An Effectiveness Study on Trajectory Similarity Measures

Haozhou Wang[1] **Han Su**[1] **Kai Zheng**[1] **Shazia Sadiq**[1] **Xiaofang Zhou**[1]

School of Information Technology and Electrical Engineering,
The University of Queensland,
St. Lucia, Brisbane, QLD 4072, Australia
Email: [1]{h.wang16, h.su1, kevinz, shazia, zxf}@uq.edu.au

Abstract

The last decade has witnessed the prevalence of sensor and GPS technologies that produce a sheer volume of trajectory data representing the motion history of moving objects. Measuring similarity between trajectories is undoubtedly one of the most important tasks in trajectory data management since it serves as the foundation of many advanced analyses such as similarity search, clustering, and classification. In this light, tremendous efforts have been spent on this topic, which results in a large number of trajectory similarity measures. Generally, each individual work introducing a new distance measure has made specific claims on the superiority of their proposal. However, for most works, the experimental study was focused on demonstrating the efficiency of the search algorithms, leaving the effectiveness aspect unverified empirically. In this paper, we conduct a comparative experimental study on the effectiveness of six widely used trajectory similarity measures based on a real taxi trajectory dataset. By applying a variety of transformations we designed for each original trajectory, our experimental observations demonstrate the advantages and drawbacks of these similarity measures in different circumstances.

1 Introduction

Driven by major advances in sensor technology, GPS-enabled mobile devices and wireless communication, large amounts of data describing the motion history of moving objects, known as *trajectories*, are currently generated and managed in many of application domains such as environmental information systems, meteorology, wireless technology, video tracking, or video motion capture (Zheng et al. n.d., Shang et al. 2012, Zheng et al. 2011, Xie et al. 2009, Chen et al. 2010, Zheng et al. 2010, Chen et al. 2011). Typical examples include collecting the GPS location histories of taxicabs for safety and management purpose, tracking animals for their migration patterns, gathering human motion data by tracking body joints, or tracing the evolution of migrating particles in biological sciences.

The trajectory of a moving object is typically modelled as a time-stamped sequence of consecutive locations in a multidimensional (generally two or three dimensional) space. Such type of data has offered unprecedented information to help understand the behaviour of moving objects, and resulted in growing interest of data analysis in such data. An important problem in such analysis is designing techniques for identifying trajectories that are similar. Such techniques can be used by many data analysis tasks including trajectory clustering, classification, and k-nearest neighbor search, which have a broad range of real applications. For instance, in many sports such as football and tennis, it is very useful for sports researchers to figure out the movement patterns of top players by finding similar trajectories of objects (players, balls) motions. By analyzing similar trajectories of animals, it is possible to determine migration patterns for them. In a city traffic monitoring system, it is helpful to locate popular routes by comparing similarity between vehicles trajectories.

A fundamental ingredient of such trajectory analysis tasks is the distance/similarity measure that can effectively determine the similarity of trajectories. But unlike other simple data types such as ordinal variables or geometric points where the distance definition is straightforward, the distance between trajectories needs to be carefully defined in order to reflect the true underlying similarity. This is due to the fact that trajectories are essentially high dimensional data attached with both spatial and temporal attributes, which needs to be considered for similarity measures. Therefore, over ten of distance/similarity measures have been proposed in the literature, e.g., Euclidean distance (ED) (Jonkery et al. 1980), Dynamic Time Warping (DTW) (Soong & Rosenberg 1988), distance based on Longest Common Subsequence (LCSS) (Kearney & Hansen 1990),Edit Distance with Real Penalty (ERP) (Chen & Ng 2004), Edit Distance on Real sequence (EDR) (Chen et al. 2005). Many of these works and some of their extensions have been widely cited in the literature and applied to facilitate query processing and data mining in trajectory data.

Given the multitude of competitive techniques, a good understanding the effectiveness of various similarity measures is important. Very often a newly introduced distance measure has claimed a particular advantage over some others by using an exemplified explanation. Also most of those works focused on evaluating the efficiency of their pruning and searching algorithms, while leaving the effectiveness study, i.e., how their proposed distance measures truly reflects the similarity between trajectories under different circumstances, inadequate or even completely omitted. In this light, we argue that there is a strong need for an empirical study on the effectiveness of trajectory similarity measures. More specifically, in this paper we have implemented 6 widely used trajectory similarity measures (shown in following list), and study their effectiveness in different circumstances using a common real world taxicab trajectory dataset.

- Euclidean Distance Measure
  - Euclidean Distance
- Dynamic Time Warping based Measures
  - DTW
  - PDTW
- Edit Distance based Measures
  - EDR
  - ERP
- Longest Common Subsequence based Measures
  - LCSS

In summary, we make the following contributions in this work.

- We observe the absence and importance of an objective effectiveness study on widely used trajectory similarity measures.
- In order to overcome the lack of benchmark dataset for effectiveness test, we devise a set of reasonable transformation functions for the original trajectory data, the variance of which is controlled by parameters.
- We evaluate the similarity between original and transformed trajectories, and study how the similarity is reflected in six different distance measures.

The rest of the paper is organised as follows. Section 2 presents the preliminary concepts and briefly review the trajectory similarity measures we are about to examine in this work. In Section 3, we discuss the trajectory dataset used in this study, the types of transformations applied to the trajectories, and the experimental observations regarding the effectiveness of the compared similarity measures. We finally give our conclusion in Section 4.

2 Similarity Measures for Trajectories

Theoretically a trajectory represents the continuous motion history of a moving object. However, due to the limitation of location positioning devices (e.g, sensors, GPS devices), a trajectory in real world is a sequence of positions observed at discrete time instances. This is also the reason that most of the existing works on trajectories assume the time is discrete rather than continuous. Without loss of generality, we make the same assumption in this paper. Formally, a trajectory can be defined as the follows:

DEFINITION 1 *A trajectory Tr is a finite sequence of geo-locations with timestamps, i.e., $Tr = (p_1, t_1), (p_2, t_2), ..., (p_n, t_n)$ with $t_i < t_{i+1}$ for $i = 1, 2, ..., n-1$. p_i is a sampling point that is observed at time t_i.*

Generally the location of a sampling point of a trajectory is represented by a coordinate in multidimensional space. But for the sake of simplicity, in this paper we focus on trajectories in two dimensional space since it is applicable to a wide variety of application scenarios. Thus, each sampling point is represented by a pair (x, y), denoting longitude and latitude respectively.

In the following subsections, we briefly review the trajectory similarity measures studied in this work. Notice that this is not meant to be a complete survey for the respective field and is only intended to provide the readers with a necessary background for following our experimental evaluations.

2.1 Euclidean Distance Measure

Euclidean distance, also known as L_2-norm, is distance measure in literature for a variety of applications. Given two trajectories T_1, T_2, the Euclidean distance $d(T_1, T_2)$ can be calculate as, $d(T_1, T_2) = \frac{\sum_{i=1}^{n} d(p_{1,i}, p_{2,i})}{n}$, where $d(p_{1,i}, p_{2,i})$ is the distance on spatial space. Euclidean distance is easy to implement and indexable with many access methods, and it is parameter-free. In addition, the complexity of Euclidean distance measure is linear, which means it can handle a large size of trajectory data set. Euclidean distance is proposed as a distance measure between time series and is one of most commonly used similarity function since 1960s (Priestley 1980, Pfeifer & Deutsch 1980, Faloutsos et al. 1994, Keogh & Pazzani 2000). Later, Euclidean distance is also extended to measure the distance between trajectories (Clarke 1976, Richalet et al. 1978, Jonkery et al. 1980, Sanderson & Wong 1980, Takens 1980), since trajectories and time series have the similar representations.

2.2 Dynamic Time Warping based Measures

Dynamic Time Warping(DTW) is a well-known algorithm for finding similar trajectory patterns between two trajectories. The definition of DTW uses a recursive manner to search all possible point combinations between two trajectories for the one with minimal distance, which can be converted to dynamic programming very easily. DTW allows to find a similar pattern between two given trajectories, which can be of different lengths, with or without time information. Moreover, the original DTW similarity measure is also parameter-free. For example, two trajectories are separately generated by a slowly moving object and a fast moving object, DTW can still report their similarity pattern. DTW was first introduced to compute the distance of time series (Myers et al. 1980). In 1980s,(Kruskal 1983, Soong & Rosenberg 1988, Picton et al. 1988, Ostendorf & Roukos 1989) introduced DTW to measure trajectory distance. For a huge data set, DTW is time-consuming and I/O-consuming. To speed up DTW and reduce I/O cost, several pruning methods have been introduced such as FastMap method and lower bound method (Sakurai et al. 2005, Yi et al. 1998).

Piecewise Dynamic Time Warping (PDTW) (Keogh & Pazzani 2000) is another dynamic time warping based similarity trajectory function, which is improved from DTW. PDTW speeds up DTW by a large constant c, where c is data dependent. PDTW uses two steps to calculate similarity trajectory pattern. The first step called Piecewise Aggregate Approximation (PAA), which cuts a given trajectory into c pieces, where $[p_{c*(i-1)+1}, p_{c*(i-1)+2}, \cdots, p_c * i]$ is i-th piece. For piece i, PAA computes $\bar{p}_i$ as a representitive point and transform trajectory T into piecewise approximation $\bar{T} = [\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_N]$. Then, in the second step, PDTW process DTW distance to find similar trajectory patterns between transformed trajectories $\bar{T}_1$ and $\bar{T}_2$.

2.3 Edit Distance based Measures

Edit distance with Real Penalty (ERP) (Chen & Ng 2004) is an edit distance (ED) based trajectory similarity measure. ERP uses L_1-norm as the distance measure. Introducing L_1-norm makes ERP a metric measure, which is a prominent advantage over DTW and LCSS, as metric measures allow for efficient pruning. In addition, ERP distance is defined on normalized trajectory data for amplitude scaling and global spatial shifting. ERP normalize a trajectory T by shifting by its mean (μ) and scaling by its standard deviation(σ): $Norm(T) = [\frac{p_1-\mu}{\sigma}, \frac{p_2-\mu}{\sigma}, \ldots, \frac{p_n-\mu}{\sigma}]$.

Edit Distance on Real sequence (EDR) (Chen et al. 2005) is another edit distance (ED) based trajectory similarity measure. EDR also uses a threshold ε to detect sample points matching, which is similar with LCSS. Like ERP, EDR also uses normalized trajectory data, in order to be invariant to scaling and shifting. Different with ERP, for each sample point p_i in T, the position values of x, y are normalized by using the corresponding mean $(\mu_x), (\mu_y)$ and standard deviation $(\sigma_x), (\sigma_y)$, respectively: $Norm(T) = [(\frac{p_{1,x}-\mu_x}{\sigma_x}, \frac{p_{1,y}-\mu_y}{\sigma_y}), \ldots, (\frac{p_{n,x}-\mu_x}{\sigma_x}, \frac{p_{n,y}-\mu_y}{\sigma_y})]$. The matching defined by EDR is $match(p_i, p_j)$ for a pair of trajectories' sample points, where $p_i \in T$ and $p_j \in T', T \neq T'$. $match(p_i, p_j)$ is true if and only if $|p_{i,x} - p_{j,x}| \leq \varepsilon$ and $|p_{i,y} - p_{j,y}| \leq \varepsilon$, where ε is the matching threshold. If $match(p_i, p_j)$ is true, the $subcost$ (i.e. edit distance) between p_i and p_j is 0, otherwise the $subcost = 1$.

2.4 Longest Common Subsequence based Measures

Some similarity measures work well based on the assumption how the trajectory data are clean. However, the trajectory data generated by GPS devices are not clean enough due to device's accuracy limitation, bad GPS signals, and other factors. Therefore, a similarity measure which is more robust for processing low quality trajectory data attracts great research interest. Longest common subsequence (LCSS) is one of most popular measurements, which is used for string similarity, (Ichiye & Karplus 1991, Robinson 1990) apply LCSS as trajectory similarity measure. For detecting sample points matching like string's characters matching, a threshold ε is used, if two points' distance less than ε, they are considered to match. The basic idea of LCSS is that it allows some sample points unmatched to match some sequences in trajectories. LCSS is good for processing with low quality trajectory data (i.e. noisy trajectory data), which can figure out similarity trajectories in high accuracy. However, it may lead to some inaccuracy, since it does not consider various unmatched sequences in trajectories. We illustrate this case in our experiment.

3 Effectiveness Study

In this section we present the trajectory dataset used in this study, the types of transformations applied to the trajectories, and the experimental observations regarding the effectiveness of the compared similarity measures.

3.1 Dataset

We employed the dataset of Beijing taxi trajectories (Zheng et al. 2009) for our experimental study. This is a real-world trajectory dataset generated by 30,000 taxicabs in Beijing in a period of 3 months. Figure 1 shows an example of the taxi trajectories superimposed on Google Map, in which the sample points of trajectory are represented by square marks on the map. The sampling rate of this data set is approximately 30 seconds, which means the time duration between consecutive sampling points is about 30 seconds. Since in this study we mainly focus on effectiveness rather than scalability, we randomly selected 1000 trajectories from the dataset, where each contains at least 100 sampling points.

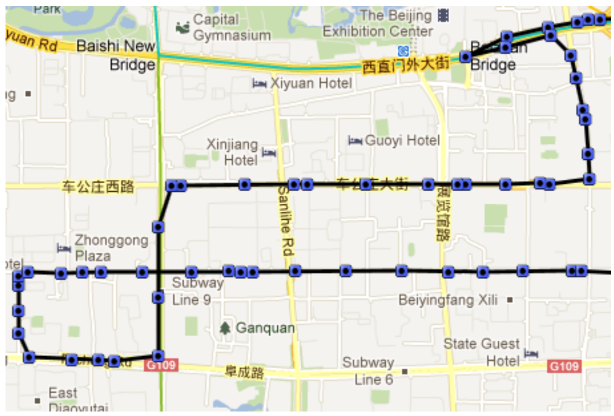

Figure 1: Trajectory Example

3.2 Trajectory Transformations

Evaluating effectiveness of different similarity measures objectively is a challenging task due to the lack of a widely recognized benchmark dataset, where the ground-truth distance between any pair of trajectories is known in advance. Therefore, while most previous works put emphasis on the scalability test for the similarity measures, none of them have conducted experiments on the effectiveness. In this work, we tackle this problem from a novel aspect by having the following two observations. First, an identical motion history can be represented by different trajectories due to the variance in sampling time, sampling rate or possible noises. Second, in spite of different representations, they should still have high similarity based on any good similarity measure since they all actually refer to the same motion record.

Based upon this, our evaluation procedure works as follows. We firstly pick up a trajectory as the *original trajectory*. Then we perform several types of transformations on the original trajectory in a controlled way (by using parameters), resulting in a set of *transformed trajectories*. For each transformation, we will evaluate the distance between the original and transformed trajectories and tune the parameter to see how their distance is affected. The rational behind this is that, with a reasonable similarity measure, the trajectory with a lower degree of transformation should have higher similarity with the original trajectory, and vice versa.

We devise three types of transformation functions, namely re-sampling a trajectory, shifting trajectory points, and adding noise. These transformations are controlled by two parameters, *rate* and *distance*. The parameter *rate* is used to specify the percentage of the trajectory points that will be transformed; for instance, $rate = 0.1$ means that 10% of the trajectory points are to be transformed by the transformation function, $distance = 0.0001$ means that the trajectory points are to be shifted around 11 meters by transformation function. The parameter *distance* is a threshold on how far a trajectory point might be shifted with respect to the original point. Table 1 summarizes all the transformation functions and the parameters.

Re-sampling trajectory. There are two ways to re-sample a trajectory, i.e., increasing sampling rate and decreasing sampling rate. To increasing sampling rate, we will randomly add *rate* extra points to the original trajectory. Analogously, to decrease sampling rate, we randomly remove *rate* points from the original trajectory. Figure 2 and Figure 3 exemplify those two opposite transformations.

Figure 2: Increase Sampling Rate Transformation Function

Point shift. Unlike the re-sampling transformation, point shift does not change the number of trajectory points. Instead, it changes the locations of them. To do so, we randomly select *rate* of the trajectory points and shift them by *distance*. There are two ways to shift the points, i.e., random shift and synchronized shift. Random shift will change the position of each selected point arbitrarily without considering the other shifted points, while synchronized shift will translate all the selected points in the same way (same offset and direction). Additionally, point

Figure 3: Decrease Sampling Rate Transformation Function

shift transformation would not change the shape of original trajectories. Figure 4 and Figure 5 illustrate these two shift transformations.

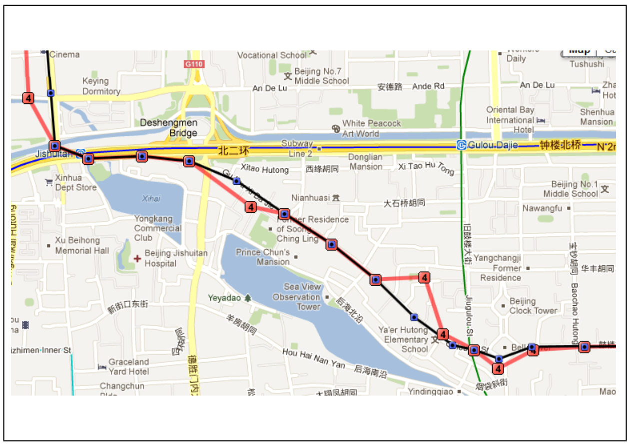

Figure 4: Random Shift Transformation Function

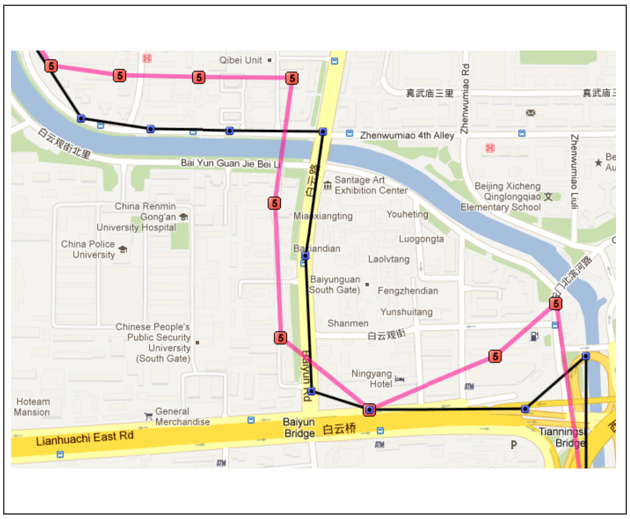

Figure 5: Synchronized Shift Transformation Function

Adding noise. The last transformation function is to add *rate* noises/outliers to the original trajec-

Table 1: Types of Trajectory Transformations

Transformation Type	Operation	Adjustable Parameters
Re-sampling	Increase sampling rate (add points)	*rate*
	Decrease sampling rate (remove points)	*rate*
Point shift	Random shift	*rate*, *distance*
	Synchronized shift	*rate*, *distance*
Noise	Add noise	*rate*, *distance*

tory. The gap between the noisy points and the original trajectory is controlled by the parameter *distance*. We use an example to demonstrate this transformation in Figure 6.

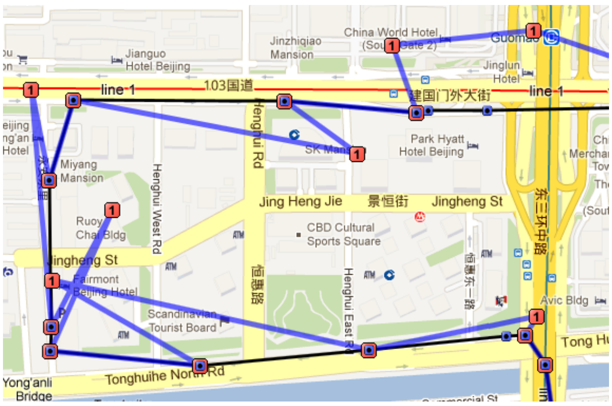

Figure 6: Add Noise Transformation Function

3.3 Experimental Observations

In this part, we apply the set of transformations to the original trajectories and compute the distance/similarity between the original and transformed trajectories based on each similarity measure. Specifically, for each similarity measure, we conduct two sets of experiments. First, we fix the parameter *distance* as constant ($distance = 0.0015$) , and then vary the parameter *rate* from 0.1 to 0.6 with the step of 0.1. However, EDR and LCSS measures use another threshold ε to determine the matched pairs of points. The relationship between *distance* and ε will heavily affect the results. Therefore, we conduct two sets of experiments for LCSS and EDR, i.e., with $\varepsilon = 0.002$ being greater than *distance*, and $\varepsilon = 0.0004$ being less than *distance*. Furthermore, adding $10\% - 60\%$ noise points into trajectory are too much and could change the shape of original trajectory, hence we reduce the transformation rate of adding noise function from 0.1 to 0.06 with the step of 0.01, which is one-tenth of previous parameter *rate*.

Second, we fix the parameter *rate* as constant ($rate = 0.3$)[1], and change the value of parameter *distance* from 0.0005 to 0.004 (Euclidean distance in spatial space) with the step of 0.0005. We only change the transformation distance for random shift, synchronised shift and adding noise as only these transformations are affected by this parameter.

[1] Transformation rate of add noise function is set to 0.03

For all the similarity measures except LCSS and EDR, we report the distance between the original and transformed trajectories. Hence a greater value indicates a lower similarity.

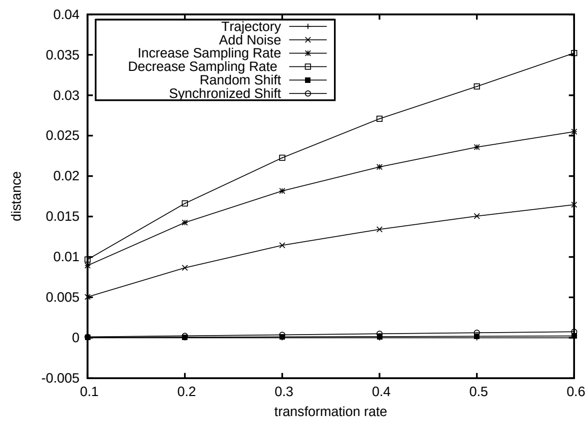

Figure 7: Result of Euclidean Distance with different transformation rate

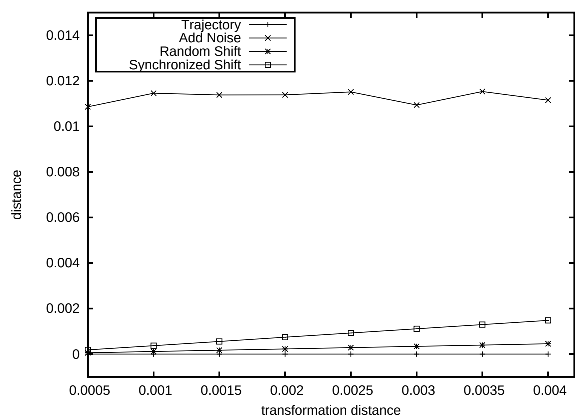

Figure 8: Result of Euclidean distance with different transformation distance

Euclidean Distance Measure. The result of Euclidean distance with varying *rate* is shown in Figure 7[1] . We can see that, the distance between the original and transformed trajectories with resampling and noise increases quickly as the transformation rate rises. This implies that the Euclidean distance is sensitive to sampling rate or noise. On the other hand, shifting sampling points within certain range has little influence on the distance.

Next, we illustrate the result of Euclidean distance with different transformation distance in Figure 8. As expected, the distance between the original and transformed trajectories with point shift gradually increases as the transformation distance increases. But adding noises will make the transformed trajectory completely dissimilar with the original one, which again indicates that Euclidean distance is sensitive to outliers.

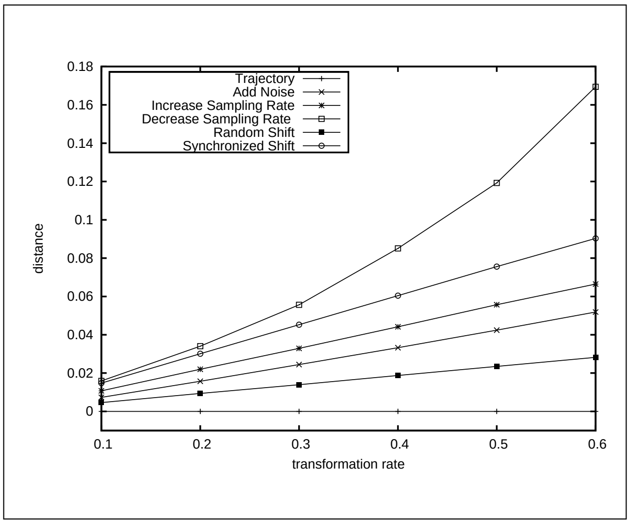

Figure 9: Result of DTW with different transformation rate

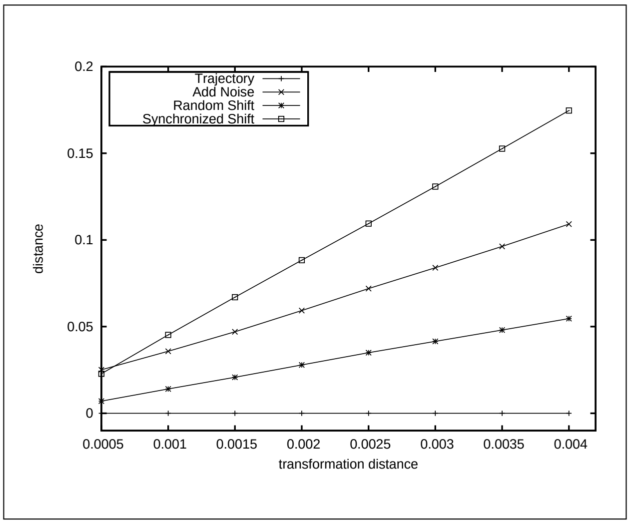

Figure 10: Result of DTW with different transformation distance

Dynamic Time Warping based Measures. The performance of DTW with changing *rate* is shown in Figure 9. It can be observed that DTW achieves a relatively good performance with low transformation rate (i.e. $rate < 20\%$). DTW is more robust to the random shift transformation. Besides it is more sensitive to decreasing sampling rate than increasing sampling rate. Also DTW may not be a good choice when the trajectory data is contaminated by noises.

We then evaluate the DTW distance with different transformation distances. As shown in Figure 10, DTW is more sensitive to transformation distance than transformation rate as all distances increase quickly when the transformation distance gets enlarged. DTW may not handle dramatic sampling points shift well especially for the synchronized shift.

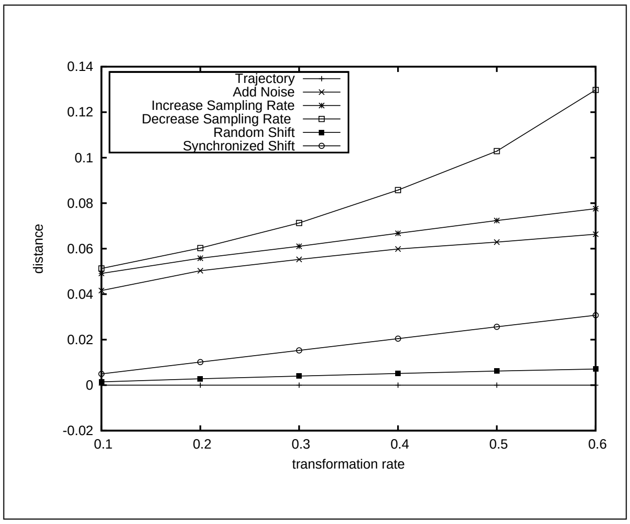

Figure 11: Result of PDTW with different transformation rate

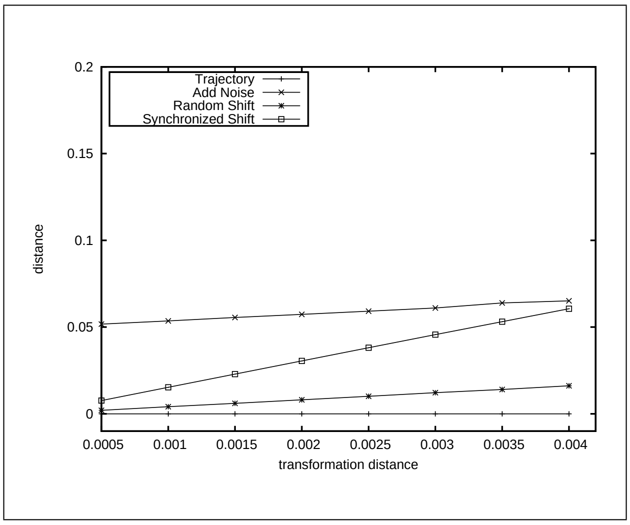

Figure 12: Result of PDTW with different transformation distance

Figure 11 shows the results of PDTW with different transformation rate, which are similar with DTW, since PDTW is a variant improved from DTW. However, after applying the PAA method, the effectiveness of PDTW is better than DTW. With the same scale and unit as in the experiments of DTW, we can see the distance reported by PDTW are less than DTW for all transformation rates, especially for

[1]Transformation rate of add noise function is one-tenth of x-axis's value, the follow figures use same setting for transformation rate of add noise. The y-axis distance value is defined by definition of trajectory similarity measures

increasing sampling rate, adding noise and synchronized shifting function.

The performance of PDTW with different transformation distance is shown in Figure 12. Unlike DTW, PDTW is not sensitive to length of distance of transformation. As a result, PDTW may work well in measuring the similarity for trajectories with a large number of inaccurate points (i.e, point with large deviation from its true location).

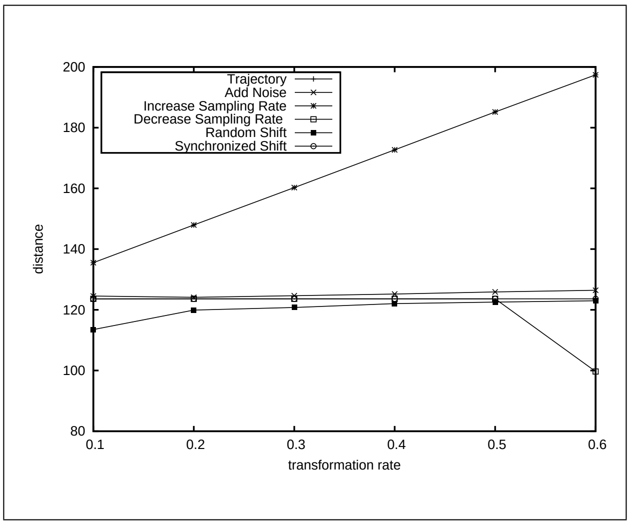

Figure 13: Result of EDR with different transformation rate and ε less than distance of transformation

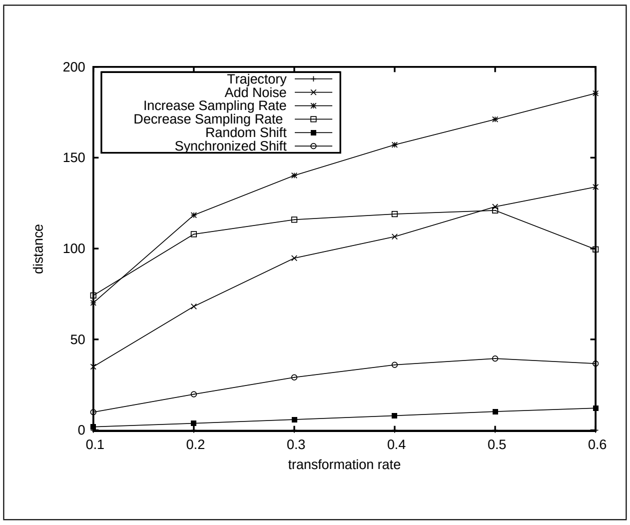

Figure 14: Result of EDR with different transformation rate and ε larger than distance of transformation

Edit Distance based Measures. We first evaluate the effectiveness of EDR with different transformation rate. In Figure 14, the distances between the original and transformed trajectories are all large, since the transformation distance of the sample points is larger than its threshold ε. In this case, most shifted points will have no matched point in the original trajectory, hence increasing the EDR distance.

We also conduct another set of experiments by restricting the transformation distance to be smaller than ε, the result of which is shown in Figure 14. Based on the result, we can see that EDR is very sensitive to altering sampling rate or adding noise. However, EDR still serves as a good distance measure for handling sampling point shift.

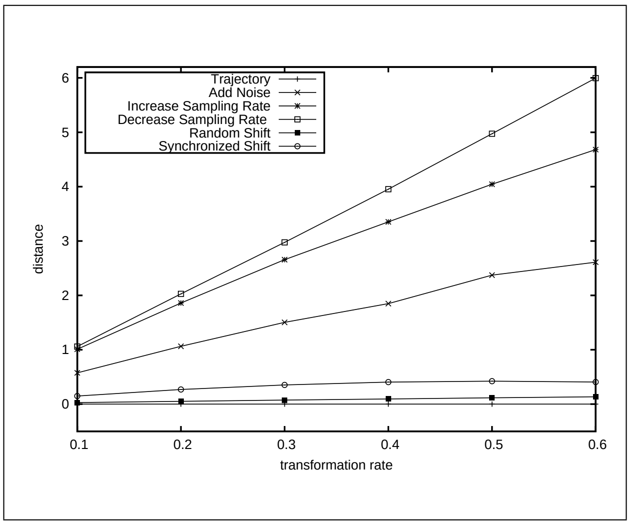

Figure 15: Result of ERP with different transformation rate

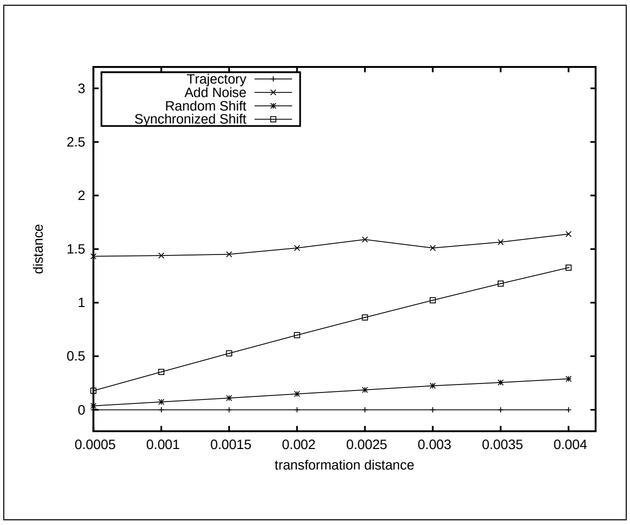

Figure 16: Result of ERP with different transformation distance

Second, we illustrate the experiment result of ERP distance with different transformation rates in Figure 15. From the result we observe that ERP is robust to sample points shifting. Even the transformation rate goes very high, ERP still achieve good performance in capturing the similarity between trajectories with sample point shifting. ERP can also handle the trajectories with a small number of noises. However, it is sensitive to the changes in sampling rate of trajectories.

Figure 16 shows the experiment result of ERP distance with different transformation distance. Based on this distance measure, the transformed trajectory with random shift is very similar to the original one, which means ERP distance is robust to random shift. But still, ERP is quite sensitive to noisy data since adding noise to the original trajectory will result in a large distance value.

Table 2: Comparative Results of Trajectory Similarity Measures

	Euclidean Distance	**DTW**	**PDTW**	**EDR**	**ERP**	**LCSS**
Add noise	Sensitive	Sensitive	Fair	Sensitive	Sensitive	Robust
Increasing sampling rate	Sensitive	Fair	Fair	Sensitive	Sensitive	Robust
Decrease sampling rate	Sensitive	Sensitive	Sensitive	Fair	Fair	Sensitive
Random shift	Robust	Robust	Robust	Robust	Robust	Fair
Synchronized shift	Robust	Sensitive	Robust	Robust	Robust	Fair

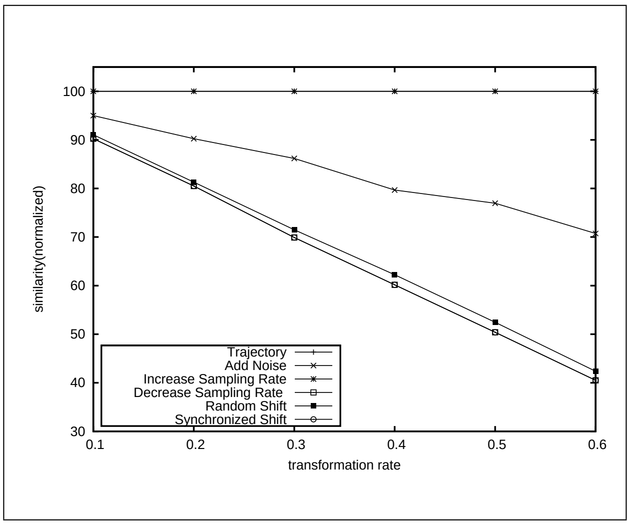

Figure 17: Result of LCSS with different transformation rate and ε less than distance of transformation

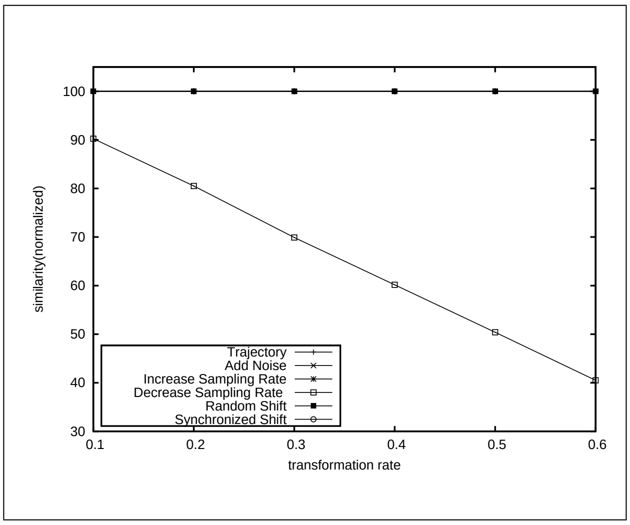

Figure 18: Result of LCSS different transformation rate and ε larger than distance of transformation

Longest Common Subsequence Measure. Finally, we evaluate the distance based on LCSS for different transformations, which is shown in Figure 17. Due to different definition of LCSS, we use normalized similarity as the output, which is instead of distance. Interestingly, LCSS has perfect performance for the transformation with increasing sampling rate, but bad performance for transformation with decreasing sample rate. This is due to the fact that LCSS is calculated based on common subsequences shared between trajectories, which is not affected by increasing sampling rate. For instance, given a trajectory sequence $Tr = [(1,1,t_1),(2,2,t_2),...,(10,10,t_{10})]$ with length $l = 10$. By increasing sampling rate, we have a transformed trajectory $Tr_I = [(1,1,t_1), (1.5,1.5,t'_1),(2,2,t_2),(2.5,2.5,t'_2),...,(10,10,t_{10})]$ with length $l = 20$; by decreasing sampling rate, we have a transformed trajectory $Tr_D = [(1,1,t_1),(3,3,t_3),...,(9,9,t_9)]$ with length $l = 5$. Clearly, the length of common sequence that is reported by LCSS between Tr and Tr_I is larger than that between Tr and Tr_D. Besides, LCSS has a good performance to process noisy trajectory data, but might not be suitable for measuring trajectories that contain shifted sampling points.

At last, we provide an extra experiment for LCSS by setting threshold ε larger than the transformation distance, the result of which is shown in Figure 18. As expected, the transformed trajectories are all treated as identical to the original one, except the one with decreasing sample rate.

In conclusion, there is no trajectory similarity measure that can beat all the others in every circumstance. Table 2 is shown summarized results for each trajectory similarity measures, which are compared based on transformation functions (first column). There are three levels, "Sensitive" , "Fair" and "Robust", which are illustrated the results. In general, Euclidean distance is a good choice when the trajectory data have similar sampling rate and high quality (small point shift) due to its simplicity in implementation and low computation complexity. PDTW is more robust to most transformations than DTW since it adopts the piece-wise aggregation to the raw trajectory before the distance computation. Edit distance based measures (EDR and ERP) achieve good effectiveness with point shift transformations, but are sensitive to altering sampling rate and outliers. On the contrary, LCSS is almost immune to increasing sampling rate and noises but sensitive to point shift. It seems no similarity measure works well for decreasing sampling rate, implying that processing low-sampling-rate trajectories can be a challenging problem (Lou et al. 2009)(Zheng et al. 2012).

4 Conclusion

In this work, we have made the effort to re-implement the investigated similarity measures and evaluate their effectiveness in an objective manner. Further we have devised a set of transformation functions for trajectory data that can serve as a basis for a comparative efficiency tests. The purpose of this work is to provide a quantitative analysis on the effectiveness of the trajectory similarity measures. We hope this work can serve as a starting point for providing benchmarks for future research on trajectory similarity measures.

Acknowledgment

We wish to thank the anonymous reviewers for several comments and suggestions that have improved the paper. This work was supported by the ARC grants DP120102829, DP110103423.

References

Chen, L. & Ng, R. (2004), On the marriage of lp-norms and edit distance, *in* 'Proceedings of the Thirtieth international conference on Very large data bases-Volume 30', VLDB Endowment, pp. 792–803.

Chen, L., Özsu, M. & Oria, V. (2005), Robust and fast similarity search for moving object trajectories, *in* 'Proceedings of the 2005 ACM SIGMOD international conference on Management of data', ACM, pp. 491–502.

Chen, Z., Shen, H. T. & Zhou, X. (2011), Discovering popular routes from trajectories, *in* 'Data Engineering (ICDE), 2011 IEEE 27th International Conference on', pp. 900 –911.

Chen, Z., Shen, H. T., Zhou, X., Zheng, Y. & Xie, X. (2010), Searching trajectories by locations: an efficiency study, *in* 'Proceedings of the 2010 ACM SIGMOD International Conference on Management of data', SIGMOD '10, ACM, New York, NY, USA, pp. 255–266.

Clarke, F. (1976), 'Optimal solutions to differential inclusions', *Journal of Optimization Theory and Applications* **19**(3), 469–478.

Faloutsos, C., Ranganathan, M. & Manolopoulos, Y. (1994), 'Fast subsequence matching in time-series databases', *SIGMOD Rec.* **23**(2), 419–429.

Ichiye, T. & Karplus, M. (1991), 'Collective motions in proteins: a covariance analysis of atomic fluctuations in molecular dynamics and normal mode simulations', *Proteins: Structure, Function, and Bioinformatics* **11**(3), 205–217.

Jonkery, R., De Leve, G., Van Der Velde, J. & Volgenant, A. (1980), 'Rounding symmetric traveling salesman problems with an asymmetric assignment problem', *Operations Research* pp. 623–627.

Kearney, J. & Hansen, S. (1990), Stream editing for animation, Technical report, DTIC Document.

Keogh, E. & Pazzani, M. (2000), Scaling up dynamic time warping for datamining applications, *in* 'Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 285–289.

Kruskal, J. (1983), 'An overview of sequence comparison: Time warps, string edits, and macromolecules', *SIAM review* pp. 201–237.

Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W. & Huang, Y. (2009), Map-matching for low-sampling-rate gps trajectories, *in* 'Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems', ACM, pp. 352–361.

Myers, C., Rabiner, L. & Rosenberg, A. (1980), 'Performance tradeoffs in dynamic time warping algorithms for isolated word recognition', *Acoustics, Speech and Signal Processing, IEEE Transactions on* **28**(6), 623–635.

Ostendorf, M. & Roukos, S. (1989), 'A stochastic segment model for phoneme-based continuous speech recognition', *Acoustics, Speech and Signal Processing, IEEE Transactions on* **37**(12), 1857–1869.

Pfeifer, P. & Deutsch, S. (1980), 'A three-stage iterative procedure for space-time modeling', *Technometrics* pp. 35–47.

Picton, T., Hunt, M., Mowrey, R., Rodriguez, R. & Maru, J. (1988), 'Evaluation of brain-stem auditory evoked potentials using dynamic time warping', *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section* **71**(3), 212–225.

Priestley, M. (1980), 'State-dependent models: A general approach to non-linear time series analysis', *Journal of Time Series Analysis* **1**(1), 47–71.

Richalet, J., Rault, A., Testud, J. & Papon, J. (1978), 'Model predictive heuristic control:: Applications to industrial processes', *Automatica* **14**(5), 413–428.

Robinson, M. (1990), 'The temporal development of collision cascades in the binary-collision approximation', *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* **48**(1-4), 408–413.

Sakurai, Y., Yoshikawa, M. & Faloutsos, C. (2005), Ftw: fast similarity search under the time warping distance, *in* 'Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems', ACM, pp. 326–337.

Sanderson, A. & Wong, A. (1980), 'Pattern trajectory analysis of nonstationary multivariate data', *Systems, Man and Cybernetics, IEEE Transactions on* **10**(7), 384–392.

Shang, S., Yuan, B., Deng, K., Xie, K., Zheng, K. & Zhou, X. (2012), 'Pnn query processing on compressed trajectories', *GeoInformatica* **16**, 467–496.

Soong, F. & Rosenberg, A. (1988), 'On the use of instantaneous and transitional spectral information in speaker recognition', *Acoustics, Speech and Signal Processing, IEEE Transactions on* **36**(6), 871–879.

Takens, F. (1980), 'Motion under the influence of a strong constraining force', *Global theory of dynamical systems* pp. 425–445.

Xie, K., Deng, K. & Zhou, X. (2009), From trajectories to activities: a spatio-temporal join approach, *in* 'Proceedings of the 2009 International Workshop on Location Based Social Networks', LBSN '09, ACM, New York, NY, USA, pp. 25–32.

Yi, B., Jagadish, H. & Faloutsos, C. (1998), Efficient retrieval of similar time sequences under time warping, *in* 'Data Engineering, 1998. Proceedings., 14th International Conference on', IEEE, pp. 201–208.

Zheng, K., Fung, P. C. & Zhou, X. (2010), K-nearest neighbor search for fuzzy objects, *in* 'Proceedings of the 2010 ACM SIGMOD International Conference on Management of data', SIGMOD '10, ACM, New York, NY, USA, pp. 699–710.

Zheng, K., Trajcevski, G., Zhou, X. & Scheuermann, P. (2011), Probabilistic range queries for uncertain trajectories on road networks, *in* 'Proceedings of the 14th International Conference on Extending Database Technology', EDBT/ICDT '11, ACM, New York, NY, USA, pp. 283–294.

Zheng, K., Zheng, Y., Xie, X. & Zhou, X. (2012), Reducing uncertainty of low-sampling-rate trajectories, *in* 'Data Engineering (ICDE), 2012 IEEE 28th International Conference on', pp. 1144 –1155.

Zheng, K., Zhou, X., Fung, P. & Xie, K. (n.d.), 'Spatial query processing for fuzzy objects', *The VLDB Journal* pp. 1–23. 10.1007/s00778-012-0266-x.

Zheng, Y., Zhang, L., Xie, X. & Ma, W. (2009), Mining interesting locations and travel sequences from gps trajectories, *in* 'Proceedings of the 18th international conference on World wide web', ACM, pp. 791–800.