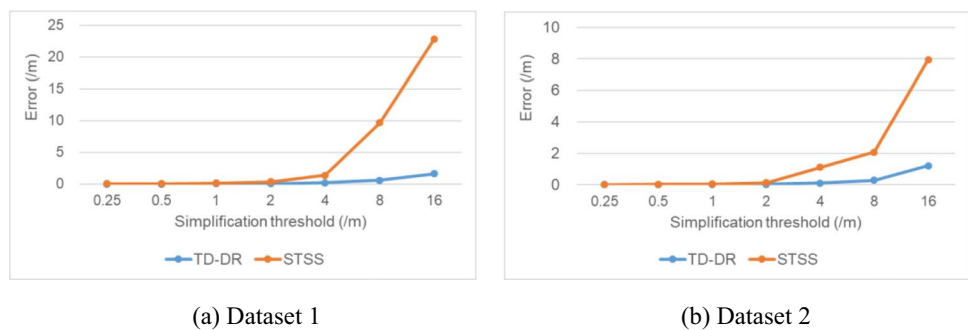
Assume  $dis$  is an initial threshold,  $step$  is an initial step of threshold adjustment,  $rat_o$  is a target compression rate,  $rat_c$  is a current compression rate, and  $tol$  is a tolerance of the target compression rate.

The procedure for this method is as follows: trajectory was simplified by the threshold  $dis$  and the  $rat_c$  is calculated, and then let  $diff = rat_c - rat_o$ . If  $|diff| \leq tol$ , end the adjustment; otherwise, modify  $dis$  and simplify trajectory again: if  $diff < 0$  and last  $diff < 0$ , then  $dis = dis + step$ ; if  $diff < 0$  and last  $diff > 0$ , then  $step = step/2, dis = dis + step$ ; if  $diff > 0$  and last  $diff > 0$ , then  $dis = dis - step$ ; and if  $diff > 0$  and last  $diff < 0$ , then  $step = step/2, dis = dis - step$ .

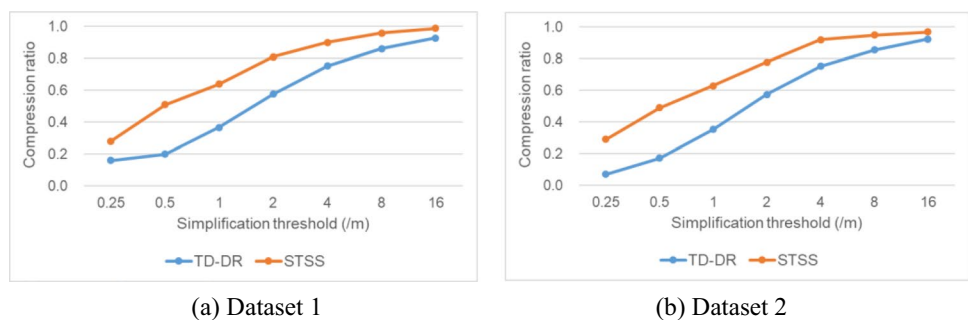
### Visual Analysis Result

The results of visual analysis are simplified by the TD-DR method and the STSS method with a compression ratio of 50%. As shown in Fig. 7, this method directly simplifies multiple trajectory points with high density, namely the stop segment trajectories (e.g.,  $s1, s2, s3, s4, s5$ ) to a single point, while the TD-DR method retains more of these trajectory points; however, the STSS method retains more trajectory points than the TD-DR method for trajectory points with low density, namely the move segment trajectory (e.g.,  $m1$ ). Some stop segments look like move segments (e.g.,  $s4, s5$ ). This is because moving objects move very slowly and then they are identified as stop features. Therefore, compared

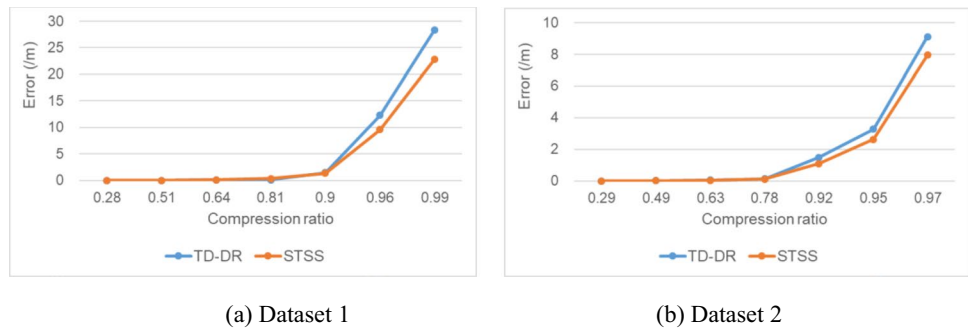
**Fig. 8** Error comparison between the STSS and TD-DR methods under different simplification threshold values



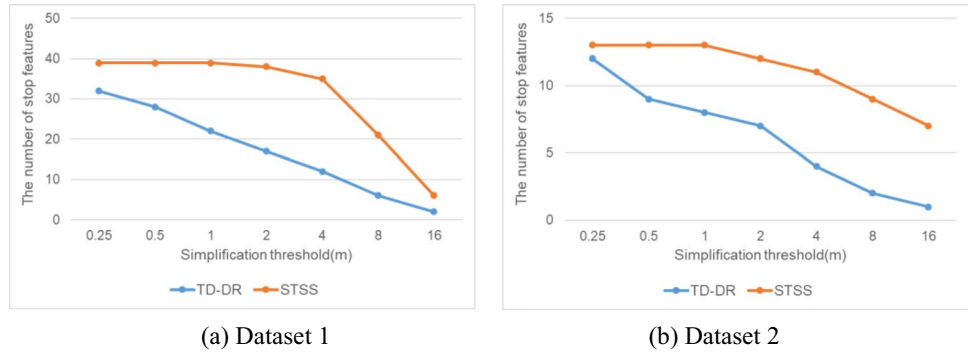
**Fig. 9** Compression ratio comparison between the STSS and TD-DR methods under different simplification threshold values



**Fig. 10** Error comparison between the STSS and TD-DR methods under different compression ratios



**Fig. 11** The comparison of the number of stop features between the STSS and TD-DR methods under different simplification threshold values



with the TD-DR method, the STSS method compresses a large number of feature points in the stop segment trajectory, and retain more feature points in the of the move segment trajectory.

**Spatial–Temporal Accuracy Analysis Result**

The results of the spatial–temporal accuracy comparison of the two methods based on the simplification threshold are shown in Fig. 8, where it can be seen that on the two datasets, the accuracy of TD-DR method was higher than that of the proposed method under the same threshold, and their accuracy difference increased with the threshold; and when the threshold was small, the accuracy difference between the two methods was very small, but when the threshold increased to a certain value (e.g., 4 m for dataset 1, 2 m for dataset 2), the accuracy difference expands rapidly.

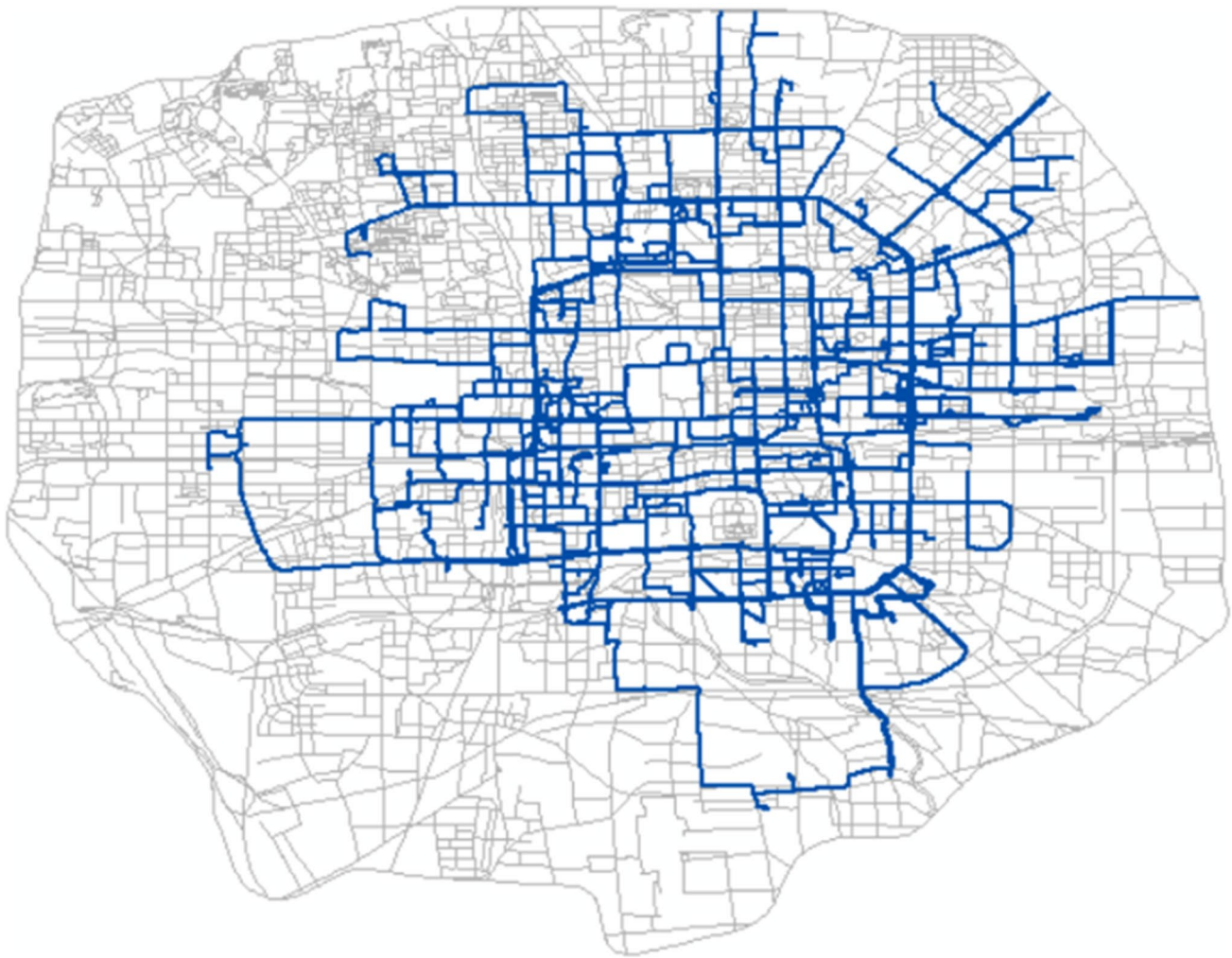
The above-presented comparison denotes a precision comparison based on the simplification threshold, which does not necessarily mean that the TD-DR method performs better than the proposed method. This result could be because although the thresholds of the two methods were the same, their simplification scales differed. The results of the simplification scale (compression ratio) of the two methods on experimental datasets under different simplification threshold values are presented in Fig. 9, where it can be seen that the compression ratio of the proposed method was significantly higher than that of the TD-DR method under the

same threshold value. The compression ratio of the proposed method was nearly twice that of the TD-DR method when the simplification threshold value was 0.5 m.

The spatial–temporal accuracy of the two methods was analyzed under different compression ratios, and the results are shown in Fig. 10. On the whole, the accuracy of the proposed method was higher than that of the TD-DR method. When the compression ratio was small, the accuracy difference between the two methods was also small. However, when the compression ratio was high (e.g., 0.9 for dataset 1, and 0.78 for dataset 2), the proposed method had a smaller error and higher accuracy than the TD-DR method.

**Semantic Accuracy Analysis Result**

The semantic accuracy comparison results of the two methods under different spatiotemporal threshold values are shown in Fig. 11, where it can be seen that compared with the TD-DR method, the proposed method extracted more stop features and achieved better semantic accuracy under different thresholds. The accuracy gap between the two methods first increased and then decreased with spatiotemporal scale value. In addition, since the proposed method had a higher compression ratio than the TD-DR method at the same threshold, the gap between the two methods was large at the same compression ratio.



**Fig. 12** Experimental data; the blue line is the taxi trajectory, and the gray line is the road network

### Experiments on Taxi Trajectory Data

Since most of the stop features in the taxi trajectory data are caused by getting on and off passengers or waiting for traffic lights, compared with the personal track data, the number of stop features in the taxi trajectory data are smaller, and the stay time of each stop feature is shorter.

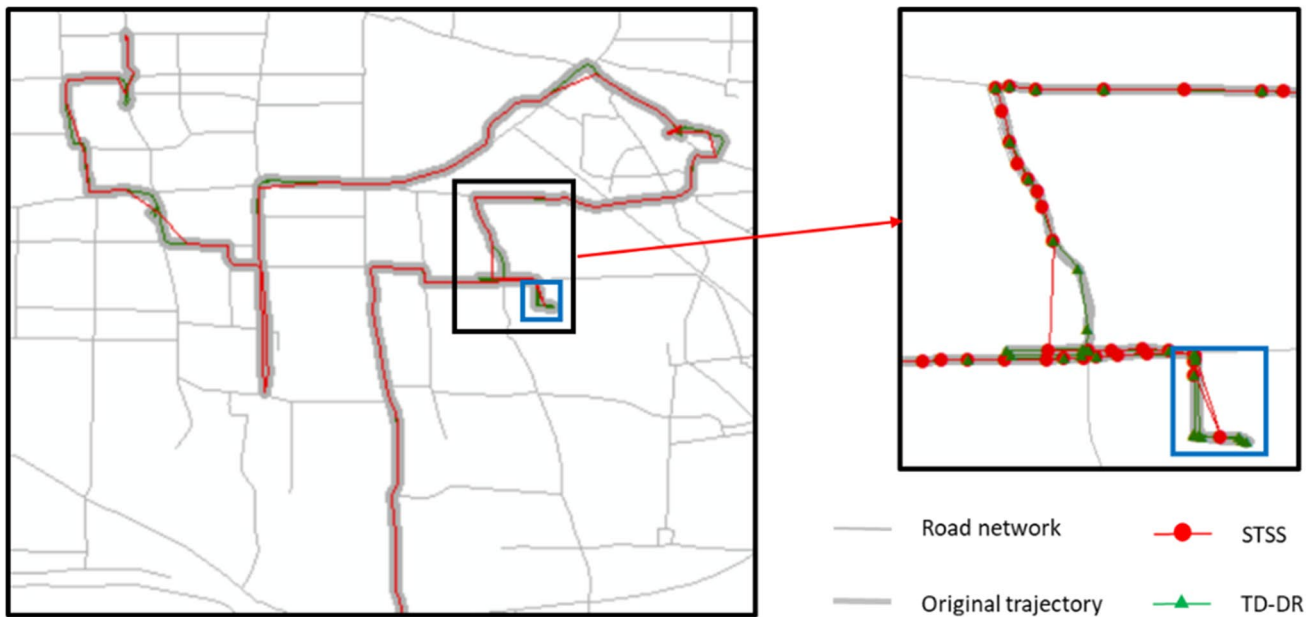
#### Experimental Data

The experimental data is one taxi trajectory data, which is selected from the taxi GPS trajectories dataset during the period of 2–8 February 2008 within Beijing (Yuan et al. 2010), as shown in Fig. 12. The sampling interval of the taxi trajectory dataset is 5 s, and the total number of the trajectory points are 30,156.

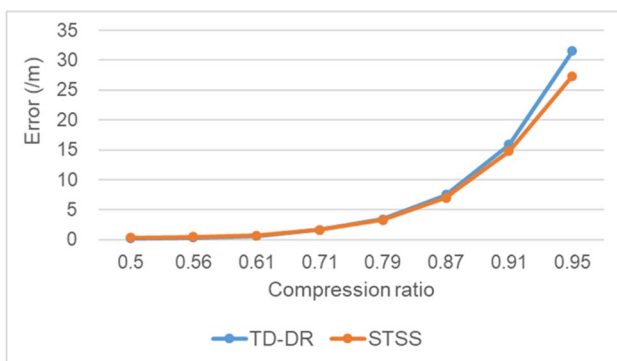
Similarly, the three thresholds in the stop feature extraction need to be set. Most taxi stops are caused by getting

on and off passengers or waiting for traffic lights, so the minimum length of stop should not be shorter than 1 min, and the trajectory sampling frequency is 5 s; *MinPts* is set to 12. Since it is generally believed that the speed of the stop state should not exceed 1 m/s, the distance threshold  $\epsilon$  should be greater than 60 m. Finally, according to the experiment, the slope threshold  $\xi$  was set to 0.02. Therefore, 233 stop features are extracted from this dataset, which are divided into two layers only; among them, there are 217 stop features of leaf nodes.

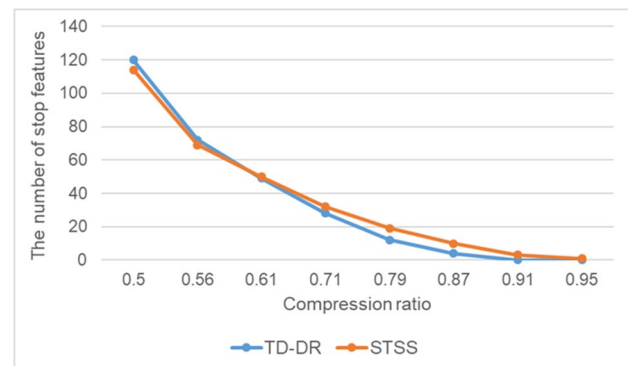
The experimental analysis was compared between the STSS method and the TD-DR method on multiple compression ratios. The process is similar to the previous experiment. Firstly, eight spatiotemporal threshold values of 0.5 m, 1 m, 2 m, 5 m, 10 m, 25 m, 50 m, and 100 m were selected. Secondly, the trajectory was simplified by STSS method on these thresholds, and the corresponding eight compression



**Fig. 13** Visual analysis results between the TD-DR method and the STSS method in partial taxi trajectory data. Simplification threshold for TD-DR method is 10 m; spatiotemporal threshold and semantic threshold for STSS method are 2 m and 50 m, respectively



**Fig. 14** Error comparison between the STSS and TD-DR methods under different compression ratios in taxi trajectory



**Fig. 15** The comparison of the number of stop features between the STSS and TD-DR methods under different compression ratios in taxi trajectory

ratios are obtained. Finally, the trajectory was simplified by TD-DR method on eight compression ratios.

**Visual Analysis Result**

The results of visual analysis are simplified by the TD-DR method with a 10-m simplification threshold and the STSS method with a 2-m spatiotemporal threshold and a 50-m semantic threshold. As shown in Fig. 13, similar to the previous experiment, compared with the TD-DR method, the STSS method compresses a large number of feature points in the stop segment trajectory and retains more feature points in

the of the move segment trajectory. In addition, as shown in this figure, most of the stop features of trajectory are located at road intersections, which are caused by vehicles stopping for traffic signals at the intersection.

**Spatial–Temporal Accuracy Analysis Result**

The spatial–temporal accuracy of the two methods was analyzed under different compression ratios, and the results are shown in Fig. 14. When the compression ratio was small, the accuracy difference between the two methods was also small. However, when the compression ratio was high, the



STSS method is better than the TD-DR method, and the accuracy gap increases with the increase of compression.

### Semantic Accuracy Analysis Result

The semantic accuracy comparison results of the two methods under different compression ratios are shown in Fig. 15, where it can be seen that the semantic accuracy of both methods decreases with compression ratio, but the TD-DR method decreases faster. When the compression ratio is less than 0.61, the TD-DR method is better than the STSS method; otherwise, the STSS method is better than the TD-DR method. The main reason is that in the STSS method, the stop features are simplified when the simplification threshold is very small, so fewer semantic features are extracted after simplification.

### Conclusion

This study proposes a semantic-based trajectory segmentation simplification method, which extracts stop features first and then performs segmentation simplification. The proposed method is verified by the experiments and compared with the classis spatiotemporal simplification method, the TD-DR method. Based on the comparison results, the following conclusions can be drawn:

- (1) The relationship between the semantic threshold and spatiotemporal threshold under the same simplification scale is linear. The parameter value of the linear functional model is determined by the experimental data.
- (2) The compression ratio of the STSS method is obviously higher than that of the TD-DR method under the same simplification threshold, and the difference first increases and then decreases with threshold value.
- (3) The spatiotemporal accuracy of the STSS method is slightly lower than that of the TD-DR method under the same simplification threshold. However, the STSS method has a smaller error and higher spatiotemporal accuracy than the TD-DR method under the same compression ratio, especially for a large simplification scale.
- (4) Compared with the TD-DR method, the proposed STSS method can retain more stop features and has higher semantic accuracy. Obviously, there is a large performance difference between the two methods under the same compression ratio.
- (5) According to the experimental analysis of personal trajectory data and taxi trajectory data, the proposed method can be applied to different types of trajectory data, but it is better for trajectory data with more stop features (e.g., travel trajectory).

In the future, research on compression and simplification of trajectories could be conducted from the perspective of trajectory semantics mining. It should be noted that the purpose of trajectory simplification is not only to reduce the amount of data but also to extract trajectory characteristics at different scales to consider different application scenarios.

**Acknowledgements** The authors would like to thank Dr. Zhang Ling for his help in the language of the article.

**Authors' Contributions** Conceptualization, YL and ML. Methodology, ML and GH. Formal analysis, ML. Writing—original draft preparation, ML. Writing—review and editing, ML and GH. Supervision, YL.

**Funding** This research was funded by the National Natural Science Foundation of China, grant number 41601499, and Chuzhou University Research Start-up Foundation, grant number 2020qd48. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

### Declarations

**Compliance with Ethical Standards** The authors certify that this manuscript has never been published. No data have been fabricated or manipulated (including images) to support our conclusions.

**Ethical Approval** The experimental of this manuscript does not contain any researches with human participants or animals.

**Informed Consent** This manuscript has been submitted with the consent of all authors.

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

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### References

- Andrienko N, Andrienko G (2010) Spatial generalization and aggregation of massive movement data. *IEEE Trans Vis Comput Graph* 17(2):205–219
- Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) OPTICS: ordering points to identify the clustering structure. *ACM SIGMOD Rec* 28(2):49–60
- Cao H, Wolfson O, Trajcevski G (2006) Spatio-temporal data reduction with deterministic error bounds. *VLDB J* 15(3):211–228
- Feng S, Xu J, Xu M, Zheng N, Zhang X (2013) EHSTC: An enhanced method for semantic trajectory compression. *Acm Sigspatial*

- International Workshop on Geostreaming, ACM Press, New York, pp 43–49
- Gudmundsson J, Katajainen J, Merrick D, Ong C, Wolle T (2009) Compressing spatio-temporal trajectories. *Comput Geom* 42(9):825–841
- Kellaris G, Pelekis N, Theodoridis Y (2009) Trajectory compression under network constraints. In *International Symposium on Spatial and Temporal Databases*, Springer, Berlin, Heidelberg, pp 392–398
- Kellaris G, Pelekis N, Theodoridis Y (2013) Map-matched trajectory compression. *J Syst Softw* 86(6):1566–1579
- Lee WC, Krumm J (2011) Trajectory preprocessing. In: Zheng Y, Zhou X (eds) *Computing with spatial trajectories*. Springer, New York, pp 3–33
- Li X, Lin H, Guo ZY (2008) Reducing vehicle tracking data volume through a network-based approach. *Acta Geodaet Cartograph Sin* 37(1):95–101
- Liu K, Li Y, Dai J, Shang S, Zheng K (2014) Compressing large scale urban trajectory data. *Proceedings of the Fourth International Workshop on Cloud Data and Platforms* ACM Press, New York, USA, pp 1–6
- Liu MS, Long Y, Fei LF (2016) Line simplification of three-dimensional drainage considering topological consistency. *Acta Geodaet Cartograph Sin* 45(4):494–501
- Long C, Wong RCW, Jagadish HV (2014) Trajectory simplification: on minimizing the direction-based error. *Proc VLDB Endowment* 8(1):49–60
- Meratnia N, de By RA (2003) A new perspective on trajectory compression techniques. In *Proc. ISPRS Commission II and IV, WG II/5, II/6, IV/1 and IV/2 Joint Workshop Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis*
- Meratnia N, de By R A (2004) Spatiotemporal compression techniques for moving point objects. In *Advances in Database Technology - EDBT 2004* Springer, Berlin, Heidelberg, pp 765–782
- Muckell J, Hwang JH, Patil V, Lawson CT, Ping F, Ravi SS (2011) SQUISH: an online approach for GPS trajectory compression. In *Proceedings of the 2nd international conference on computing for geospatial research & applications, com.geol'11*, Washington DC, USA, pp 1–8
- Muckell J, Olsen PW, Hwang JH, Lawson CT, Ravi SS (2014) Compression of trajectory data: a comprehensive evaluation and new approach. *GeoInformatica* 18(3):435–460
- Popa IS, Zeitouni K, Oria V, Kharrat A (2015) Spatio-temporal compression of trajectories in road networks. *GeoInformatica* 19(1):117–145
- Potamias M, Patroumpas K, Sellis T (2006) Sampling trajectory streams with spatiotemporal criteria. In *18th International Conference on Scientific and Statistical Database Management (SSDBM'06)*, IEEE Computer Society, Vienna, Austria, pp 275–284
- Richter K, Schmid F (2012) Laube P (2012) Semantic trajectory compression: representing urban movement in a nutshell. *J Spat Inf Sci* 4:3–30
- Schmid F, Richter KF, Laube P (2009) Semantic trajectory compression. In *International Symposium on Spatial and Temporal Databases*, Springer, Berlin, Heidelberg, pp 411–416
- Song R, Sun W, Zheng B, Zheng Y (2014) Press: a novel framework of trajectory compression in road networks. *Proc vldb Endowment* 7(9):661–672
- Trajcevski G, Cao H, Scheuermann P, Wolfson O, Vaccaro D (2006) On-line data reduction and the quality of history in moving objects databases. In *Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, ACM, Chicago, Illinois, USA, pp 19–26
- Wu P, Tan YA, Zheng J, Zhang Q, Li Y, Cheng Z (2015) A hybrid compression framework for large scale trajectory data in road networks. *Chin J Electron* 24(4):730–739
- Wu F, Fu K, Wang Y, Xiao Z (2017) A graph-based min-# and error-optimal trajectory simplification algorithm and its extension towards online services. *ISPRS Int J Geo Inf* 6(1):19
- Yang M, Yan X, Zhang X, Li X (2019) Constrained trajectory simplification with speed preservation. *Cartogr Geogr Inf Sci* 47(2):110–124
- Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun, G, Huang Y (2010) T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, ACM, New York, NY, USA, pp 99–108
- Zhao L, Shi G (2018) A method for simplifying ship trajectory based on improved douglas-peucker algorithm. *Ocean Eng* 166(10):37–46
- Zhang L, Liu M, Long Y (2018) Trajectory compression with constraints of road networks. *Int Arch Photogramm Remote Sens Spat Inf Sci* 42:4

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.