A FLEXIBLE TRAJECTORY COMPRESSION ALGORITHM FOR MULTI-MODAL TRANSPORTATION

S. S. Mirvahabi[1]*, R. Ali Abbaspour[1], C. Claramunt[2]

[1] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran
– {mirvahabi.s, abaspour}@ut.ac.ir
[2] Naval Academy Research Institute Lanveoc-Poulmic, BP 600, 29240 Brest, France– christophe.claramunt@gmail.com

Commission IV, WG IV/3

KEYWORDS: Trajectory Compression, Douglas–Peucker Method, Transportation Mode, Flexible Douglas–Peucker Method, FDP Compression Method.

ABSTRACT:
Continuous progress in navigation, sensor-based, and GPS technologies have made smart devices essential to our daily lives and many location-based applications. However, the trajectory datasets generated by these applications require the management of large data volumes while preserving their main properties and semantics. One of the most popular methods for compressing trajectory data offline is the Douglas–Peucker (DP) algorithm, but its principles should be applied to a diverse range of contexts when considering real-time trajectory data. This paper introduces a Flexible Douglas-Peucker algorithm (FDP) that takes into account the data's diversity, underlying properties, and semantics. The proposed framework is applied to the Geolife benchmark dataset with a series of different thresholds that reflects different contexts and constraints when performing a trajectory compression process. The results show that the proposed algorithm achieves a significant compression rate while preserving trajectory data points that have a semantic role concerning different modes of transportation.

1. INTRODUCTION

Recent advances in navigation and positioning have made smart devices (such as smartphones, tablets, and wearable gadgets) essential to our daily lives (Yang, Stewart, Tang, Xie, and Li, 2018). GPS positional data collected by these devices support a wide range of applications such as route planning, anomaly detection, and decision-making (Zhang, Zhao, and Liu, 2022; Zhao and Shi, 2019). However, data collection processes are typically performed in real-time with usually very short time intervals, resulting in very large data volumes. These data are usually formalized and stored as sequences of annotated trajectories. A problem that immediately arises is that raw trajectories contain a large amount of duplicate data. Therefore, storing, retrieving, managing, processing, and querying these trajectories is computationally expensive and requires large storage spaces (Nasiri, Azimi, and Abbaspour, 2018; Zhao and Shi, 2019). A critical step in trajectory representation and pre-processing is to reduce incoming data volumes while preserving the main properties and semantics associated with these trajectories and resulting movement patterns (Makris, Kontopoulos, Alimisis, and Tserpes, 2021).

Over the past few years, several compression methods have been proposed to reduce the volume of trajectory data. These methods can be categorized into two groups: offline and online. When applying offline methods, compression is applied after trajectory data collection and according to preselected spatial, temporal, and semantic parameters. In online methods, trajectory data selection is processed on the fly then reducing data storage but increasing computational time (Sun, Xia, Yuan, and Li, 2016). Although offline methods often result in higher accuracy, online methods are usually preferred to avoid the generation of large data volumes and to facilitate further data processing and mining (Zhang et al., 2022).

This paper introduces a Flexible Douglas–Peucker (FDP) for trajectory compression and whose peculiarity is to consider the semantic and spatial dimensions. The rest of this paper is organized as follows. Section 2 gives the motivation of this research while Section 3 briefly outlines related work. The proposed framework is introduced in Section 4. Section 5 describes the implementation and evaluation of the results, while finally section 6 draws the conclusions and a few perspectives for further work.

2. MOTIVATIONS

One of the first methods applied for compressing trajectory data offline is the Douglas–Peucker (DP) algorithm (Douglas and Peucker, 1973). The DP algorithm reduces the number of points in a curve that is approximated by a series of points. It is based on the deviation distance of the trajectory curve from the straight line connecting the starting point and the endpoint of this curve, suitable points of the trajectory are selected and maintained. Then the trajectory from these selected points is divided into two sub-trajectories. This process continues recursively and hierarchically until the distance of the mentioned deviation is less than the threshold of the distance determined at the beginning of the process. The distance used is the Perpendicular Euclidean Distance (PED). shows the

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-4-W1-2022-501-2023 | © Author(s) 2023. CC BY 4.0 License.

successive steps of the DP method to compress one single trajectory (from a to d).

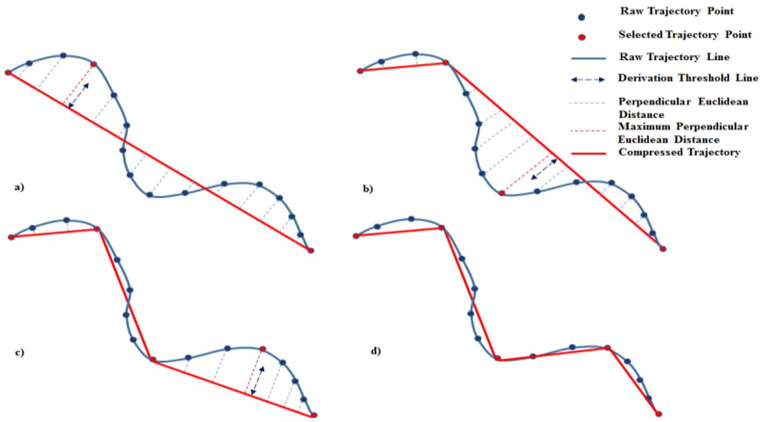

Raw Trajectory Point
Selected Trajectory Point
Raw Trajectory Line
Derivation Threshold Line
Perpendicular Euclidean Distance
Maximum Perpendicular Euclidean Distance
Compressed Trajectory
a)
b)
c)
d)

Figure 1. Douglas Peucker compression algorithm (a-d)

A limitation of the DP algorithm is that it relies on a constant threshold to determine the minimum deviation to maintain spatial points along the trajectory (Xiao-li and De, 2010). This constant threshold is suitable when the trajectory properties are solely spatio-temporal and not specifically associated with additional semantics. When representing multi-modal trajectories in urban environments, this is not the case as most human routes combine several modes of transportation with different speeds and then spatio-temporal constraints. This implies considering variable thresholds during the compression processes. A conventional application of the DP algorithm with a constant threshold to a multi-modal trajectory is illustrated in Figure 2. Based on the usual DP method and setting a constant threshold for the entire trajectory, several points in the 'Walk' mode (points B, E, H) that embed useful semantic information have been removed.

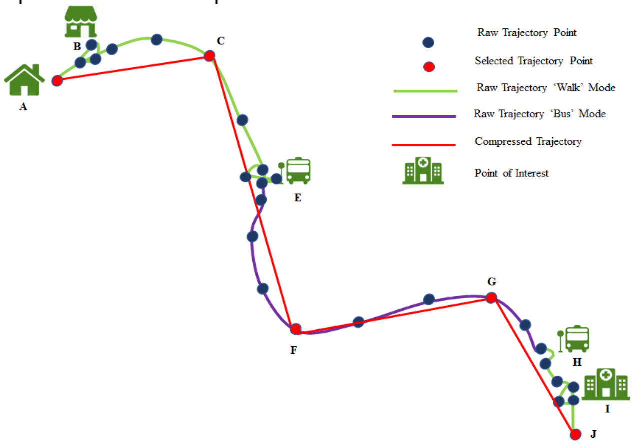

Raw Trajectory Point
Selected Trajectory Point
Raw Trajectory 'Walk' Mode
Raw Trajectory 'Bus' Mode
Compressed Trajectory
Point of Interest
A
B
C
E
F
G
H
I
J

Figure 2. Constant threshold limitation

3. RELATED WORK

This section describes some of the existing methods developed for trajectory compression. Tobler (Tobler and Geovisualization, 1989) developed a trajectory compression method known as Uniform Sampling Algorithm specifically applied to map generalization. The principle is to keep the most representative trajectory points from a cartographical point of view. Bellman (Bellman, 1961) proposed a dynamic compression algorithm for fitting line segments to trajectory curves. These line segments are chosen in such a way that they have the best fit and connect the main trajectory points. One of the most applied algorithms for line compression is the Douglas Peucker algorithm (Douglas, Peucker, and geovisualization, 1973). This algorithm applies a recursive process in which points are selected based on the amount of deviation they create in the straight line connecting the starting point and the endpoints. The Douglas Peucker algorithm's main limitation is that it is primarily based on the trajectory spatial dimension. Meratnia and de By (Meratnia and de By, 2003) have extended the Douglas-Peucker so-called Top-Down Time-Ratio algorithm that integrates the time dimension. A time synchronized Euclidean distance is used to calculate the deviation. The Opening Window algorithm (Keogh, Chu, Hart, and Pazzani, 2001) is still based on the Douglas Peucker algorithm and applies a moving window, where on each trajectory point, the amount of deviation is calculated, compared to the threshold, and

https://doi.org/10.5194/isprs-annals-X-4-W1-2022-501-2023 | © Author(s) 2023. CC BY 4.0 License.

when this deviation is greater than the threshold, the aforementioned point is saved. This point is considered the first point of the next moving window and this process continues until the end point of the trajectory. Similarly, Wu and Cao (Wu and Cao, 2002) suggested an Opening Window Time-Ratio algorithm based on a synchronized Euclidean distance. Additional spatial constraints such as speed and direction of the trajectory points have been considered by Potamias et al (Potamias, 2006) and the STTrace algorithm. Trajcevski et al. (Trajcevski, Cao, Scheuermanny, Wolfsonz, and Vaccaro, 2006) presented an online algorithm, Dead Reckoning, which selects the most appropriate points with the location and velocity of each trajectory point. The SQUISH method (Muckell et al., 2011) selects the most important points locally for the compressed trajectory by prioritizing the trajectory points and using a fixed-size buffer. The SQUISH-E method (Muckell, Olsen, Hwang, Lawson, and Ravi, 2014) is an extension of the previous method in which, in addition to prioritizing the points, the minimization of the synchronized Euclidean distance error is also considered. Stop points can be also considered to extract the main semantics of urban trajectories (Hosseinpoor, Abbaspour and Claramunt, 2018). Overall, most of existing research consider a constant threshold or allowed error value, for all trajectory points and all trajectories in the dataset. To the best of our knowledge, none of these works take into account variable thresholds or diverse error values influenced by semantic parameters in addition to spatial parameters.

4. PROPOSED METHODOLOGY

We introduce the principles of a Flexible DP algorithm (FDP) applied to a multi-modal trajectory, in which appropriate thresholds for each part of a multi-modal trajectory are based on the semantic properties of the associated transportation modes. The main steps of the proposed framework are briefly illustrated in Figure 3 and hereafter described.

First, at the pre-processing stage, the raw trajectory is examined to remove outliers and noisy data. At this stage, incomplete data and data without time stamps are removed from the dataset. Then, raw trajectory data is enriched by semantic information related to respective transportation modes.

Secondly, and to reflect different contexts, trajectory data are also annotated by specific user data and temporal data (e.g., day of occurrence). The displacement between two consecutive trajectory points is calculated based on the Haversine formula (Kerley, 1965):

$$\Delta lat_i = lat_i - lat_{i-1} \quad (1)$$

$$\Delta lon_i = lon_i - lon_{i-1} \quad (2)$$

$$D_i = 2R\sqrt{sin^2(\frac{\Delta lat_i}{2}) + \cos(lat_{i-1})\cos(lat_i)\, sin^2(\frac{\Delta lon_i}{2})} \quad (3)$$

where R denotes the Earth radius and lat_i, lat_{i-1}, lon_i, lon_{i-1} successive trajectory points

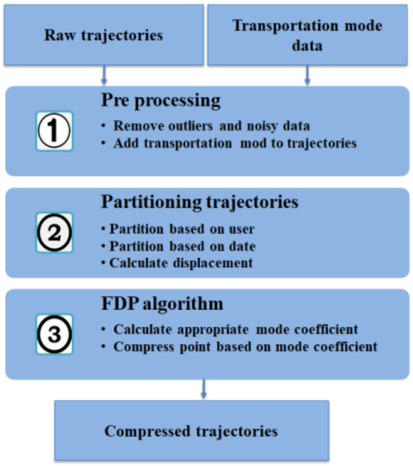

Raw trajectories
Transportation mode data
1
Pre processing
• Remove outliers and noisy data
• Add transportation mod to trajectories
2
Partitioning trajectories
• Partition based on user
• Partition based on date
• Calculate displacement
3
FDP algorithm
• Calculate appropriate mode coefficient
• Compress point based on mode coefficient
Compressed trajectories

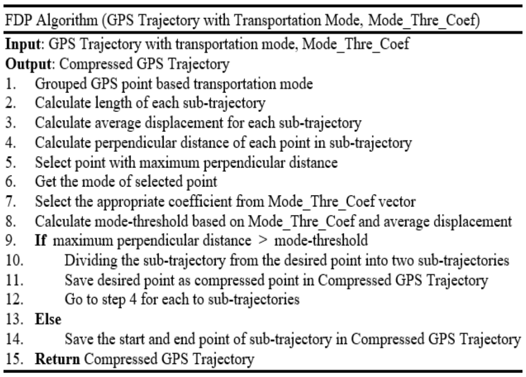
Figure 3. Proposed Framework

Thirdly, appropriate scale factors and thresholds are determined for each transportation mode and associated with the FDP algorithm. Therefore, the average displacement for each mode of transportation and each user per day is calculated as follows:

$$\bar{D}_j = (\sum_{i=1}^{n_j} D_i)/n_j \quad (4)$$

where $\bar{D}_j$ = Average displacement of the j^th^ mode
D_i = Displacement of ith point in the j^th^ mode
n_j = Number of points in the j^th^ mode

Since the amount of displacement for each transportation mode is likely to be specific to it, every threshold should be specific too to each mode. However, as the displacement rate and the threshold have a direct relationship, the threshold value should be considered as a suitable coefficient of average displacement rate for each mode of transportation. Finally, the compression algorithm is applied, and the most relevant points generated by the FDP algorithm are kept. Figure 4 depicts the pseudocode of the generalized FDP algorithm. The output of the framework is a compressed trajectory data set based on spatial, and semantic parameters, and that reflects the specific properties of the multi-modal and contextual trajectory properties.



```
FDP Algorithm (GPS Trajectory with Transportation Mode, Mode_Thre_Coef)
Input: GPS Trajectory with transportation mode, Mode_Thre_Coef
Output: Compressed GPS Trajectory
1.  Grouped GPS point based transportation mode
2.  Calculate length of each sub-trajectory
3.  Calculate average displacement for each sub-trajectory
4.  Calculate perpendicular distance of each point in sub-trajectory
5.  Select point with maximum perpendicular distance
6.  Get the mode of selected point
7.  Select the appropriate coefficient from Mode_Thre_Coef vector
8.  Calculate mode-threshold based on Mode_Thre_Coef and average displacement
9.  If maximum perpendicular distance > mode-threshold
10.     Dividing the sub-trajectory from the desired point into two sub-trajectories
11.     Save desired point as compressed point in Compressed GPS Trajectory
12.     Go to step 4 for each to sub-trajectories
13. Else
14.     Save the start and end point of sub-trajectory in Compressed GPS Trajectory
15. Return Compressed GPS Trajectory
```

Figure 4. FDP Algorithm

5. IMPLEMENTATION AND EVALUATION

The trajectories used to implement the proposed framework are taken from the Geolife benchmark dataset (Zheng, Li, Chen, Xie, and Ma, 2008; Zheng, Xie, and Ma, 2010; Zheng, Zhang, Xie, and Ma, 2009). The Geolife dataset was collected by 182 users over 5 years from 2007 to 2011. A large part of this data is recorded in the city of Beijing in China. The GPS points of this collection contain information such as longitude and latitude, time, date of collection, user index, and transportation mode. The data are collected using different GPS devices and mobile phones and have different sampling rates. Several modes of transportation are available such as 'walk', 'bus', 'train', 'taxi', and 'car'.

Daily trajectories were selected representing 4326 points and four transportation modes (bus, train, walk, taxi) for implementation purposes, but the proposed algorithm can be extended to the entire data set and all transportation modes. Figure 5 shows a trajectory sample used for the implementation, where the transportation modes are shown in different colors.

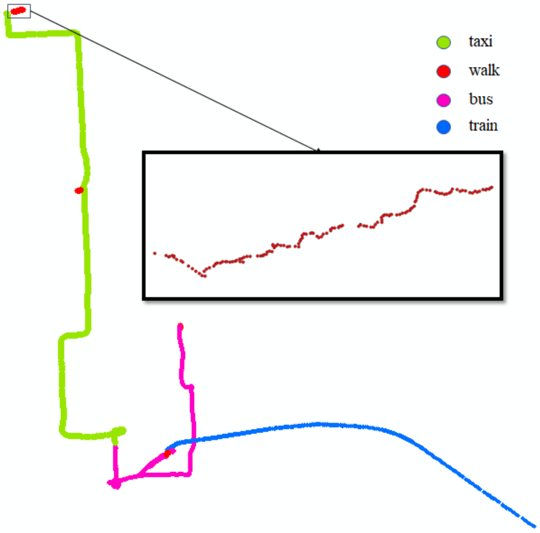


Figure 5. Case study

To approximate potential thresholds for each transportation, vector coefficients are defined according to common displacement speeds for each transportation mode, as these represent levels of precision that should be considered when applying the FDP algorithm (Table 1). Accordingly, for each transportation mode, the product of these respective coefficients with the average distance in each transportation mode generates the threshold value. The higher the velocity of the moving average, the higher the possibility of moving it per second. Therefore, to calculate the threshold limit of the coefficient, a higher value should be considered for the mentioned threshold limit. These values have been calculated as optimal values after repeating the algorithm several times.

The proposed framework has been compared to current DP algorithms to illustrate its performance. First, trajectories were compressed using the DP algorithm with different thresholds (i.e., 20 m, 50 m, 100 m, 200 m). The results obtained from these three different thresholds are shown in Figures 6-9. Next, these trajectories were also compressed using the FPD algorithm for comparison purposes.

Transportation Mode	Coefficient
Walk	4
Run	5
Bus	7
Train	9
Car	8
Taxi	8
Bicycle	6
Motorcycle	7

Table 1.Transportation mode coefficients

Figure 10 shows the output of the proposed algorithm for the mentioned trajectory. As it is seen in Figures 6-9, regarding the trajectories compressed by the DP algorithm, except for the start and endpoints of the sub-trajectory with transportation mode 'walk', no other point of this sub-trajectory is preserved. The exception is the compressed trajectory with a threshold limit of 20 meters, where a middle point of the path with the mode 'walk' is preserved. However, when considering the 'walk' mode, changes in the user's location are semantically more important than in the 'train' or 'bus' transportation modes. Therefore, semantically, the more points of this mode are preserved, the compressed trajectory gives a better generalization for reconstructing the original trajectory and a more suitable representative for extracting additional information and knowledge. On the other hand, for other transportation modes such as 'bus' and 'train' where the average displacement is higher and small changes are not very important, the number of compressed points can be lower compared to the 'walk' mode. Using the semantic information of the transportation mode, it appears that the flexible threshold has significantly improved the quality and performance of the FDP algorithm in maintaining the points of 'walk 'mode during the compression process. Moreover, in other modes of transportation, the overall shape of the trajectory is preserved with minimal compressed points (Figure 10).

To compare the proposed algorithm with the DP algorithm, the trajectory compression rate (Zheng and Zhou, 2011) criterion is used. The compression rate is the ratio of the number of compressed trajectory points to the number of original trajectory points, it is given as follows.

$$CR = \frac{|N_C|}{|N_O|} \tag{4}$$

Where $CR =$ Compression Rate
N_C = Compressed trajectory points
N_O = Original trajectory points

The compression rate results of the DP algorithm with four different thresholds and our algorithm (FDP algorithm) are given in Table 2.

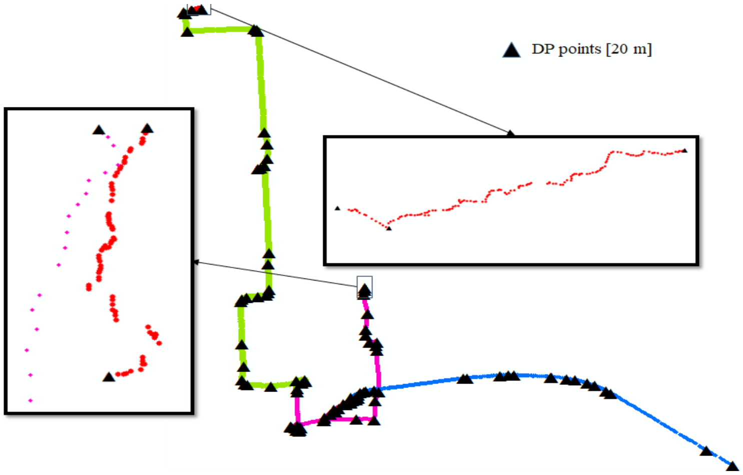


Figure 6. DP trajectory compression with a threshold of 20 m

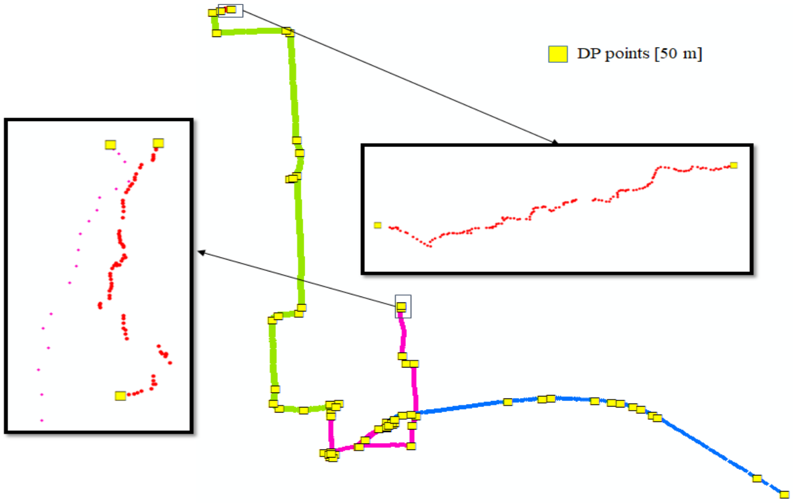


Figure 7. DP trajectory compression with a threshold of 50 m

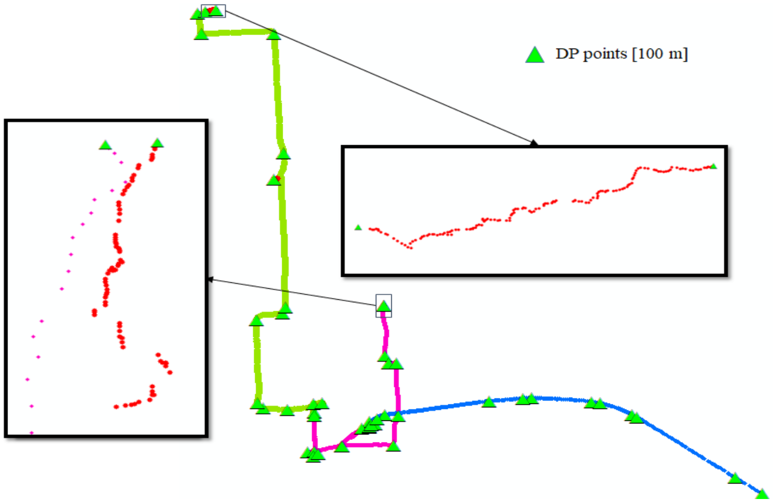


Figure 8. DP trajectory compression with a threshold of 100 m

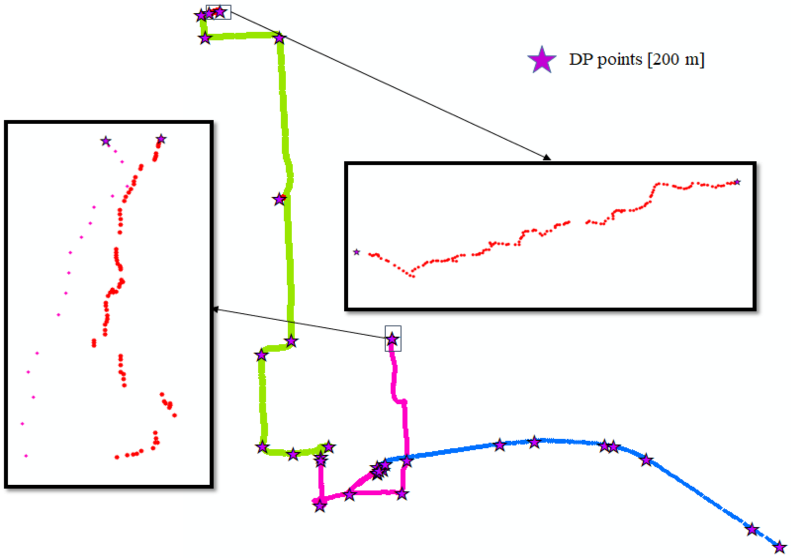


Figure 9. DP trajectory compression with a threshold of 200 m

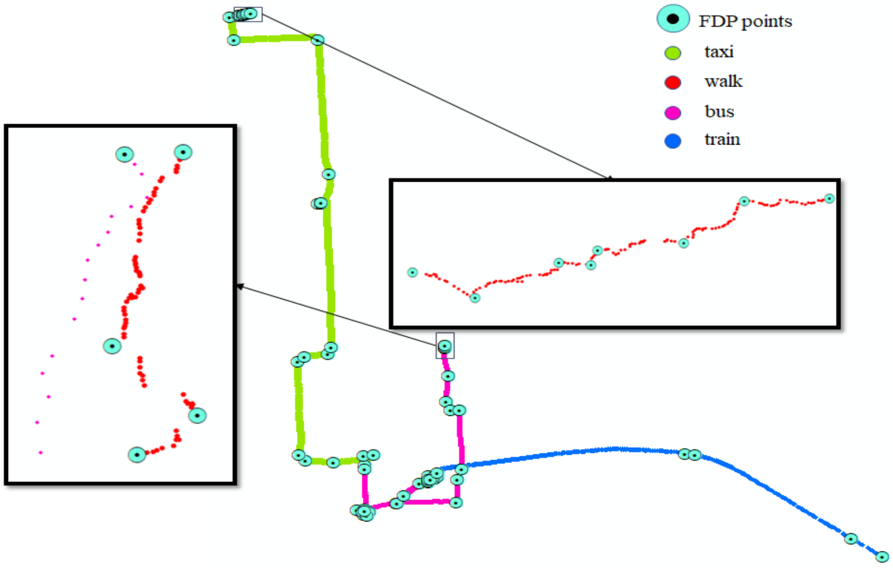


Figure 10. FDP trajectory compression with flexible threshold

As expected, the overall compression rate decreases with the increase of the threshold value in the DP algorithm. The compression rate of the proposed algorithm is 0.0143, which is lower than the compression rate of the DP algorithm with a threshold of 20 meters, and of the DP algorithm with a threshold of 50 meters. However, the compressed trajectory based on the FDP algorithm provides a better approximation of the original trajectory, while preserving the underlying semantics. Also, the compression rate for the 'walk' mode in the proposed algorithm is higher than the compression rate of the other four trajectories, which shows that the 'walk' mode points are better preserved in the compressed trajectory of the FDP algorithm. The Computational times of all algorithms are exhibited in Table 2. Unsurprisingly, it appears that the FDP execution time is greater than that of the DP algorithm with the threshold of 100 meters and 200 meters, while it is equal to the DP execution time with a 50 meters threshold. These comparative results show that the application of the FDP algorithm does not increase computational times while significantly improving the quality and accuracy when considering the spatial and semantic dimensions.

Compression method	Number of compressed points	Compression rate	Transportation mode	Number of Total points for each Transportation mode	Number of compressed points for each Transportation mode	Compression rate points for each Transportation mode	Run time
DP with 20m threshold	102	0.0236	Walk	344	10	0.0291	15 s
			Bus	1903	41	0.0375	
			Taxi	1586	30	0.0189	
			Train	493	21	0.0426	
DP with a 50m threshold	66	0.0153	Walk	344	8	0.0233	13 s
			Bus	1903	23	0.0121	
			Taxi	1586	19	0.0120	
			Train	493	16	0.0324	
DP with 100m threshold	47	0.0109	Walk	344	6	0.0174	12 s
			Bus	1903	16	0.0084	
			Taxi	1586	13	0.0082	
			Train	493	12	0.0243	
DP with 200m threshold	32	0.0074	Walk	344	6	0.0174	11 s
			Bus	1903	8	0.0042	
			Taxi	1586	9	0.0056	
			Train	493	6	0.0122	
FDP with flexible threshold	62	0.0143	Walk	344	18	0.0523	13 s
			Bus	1903	24	0.0126	
			Taxi	1586	14	0.0088	
			Train	493	6	0.0122	

Table 2. Results of compression algorithms

6. CONCLUSION AND FUTURE WORK

This paper introduces a framework for spatial trajectory compression, which is a flexible extension of the well-known DP algorithm applied to line compression. The peculiarity of this FDP algorithm is that, while also considering the spatial dimension, additional semantic constraints are considered to reflect different transportation modes that qualify the considered trajectories. Accordingly, different thresholds are derived to reflect respective displacement speeds associated with these transportation modes when applying point displacements and filtering constraints. The framework has been applied to the Geolife data set that has the advantage of qualifying trajectories according to transportation modes. The results show that the compression algorithm not only has a substantial compression rate of 0.0143, but it also preserves trajectory data points that have a semantic role in relation to different modes of transportation, while finally maintaining a satisfactory accuracy and similarity of the compressed trajectory to the original trajectory. Further work, in addition to the semantic dimension obtained from the mode of transportation, additional semantic information such as temporal data will be used to enhance the trajectory compression process.

REFERENCES

Bellman, R., 1961. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM, 4*(6), 284.

Douglas, D. H., and Peucker, T. K., 1973. 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, 10*(2), 112-122.

Douglas, D. H., Peucker, T. K. J. C. t. i. j. f. g. i., and geovisualization., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *10*(2), 112-122.

Keogh, E., Chu, S., Hart, D., and Pazzani, M., 2001. *An online algorithm for segmenting time series.* Paper presented at the Proceedings 2001 IEEE international conference on data mining.

Kerley, E. R. J. A. j. o. p. a., 1965. The microscopic determination of age in human bone. *23*(2), 149-163.

Makris, A., Kontopoulos, I., Alimisis, P., and Tserpes, K., 2021. A comparison of trajectory compression algorithms over AIS data. *IEEE Access, 9*, 92516-92530.

Meratnia, N., and de By, R. A., 2003. *A new perspective on trajectory compression techniques.* Paper presented at the 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.

Muckell, J., Hwang, J.-H., Patil, V., Lawson, C. T., Ping, F., and Ravi, S., 2011. *SQUISH: an online approach for GPS trajectory compression.* Paper presented at the Proceedings of the 2nd international conference on computing for geospatial research and applications.

Muckell, J., Olsen, P. W., Hwang, J.-H., Lawson, C. T., and Ravi, S. J. G., 2014. Compression of trajectory data: a comprehensive evaluation and new approach. *18*(3), 435-460.

Nasiri, A., Azimi, S., and Abbaspour, R. A., 2018. Data Reduction of Spatio-temporal Trajectories using a Modified Online Compression Algorithm. *Engineering Journal of Geospatial Information Technology, 6*(3), 15-30.

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)* (pp. 275-284). IEEE.

Sun, P., Xia, S., Yuan, G., and Li, D., 2016. An overview of moving object trajectory compression algorithms. *Mathematical Problems in Engineering*.

Tobler, W. J. C. T. I. J. f. G. I., and Geovisualization., 1989. An update to ‘Numerical map generalization’. *26*(1), 7-25.

Trajcevski, G., Cao, H., Scheuermanny, P., Wolfsonz, O., and Vaccaro, D. (2006). *On-line data reduction and the quality of history in moving objects databases.* Paper presented at the Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access.

Wu, X., and Cao, Z. J. E. S.-J. o. C. U. o. G., 2002. Basic conception, function and implementation of temporal GIS. *27*(3), 241-245.

Xiao-li, W., and De, Z., 2010. *Selecting optimal threshold value of Douglas-Peucker algorithm based on curve fit.* Paper presented at the 2010 First International Conference on Networking and Distributed Computing.

Yang, X., Stewart, K., Tang, L., Xie, Z., and Li, Q., 2018. A review of GPS trajectories classification based on transportation mode. *Sensors, 18*(11), 3741.

Zhang, K., Zhao, D., and Liu, W., 2022. Online vehicle trajectory compression algorithm based on motion pattern recognition. *IET Intelligent Transport Systems*.

Zhao, L., and Shi, G. , 2019. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. *Ocean Engineering, 172*, 456-467.

Zheng, Y., Li, Q., Chen, Y., Xie, X., and Ma, W.-Y., 2008. *Understanding mobility based on GPS data.* Paper presented at the Proceedings of the 10th international conference on Ubiquitous computing.

Zheng, Y., Xie, X., and Ma, W.-Y. J. I. D. E. B., 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *33*(2), 32-39.

Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y., 2009. *Mining interesting locations and travel sequences from GPS trajectories.* Paper presented at the Proceedings of the 18th international conference on World wide web.

Zheng, Y., and Zhou, X., 2011. *Computing with spatial trajectories*: Springer Science and Business Media.