

The Abstraction for Trajectories with Different Numbers of Sampling Points

Peng Li, Qing Xu^(✉), Hao Wei, Yuejun Guo, Xiaoxiao Luo, and Mateu Sbert

School of Computer Science and Technology, Tianjin University, Tianjin, China
qingxu@tju.edu.cn

<http://cs.tju.edu.cn/faculty/qingxu/pages/index.html>

Abstract. Trajectory abstraction is an efficient way to handle the large amount of information included in complex trajectory data. Based on the previous work, this paper proposes an improved framework for abstracting trajectories, which consists of three major stages. First, the original trajectories in different lengths are matched into groups according to their similarities, and then a non-local denoising approach, based on the wavelet thresholding technique, is performed on these groups to summarize trajectories. Last, a combined version of the compacted trajectories is obtained as the final trajectory abstraction. To avoid loss of trajectory features introduced by the resampling technique, we provide a novel method to convert trajectories in different lengths into suppositional equal, which serves for the similarity measurement and the wavelet thresholding. Extensive experiments on real and synthetic trajectory datasets demonstrate that the proposed trajectory abstraction achieves very potential results dealing with complex trajectory data.

Keywords: Trajectory abstraction · Outliers detection · Different sampling points · Similarity measurement · Wavelet thresholding

1 Introduction

With rapid development of location-aware sensors in a variety of new applications, massive spatial temporal data, i.e., trajectory data, will soon be accumulated [1]. Trajectory data has a brand range of practical applications in many fields such as intelligent transportation, location-based social networks [2] and so on [3]. The analysis of trajectory data is traditionally based on clustering to exact the patterns and underlying knowledge of these data. Unfortunately, clustering performance degrades when handling trajectory data with complex appearance [4].

To better understand the trajectory data, the framework [4] has been proposed for abstracting trajectories from the perspective of signal processing. In that framework, a resampling technique is firstly exploited to make trajectories have the same number of sampling points for trajectory abstracting framework. Extensive experiments show that the framework for trajectory abstraction gives

very pleasant results for most trajectory data. Unfortunately, the performance degrades in special cases in which there are very tortuous trajectories, as shown in Figs. 4 and 5. The important shape or direction attributes of trajectories have inevitable distortion, leading to information loss of original trajectories. This is mainly because the resampling procedure discards original sampling points, meanwhile new points maybe introduced in the resampled trajectories, thus detailed information of original trajectories are changed to some extent.

This paper is a great leap of the work [4], and the main contribution is that we develop a new method for combating the situation that trajectories have different numbers of sampling points. In concrete, the reinforced framework starts without resampling, and matches original trajectories to form similarity groups. Here, the distance between trajectories is computed in a new way. That is, given any two trajectories in a dataset, for every sampling point of each trajectory, we obtain its corresponding point of the other trajectory by determining a position, where has the same length percentage with the sampling point under consideration. The distance between two trajectories is an average of all distances between sampling points and their corresponding points. We reuse all the virtual points in the following denoising approach. And the final combining is consistent with the previous framework.

The rest of this paper is organized as follows. The next section covers the related work. The improved trajectory abstracting framework is described in Sect. 3. Experimental results are presented and discussed in Sect. 4. The final section concludes the paper.

2 Related Work

Data mining is an interdisciplinary field of computer science [5–7]. It is the computational process of discovering patterns in large datasets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems [5]. Among many data mining algorithms, clustering maybe the most popular way to present patterns of data. Many clustering algorithms have been proposed and developed, such as Density Based Spatial Clustering of Applications with Noise (DBSCAN) [8], Ordering Points to Identify the Clustering Structure (OPTICS) [9], k -means [10], Statistical Information Grid (STING) [11] and so on [12]. Due to simplifying and easily understanding, k -means is widely used, however, it is hard to obtain the number of clusters k adaptively, which influences the clustering performance directly. DBSCAN is a density-based clustering algorithm, which is less effective when handling high-dimensional data. Moreover, border points that are reachable from more than one cluster can be classified into either cluster, depending on the order in which the data is processed.

These clustering algorithms are often used to mine information and find outliers from the data, but for complex trajectory data, the performance may not be so satisfactory. To process and analyze trajectories effectively, a trajectory abstraction framework is proposed in [4], which summarizes trajectories from

perspective of signals. The experiments show that it is suitable to deal with most of the general trajectory data, meanwhile some distortions maybe introduced. Therefore, this paper provides a novel framework based on the previous framework in order to handle more intractable trajectory abstracting application and distinguish outliers in a more appropriate way.

3 The Framework for Trajectory Abstraction

3.1 Distance Measurement

In practical applications, different trajectories always have different numbers of sampling points, which brings difficulty to measure the distance between trajectory data. In the work [4], all the trajectories are firstly resampled to have the same numbers of sampling points, making the distance measurement much easier. While in fact, the sampling technique may somehow destroy original shapes of trajectories, especially when dealing with very tortuous trajectory data. For instance, the sampling procedure may smooth out several turning points of a complex trajectory. As a result, the accuracy of distance measurement can be largely degraded. In order to improve the performance, we propose a new distance measurement to overcome the distortion generated by sampling.

In concrete, given the i -th trajectory in a dataset with m sampling points defined as

$$S_i = \{S_{i,1}, S_{i,2}, S_{i,3}, \dots, S_{i,m}\}, \quad (1)$$

where $S_{i,j}$ is the j -th sampling point of trajectory S_i . The key to compute the distance between S_i and another trajectory S_j is to find out the “corresponding” sampling points in S_j for each point in S_i . We define the points with the equal *percentage* values have the “corresponding” relation. The *percentage* of the sampling point $S_{i,j}$ is calculated by the ratio of the trajectory length between $S_{i,1}$ and $S_{i,j}$ to the total length. Here, by length we mean the sum of Euclidean distances between adjacent sampling points. For example, as shown in Fig. 1. The length between $S_{i,1}$ and $S_{i,4}$ is actually the length of the red line, and similarly, the total length of the trajectory is the sum length of red and blue segments. Notice that, the corresponding point in S_j may a the new one, the position of which is computed by *percentage*. With all the corresponding points in S_j being located for sampling points in S_i , we can obtain the distance between $S_{i,1}$ and $S_{i,4}$ by the average of all distances between the pairwise points with corresponding relations. Figure 2 illustrates such an example of calculating the distance between trajectories S_i and S_j . The corresponding sampling point of $S_{i,k}$ is noted as $S'_{i,k}$, and for $S_{j,p}$ it is $S'_{j,p}$. Obviously, the final distance is the average length of all dashed lines.

Hausdorff and Euclidean distances may be the most widely used distance measures. While in fact, Euclidean distance is simple but requires the trajectories under consideration to have equal numbers of sampling points. Hausdorff distance does not have such requirement, but it is difficult to distinguish the directions of the trajectories, and it fails to deal with complex trajectories with

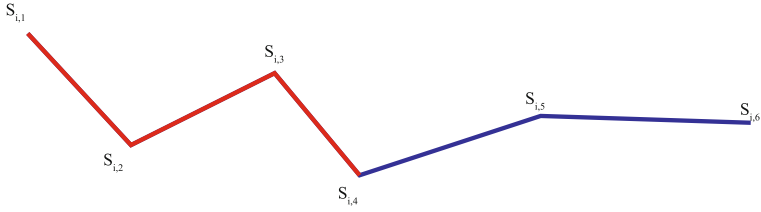


Fig. 1. An example of a trajectory with 6 sampling points.

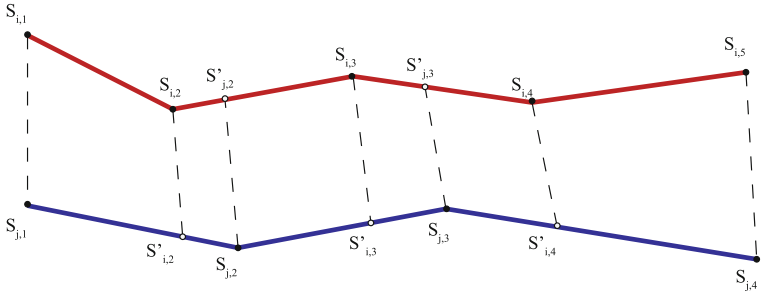


Fig. 2. Computation the trajectory distance between S_i and S_j

circular paths. By contrast, our proposed distance measurement possesses the simplicity of Euclidean distance and the general applicability of Hausdorff distance, and simultaneously overcomes the shortcomings of them.

3.2 Trajectory Abstraction

The improved trajectory abstraction framework is based on the work [4]. Different from the prior work, resampling is ignored in this paper, remaining the non-local denoising phase, including matching, thresholding and combining. Due to different numbers of sampling points, we also make a variant in thresholding. The reinforced framework can also iteratively output trajectory abstractions with multi-granularities and outliers.

Matching. In this step, each trajectory is regarded as a reference, for the purpose of matching its similar trajectories to establish groups. During the matching procedure, two trajectories S_i and S_j is matched into the same group when their distance is less than a threshold τ , which can be selected adaptively [4]. We adopt the new distance metric to measure the similarity between trajectories. Thus resampling can be avoided and shape feature of trajectories can be reserved. Note that a trajectory can be matched into more than one groups. That means a trajectory may have several duplicates in different groups, and they are independent of each other.

Thresholding. After matching, the wavelet thresholding technique is operated on every group. Notice that the similarity groups consist of similar trajectories but with different numbers of actual sampling points. Assume that the reference trajectory is S_r , and its similarity group TG_r with m trajectories is defined as

$$TG_r = \{S_j | Diff(S_r, S_j) < \tau\} \tag{2}$$

where $Diff(\cdot)$ is the new distance metric mentioned in Sect. 3.1. We perform the wavelet thresholding on each sampling points of every trajectories in the group. That is, given a trajectory S_j from the group, we find all the virtual points in the whole group corresponding to each actual point of this trajectory. The collection of all virtual points together with the sampling point $S_{j,k}$ is now denoted as

$$s_{j,k} = [S'_{j,1}, S'_{j,2}, \dots, S_{j,k}, \dots, S'_{j,m}]^T. \tag{3}$$

Thus, we transform trajectories into signals, which are then filtered by the wavelet thresholding technique.

Figure 3 presents a simple example. Suppose we have already formed a group of three trajectories. The red trajectory consists of two sampling points. The green is of three sampling points and the blue is four.

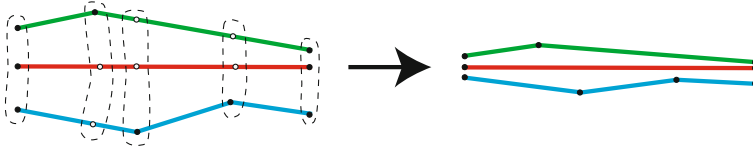


Fig. 3. Construction of points for filtering in a group (Color figure online)

We do the thresholding for every sampling point, i.e., each solid point in Fig. 3 will be filtered. In the figure, each dotted circle contains three points, which are either all solid points or a solid point with two hollow points. And the three points in a dotted circle have the same percentage value in their respective trajectories, and they will be filtered by wavelet thresholding. After filtering, only solid points will update their respective trajectories in the group, and hollow points will be discarded due to their fictionality. In case that there is a real sampling point corresponding to the sampling point to be filtered, we always use the original version of the real point, instead of the one being filtered.

Combining. With all groups filtered by thresholding, we obtain the filtered and condensed trajectories in each group. Notice that a trajectory can have different duplicates after filtering, since it is reasonable to exploit a same trajectory in several groups. Therefore, we perform the combining, by averaging its duplicates in all groups, to get the final form for each trajectory.

4 Experimental Results

In this section, to evaluate the performance of the improved framework, we have studied 7 trajectory datasets, including real and synthetic data, as listed in Table 1. In addition, comparison with the previous method [4] has been made. For reasons of space, only two real datasets, *video*¹ and *GPS*² [13–15], are illustrated in the following.

Table 1. Evaluation on outlier detection [4]

Database	Recall			Precision		
	Improved	Previous	DBSCAN	Improved	Previous	DBSCAN
Video ^d	1.0	0.67	0.67	1.0	1.0	1.0
GPS ^b [13–15]	1.0	0.67	0.67	1.0	1.0	1.0
Highway [16]	0.93	0.89	0.94	0.89	0.89	0.84
Simulation [17]	1.0	1.0	1.0	1.0	1.0	0.95
Edinburgh ^a	0.95	0.95	0.76	0.94	0.81	0.86
Aircraft ^c	0.95	0.88	0.89	1.0	0.95	0.81
CROSS [18]	0.93	0.98	0.93	0.91	0.92	0.81

^a <http://homepages.inf.ed.ac.uk/rbf/forumtracking/>

^b <http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/default.aspx>

^c <https://c3.nasa.gov/dashlink/resources/132/,2011>

^d <http://www-users.cs.umn.edu/~aleks/inclof/>

As shown in Fig. 4(a), the *video* is a really complex dataset, which includes 189 trajectories with the number of sampling points ranging from 174 to 645. The abstraction results by the improved and the previous methods are presented in Figs. 4(b) and (c), respectively. Note that, the previous method requires the resample process, which firstly smooths the trajectories and makes them in equal numbers of sampling points. Obviously, it is very difficult to summarize the trajectories due to their intricate shapes. By contrast, the performance of the improved method is more satisfying. For instance, in Fig. 4(a), 24 blue and 1 red trajectories seem to be similar with respect to the small waves of the lower half parts, while obviously the red one has relatively gradual shape changes of the upper half parts. Our improved method successfully smooths the blue trajectories, without largely losing the shape information from the overall perspective, and clearly detects the red one as an outlier. Unfortunately, the previous method mixes the blue trajectories with the outlier and gives the final abstraction result.

¹ <http://www-users.cs.umn.edu/~aleks/inclof/>.

² <http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/default.aspx>.

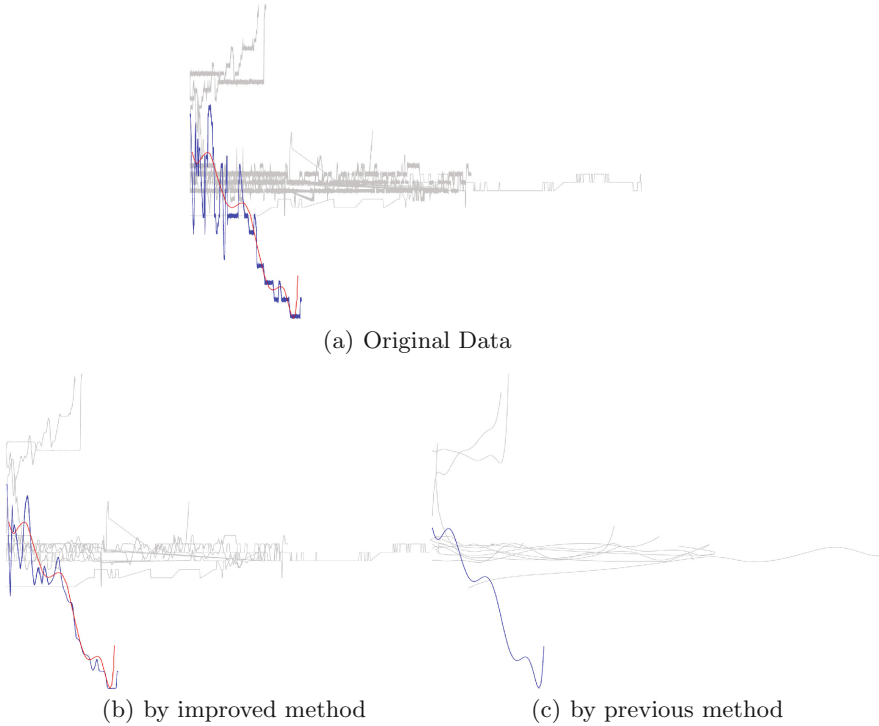


Fig. 4. Comparison of abstraction results on *Video* (Color figure online)

Figure 5(a) is the original 77 trajectories of *GPS*, where the red one is an outlier. These trajectories are of different numbers of sampling points, up to 370. The abstracted trajectories of original data are in Fig. 5(b). The resampled abstracted trajectories are in Fig. 5(c). In Fig. 5(b), three normal trajectories in blue overlap with each other, the same situation happens in Fig. 5(c). The difference is that the red anomaly trajectory is identified by our improved method due to the appropriate distance metric, as shown in Fig. 5(b), while the previous method treated the blue normal trajectories and the outlier as similar items due to the smoothing by resampling, as shown in Fig. 5(c).

Additionally, we make use of recall and precision metrics [16] to quantitatively measure the effect of our improvements in terms of anomaly detection. Table 1 shows the comparison results of our improved method, the previous method and DBSCAN. In all, our improved method and the previous method outperform the typical DBSCAN, and the improved version indeed enhances the ability of handling complex trajectory data.

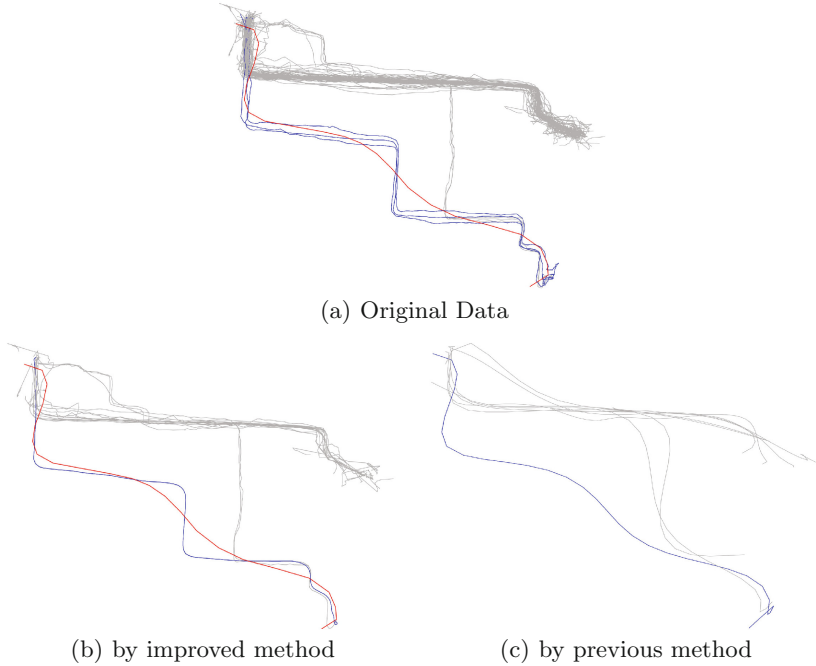


Fig. 5. Comparison of abstraction results on *GPS* (Color figure online)

5 Conclusion

The framework of doing trajectory abstracting [4] is able to process trajectory data more effectively than the common clustering algorithms. In order to preserve the advantages of the trajectory abstraction framework and meanwhile, to avoid the problem introduced by resampling, this paper has enhanced the previous trajectory abstracting framework to better deal with complex trajectories with massive details. We have made progresses in trajectory thresholding, that is, the wavelet thresholding can be handled on trajectories of various numbers of sampling points. And we have designed a new distance metric for tortuous and littery trajectories. The experimental results show that the improved framework has a stronger ability to abstract trajectory data with varied lengths and distinguish outliers.

Several improvements for the framework of trajectory abstraction will be tried in our future research. The time consumption of the trajectory abstraction should be reduced in order to handle a greater amount of trajectories efficiently. Furthermore, the abstraction method should be reinforced to handle trajectories of 3-Dimension or higher dimension data.

Acknowledgments. This work has been funded by Natural Science Foundation of China (61471261, 61179067, U1333110), and by grants TIN2013-47276-C6-1-R from Spanish Government and 2014-SGR-1232 from Catalan Government (Spain).

References

1. Chakka, V.P., Everspaugh, A.C., Patel, J.M.: Indexing large trajectory data sets with seti. *Ann Arbor* **1001**(48109–2122), 12 (2003)
2. Zheng, Y.: Tutorial on location-based social networks. In: *Proceedings of the 21st International Conference on World Wide Web, WWW*, vol. 12 (2012)
3. Morris, B.T., Trivedi, M.M.: Understanding vehicular traffic behavior from video: a survey of unsupervised approaches. *J. Electron. Imaging* **22**(4), 041113 (2013)
4. Luo, X., Xu, Q., Guo, Y., Wei, H., Lv, Y.: Trajectory abstracting with group-based signal denoising. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) *ICONIP 2015. LNCS*, vol. 9491, pp. 452–461. Springer, Cham (2015). doi:[10.1007/978-3-319-26555-1_51](https://doi.org/10.1007/978-3-319-26555-1_51)
5. Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., Wang, W.: Data mining curriculum: a proposal (version 1.0). Intensive Working Group of ACM SIGKDD Curriculum Committee (2006)
6. Christopher, C.: *Encyclopaedia britannica: definition of data mining*. Technical report (2010). Accessed 09 Dec 2010
7. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. *Math. Intell.* **27**(2), 83–85 (2005)
8. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD 1996*, pp. 226–231 (1996)
9. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: ordering points to identify the clustering structure. In: *ACM SIGMOD Record*, vol. 28, pp. 49–60. ACM (1999)
10. Lloyd, S.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
11. Wang, W., Yang, J., Muntz, R., et al.: Sting: a statistical information grid approach to spatial data mining. In: *VLDB 1997*, pp. 186–195 (1997)
12. Rui, X., Wunsch, D., et al.: Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **16**(3), 645–678 (2005)
13. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y.: Understanding mobility based on GPS data. In: *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 312–321. ACM (2008)
14. Zheng, Y., Xie, X., Ma, W.-Y.: Geolife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* **33**(2), 32–39 (2010)
15. Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y.: Mining interesting locations and travel sequences from GPS trajectories. In: *Proceedings of the 18th International Conference on World Wide Web*, pp. 791–800. ACM (2009)
16. Anjum, N., Cavallaro, A.: Multifeature object trajectory clustering for video analysis. *IEEE Trans. Circ. Syst. Video Technol.* **18**(11), 1555–1564 (2008)
17. Piciarelli, C., Micheloni, C., Foresti, G.L.: Trajectory-based anomalous event detection. *IEEE Trans. Circ. Syst. Video Technol.* **18**(11), 1544–1554 (2008)
18. Morris, B.T., Trivedi, M.M.: Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2287–2301 (2011)