Trajectory Similarity Measures

Kevin Toohey, Matt Duckham
University of Melbourne, Australia

Abstract

Storing, querying, and analyzing trajectories is becoming increasingly important, as the availability and volumes of trajectory data increases. One important class of trajectory analysis is computing trajectory similarity. This paper introduces and compares four of the most common measures of trajectory similarity: longest common subsequence (LCSS), Fréchet distance, dynamic time warping (DTW), and edit distance. These four measures have been implemented in a new open source R package, freely available on CRAN [19]. The paper highlights some of the differences between these four similarity measures, using real trajectory data, in addition to indicating some of the important emerging applications for measurement of trajectory similarity.

1 Introduction

As the technology for tracking moving objects becomes cheaper and more accurate, the amount and availability of stored movement data is continuing to increase rapidly. Most movement data is captured and stored in the form of trajectories, defined as "a sequence of time-stamped locations" [11]. However, the analysis of these increasing volumes of trajectory data can be challenging, due in large part to the way the same continuous movement can have innumerable different discretized trajectory representations.

One important class of trajectory analysis is the measurement of similarity between trajectories. Several measures exist for calculating the similarity between two trajectories, each with their own strengths and weaknesses. Several surveys of trajectory similarity measures have been performed [7, 11, 18]. After first outlining some of the useful applications of trajectory similarity measures, four of the most commonly used similarity measures will be discussed in detail: longest common subsequence (LCSS), Fréchet distance, dynamic time warping (DTW), and edit distance. These four measures have been implemented within a new R package called "SimilarityMeasures," available on CRAN [19]. The four similarity measures are compared empirically using a sample movement dataset, highlighting where differences in computed similarity value are expected to occur.

2 Applications of trajectory similarity measures

The most common use of trajectory similarity measures is for database indexing. For example, Vlachos et al. apply the longest common subsequence similarity measure to index a set of marine animal trajectories [20]. The study shows considerable speed increases for nearest neighbor computations when using this index over brute force linear scans. Other examples of indexing trajectories using similarity measures can be seen in [6, 8, 14].

Movement patterns in vehicle and pedestrian traffic have of course been analyzed using trajectory similarity measures. Information about the similarities in movement patterns can enable traffic managers to adjust timings on a road network, to find where problems are occurring, or to increase safety and security. Suspicious behavior, for example, can be detected from dissimilarity from predefined "normal" behavior [10].

Using Fréchet and discrete Fréchet similarity measures, Buchin et al. [3] explored the detection of commuting patterns in trajectories. Li et al. [16] used longest common subsequence similarity measures to compare calculated paths with actual paths in an analysis on crowded scene movements. Use of tracking data in sports is also becoming more common. Analyzing tracking data from sports can allow players to increase their efficiency and effectiveness. Haase and Brefeld [12] used a dynamic time warping similarity measure to explore similar movements in a soccer game, while Perše et al. [17] analyzed and segmented basketball games using an edit distance similarity measure.

Similarity measures can be used on animal trajectories in behavioral science to explore information about popular tracks, movements, and social interactions [10], such as to compare movements of albatross [4] or of cattle [15].

3 Trajectory similarity computation

A trajectory T_A, contains a series of m timestamped n dimensional points $a_i = (a_{i,1}, \ldots, a_{i,n})$:

$$T_A = ((t_1, a_1), \ldots, (t_m, a_m))$$

where t_i are discrete timestamps and $t_i < t_{i+1}$. The *length* of a trajectory is defined here as the number of discrete timestamps ("fixes"). Spatial trajectory points are commonly recorded in 2 or 3 dimensions, although higher dimensionality trajectories are of course possible.

The key challenge in identifying a satisfactory trajectory similarity measure is the arbitrary nature of the discretization. For example, a naïve similarity measure is Euclidean distance, calculated as the sum of the distances between ordered pairs of points in two trajectories. However, such a simple measure struggles with different sampling rates, outliers, and requires trajectories of different lengths to be cut to equal size [20]. Thus, many more sophisticated similarity measures have been proposed and implemented to overcome the challenges resulting from discretization.

Four of the most commonly encountered advanced similarity measures are explored in the next sections. Three of the discussed measures were included in a previous effectiveness study of six similarity measures tested on a taxi dataset [21], while two were also contained in a comparison of another six similarity measures in [23]. A fully documented R package, called "SimilarityMeasures," is freely available online [19], and has been written to enable easier access to these analyses. The functions in this package are able to compute each of the following similarity measures on n-dimensional trajectories. Please see the package documentation for further details, including how to use the various functions. More information on R, and R packages, can be found at the R Project web page[1].

3.1 Fréchet metric

The Fréchet metric (or Fréchet distance) is amongst the most popular of similarity measures [11]. The metric was first defined by Fréchet [9] and can be applied to both continuous directed curves as well as the discretized trajectories considered here. The Fréchet metric is generally described in the following way: a person is walking a dog on a leash. The person walks on one curve while the dog walks on the other [1]. The dog and the person are able to vary their speeds, or even stop, but not go backwards. The Fréchet metric is the minimum leash length required to complete the traversal of both curves. As with most similarity measures, the choice of distance function can be adapted to suit the specific application and trajectories. Euclidean distance is used in this paper.

[1] `http://www.r-project.org`

Trajectory points are not matched together using the Fréchet metric. This allows the metric to perform well with even the most widely varying sampling rates and trajectory lengths. Unfortunately, the Fréchet metric can be greatly affected by outliers if they are not removed before performing the calculation. This is caused by the fact that *every* point of the two trajectories is used in the calculation.

The Fréchet distance computation contained in our "SimilarityMeasures" R package is implemented using an algorithm discussed by Alt and Godau [1]. This algorithm allows computation of the Fréchet distance between two trajectories of length m and k, with a worst case complexity of $O((m^2k+k^2m)\log mk)$. Alt and Godau use the idea of free space diagrams to allow the efficient calculation of this similarity measure. For more information on the calculations used in this algorithm see Alt and Godau [1].

3.2 Dynamic time warping (DTW)

Unlike Fréchet distance, dynamic time warping (DTW) is a similarity measure that relies on matching points in trajectories. Using DTW, the trajectories are "warped" in a non-linear way to measure similarity while allowing for varying sampling rates [22]. The calculation is again performed using a chosen distance function (Euclidean distance in our examples).

For two trajectories T_A and T_B, with lengths m and k, an $m \times k$ grid can be created where each grid point (i, j) represents the distance between points a_i and b_j [2]. A warping path W is created by starting at grid point $(1, 1)$, and incrementing either i or j or both by 1 each step until reaching point (m, k). For example, a path beginning at grid point $(1, 1)$ could move to one of grid points $(1, 2)$, $(2, 1)$ or $(2, 2)$.

Definition 1: If w_l represents a grid point $(i, j)_l$, then a warping path W can be represented as the following sequence of grid points:

$$W = w_1, \ldots, w_p$$

Exponentially many paths satisfy the conditions above [14]. A warping cost is calculated from a warping path in various ways. A common warping cost for calculating DTW is the total of all of the distances calculated along the warping path. Finally, the DTW similarity value is the minimum of all possible warping costs [2].

Using DTW, a single point on one trajectory can be matched to multiple points on the other. This allows DTW to perform well with trajectories of different lengths and even widely varying sampling rates. However, outliers can again greatly affect this method because every point of both trajectories must have at least one match. The choice of a distance function is also clearly important to DTW, and the warping cost calculation can be changed to suit different needs (cf. Keogh and Ratanamahatana [14] and Keogh and Pazzani [13] for more ways to compute warping cost).

Our R package DTW calculation was implemented using the warping cost algorithm discussed in Berndt and Clifford [2]. This implementation allows the DTW calculation to be performed between two trajectories of length m and k, with complexity of $O(mk)$. This DTW algorithm calculates and returns the total of the distances between each pair of points on the optimal warping path using Euclidean distance.

3.3 Longest common subsequence (LCSS)

Longest common subsequence (LCSS) is a similarity measure where trajectories can be stretched, while some points are able to remain unmatched in an attempt to provide an accurate similarity analysis [20]. The LCSS value represents a count of the maximum number of points which can be considered equivalent, while the trajectories are traversed monotonically from start to end.

Definition 2: Using trajectories T_A and T_B, with lengths m and k, an integer $\delta \geq 0$ and a matching threshold

$\varepsilon \geq 0$, the $LCSS_{\delta,\varepsilon}$ definition from Vlachos et al. [20] is adapted to the following:

$$LCSS_{\delta,\varepsilon}(T_A, T_B) = \begin{cases} 0, & \text{if } T_A \text{ or } T_B \text{ is empty} \\ 1 + LCSS_{\delta,\varepsilon}(Head(T_A), Head(T_B)), & \text{if } |m-k| \leq \delta \text{ and } |a_{m,1} - b_{k,1}| \leq \varepsilon \\ & \text{and} \ldots \text{ and } |a_{m,n} - b_{k,n}| \leq \varepsilon \\ max(LCSS_{\delta,\varepsilon}(Head(T_A), T_B), & \\ \qquad LCSS_{\delta,\varepsilon}(T_A, Head(T_B)), & otherwise, \end{cases}$$

In this definition, the constant δ provides a maximum index difference when comparing points from the two trajectories. The constant ε defines the maximum distance in each dimension allowed for two points to be considered equivalent. Finally, $Head(T_A)$ represents T_A with the last point removed. With careful use of the two constants, δ and ε, this method is highly robust to outliers, while performing well with trajectories of different lengths. The LCSS measure also generally functions well with different sampling rates. However, widely varying sampling rates may cause issues when many points must be left unmatched in the calculation.

The length of the shorter trajectory can be used to normalize this method as an LCSS ratio, allowing for comparisons in the same scale. The LCSS computation implemented in the R package uses the algorithm discussed by Vlachos et al. [20]. Using dynamic programming, the LCSS value for two trajectories with lengths m and k, can be found with a complexity of $O((m+k)\delta)$.

3.4 Edit distance

The fundamental idea of edit distance is to count the minimum number of edits required to make two trajectories equivalent. Several variations of edit distance exist including edit distance with real penalty (ERP) [5] and edit distance on real sequence (EDR) [6]. The discussion below concerns edit distance on real sequence (EDR) as described by Chen et al. [6].

Definition 3: Using a matching threshold $\varepsilon \geq 0$, and trajectories T_A and T_B with lengths m and k, the EDR_ε value (edit distance) defined in Chen et al. [6] is adapted to the following:

$$EDR_\varepsilon(T_A, T_B) = \begin{cases} k, & \text{if } m = 0 \\ m, & \text{if } k = 0 \\ min(EDR_\varepsilon(\textbf{\textit{Rest}}(T_A), \textbf{\textit{Rest}}(T_B)) + \textit{subcost}, & \\ \qquad EDR_\varepsilon(\textbf{\textit{Rest}}(T_A), T_B) + 1, & otherwise \\ \qquad EDR_\varepsilon(T_A, \textbf{\textit{Rest}}(T_B)) + 1), & \end{cases}$$

In this definition, *subcost* $= 0$ if the first point of T_A lies within the matching threshold of the first point of T_B in every dimension, and $subcost = 1$ otherwise. Finally, *Rest*(T_A) represents trajectory T_A with its first point removed (the trajectory now starts from the second point if one exists, otherwise it now has length 0).

EDR is relatively unaffected by outliers because the matching threshold reduces the increments to values of 0 and 1 only [6]. Therefore, even though outliers must still be processed using this method, each outlier can potentially only increase the EDR value by 1, and not some arbitrarily large value as in DTW or Fréchet. EDR also performs well with trajectories that have varying sampling rates. The method does not require trajectories of equal length. However, different length trajectories will automatically inflate the edit distance. This is because every extra point is required to be edited out (or in) for the trajectories to be considered equivalent. This fact needs to be considered when choosing this method in practical applications.

Edit distance on real sequence (EDR), as discussed in Chen et al. [6], was implemented as the edit distance function in our R package. Dynamic programming was used to obtain an efficient calculation of EDR. With two trajectories T_A and T_B, of length m and k, this implementation has a complexity of $O(mk)$.

4 Comparison and evaluation

A sample dataset, containing trajectory data of delivery drivers in the UK, was used to help evaluate the similarity measures discussed in this paper. Our of a total of 23,400 segmented trajectories in our data set, a small sample of 50 randomly chosen pairs of trajectories was used for the analysis. All calculations were performed using the R package discussed earlier.

4.1 Normalization

The trajectories in the dataset vary significantly in terms of location and scale. This limits the amount of useful comparisons and analysis between raw, untransformed trajectories. Therefore, each pair of trajectories was normalized to approximately match, to enable meaningful comparisons across the four similarity measures. The trajectories were rotated, scaled, and translated to align their start and end points.

This normalization, however, does lead to some bias in the results, which must be taken into account. As a result of the normalization, the LCSS calculation is guaranteed to contain a minimum of two points (start and end) if the index spacing distance allows it, while the edit distance calculation is guaranteed to have two points which don't require edits. This problem can also be seen in DTW where the start and end points will both add zero distance to the final value. Although this fact changes all of the absolute values in a constant way, the ratio values are altered in a non-linear manner, and this was taken into consideration for the analysis.

4.2 The similarity measures

Each of the four similarity measures was performed on the 50 pairs of trajectories. The allowed point index spacing (for LCSS and DTW) was set to unlimited to allow for the large variance in trajectory length. The distance for points to be considered equivalent (for LCSS and edit distance) was set to 100m. This value was set using the knowledge that most trajectories range from hundreds of meters to several kilometers, and allows for a wide range of values to be obtained in the analysis.

The LCSS, DTW, and edit distance values were converted to ratios. This was done to ensure that the large variances in trajectory length did not dominate the results. The Fréchet distance was left unchanged because it is mainly unaffected by the length of the trajectories. The end points were left out of the ratio calculations to account for the normalization performed earlier. The DTW and edit distances used the larger trajectory length (minus 2 for the end points) to calculate their ratios (e.g. $DTW(T_A, T_B)/(max(|T_A|, |T_B|) - 2)$). The LCSS ratio used the minimum trajectory length (minus 2 for the end points) as discussed earlier.

Table 1 shows the correlations between the different similarity values computed across the four similarity measures[2]. There is a strong (positive) correlation between Fréchet distance and DTW ratio similarity values, while a strong (negative) correlation can be seen between LCSS ratio and edit distance ratio. Correlations in all other pairings of similarity measures are much weaker (Table 1).

Table 1: The correlation coefficients between each of the similarity measures.

	LCSS Ratio	**Fréchet Distance**	**DTW Ratio**
Fréchet Distance	-0.3707	–	
DTW Ratio	-0.4391	0.9587	–
Edit Distance Ratio	-0.8340	0.3202	0.3866

Figure 1 shows scatterplots comparing the two pairs of most highly correlated similarity measures. Although some correlation was expected between the Fréchet distance and DTW ratio (both use absolute distances in their

[2]Strictly speaking, Fréchet, DTW, and edit distance are *dissimilarity* measures (higher values equal greater dissimilarity) while LCSS is a similarity measure (higher values equal greater similarity)

calculation), the strength of the correlation is remarkably strong (correlation coefficient of 0.9587), particularly given that the underlying calculation is rather different (cf. Section 3.1 and 3.2).

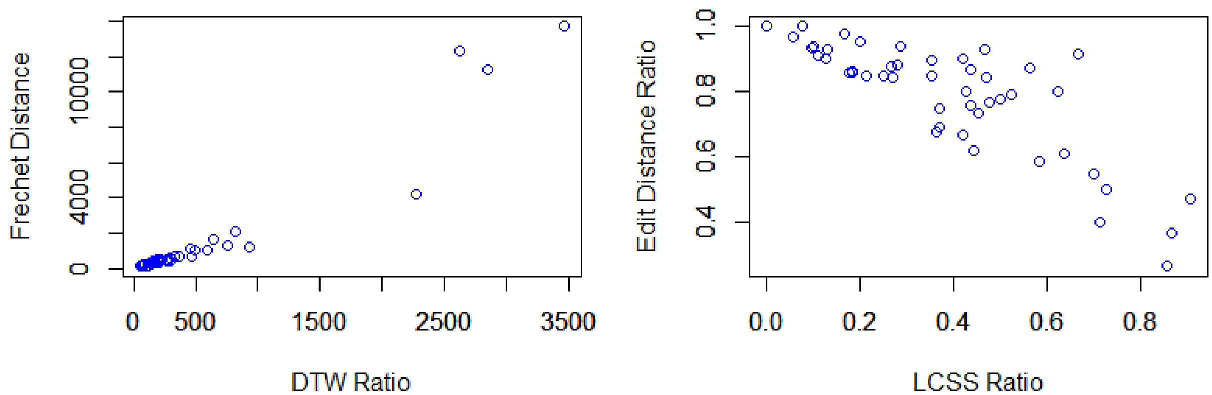

Figure 1: Scatter plots comparing the most largely correlated similarity measures, DTW ratio against Fréchet distance (left) and LCSS ratio against edit distance ratio (right).

LCSS and edit distance also exhibit strong (negative) correlations to one another. As discussed above, the LCSS and edit distances were converted to ratios, and so their values range from 0 to 1 (although 1 is most highly dissimilar in the case of edit distance, and most highly similar in the case of LCSS). Thus, the strong negative correlation coefficient (-0.8340) between the two is in line with the expectation that these methods would yield similar results. The slightly lower correlation seen between the LCSS ratio and edit distance, when compared to Fréchet distance and DTW ratio, is likely caused by the large variations in trajectory length.

Despite the agreement, occasional differences in DTW ratio and Fréchet distance are visible in Figure 1. Large discrepancies are always possible when comparing DTW and Fréchet distances, because they present no bounds on how large the (dis)similarity can be. This is unlike LCSS and edit distance ratios, which have a maximum ratio of 1. The fact that Fréchet distance and DTW values can grow so large makes them sensitive to outliers as discussed earlier. However, this feature can also help to and emphasize extreme cases of trajectory difference. At the other extreme, comparing two equal trajectories using any of the above similarity measures will always yield a value of 0 (1 for LCSS ratio).

These two pairs of measures, Fréchet and DTW, and LCSS and edit distance, appear well correlated in practice for this dataset. However, it is possible to find instances of trajectories that have dramatically different similarity values across these pairs. The left image in Figure 2, for example, shows two trajectories that might be considered similar, although one has a large displacement "peak" near its center. When considering the number of distances combined, this peak will not influence the DTW value greatly. However, the Fréchet distance will reflect considerable dissimilarity in this pair of trajectories, due to the outlying peak.

The right image in 2 presents another pair of trajectories to be considered. Again, these trajectories are relatively similar, although one contains more points clustered in the middle, while the other only contains two points. With a reasonable matching threshold, LCSS will consider these trajectories perfectly equivalent because all of one trajectories points are matched up. However, edit distance highlights that the five middle points have no match and therefore the trajectories are quite dissimilar. The DTW value will also be relatively large because it requires each point to be matched to another, and therefore the trajectories are considered more dissimilar. Finally, the Fréchet distance value will be very low, representing a view of very similar trajectories because it does not require point matching.

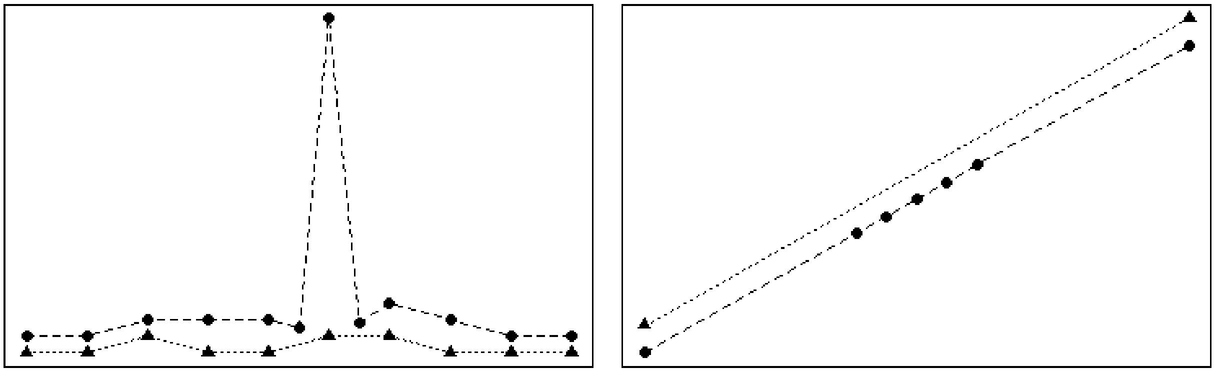

Figure 2: Two pairs of example trajectories. All four trajectories begin in the lower left corner.

5 Concluding remarks

Similarity measures are a useful, and often underused, tool for the analysis of trajectories across a wide range of application. This paper highlights some of the commonalities and differences amongst four of the the most frequently encountered similarity measures, both in theory and in practice. While these measures are today most often used for indexing of spatial databases, they have important applications to the interpretation of pedestrian, vehicle, sport, and animal movement. By implementing an R package capable of easily computing these measures in a single, integrated environment, our aim has been to provide users with easier access to exploring the use of these measures in practical applications.

Acknowledgments

The authors would like to acknowledge collaborators at the 2014 Lorentz Workshop for “Geometric Algorithms in the Field” for providing the initial idea of comparing these four similarity metrics: Kevin Buchin, Patrick Laube, Dongliang Peng, Ross Purves, Stef Sijben, Rodrigo Silveira.

References

[1] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5(01n02):75–91, 1995.

[2] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[3] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. In S.-H. Hong, H. Nagamochi, and T. Fukunaga, editors, *Algorithms and Computation*, volume 5369 of *Lecture Notes in Computer Science*, pages 644–655. Springer, 2008.

[4] M. Buchin, S. Dodge, and B. Speckmann. Similarity of trajectories taking into account geographic context. *Journal of Spatial Information Science*, 9(1):101–124, 2014.

[5] L. Chen and R. Ng. On the marriage of l_p-norms and edit distance. In *Proc. 30th International Conference on Very Large Data Bases*, pages 792–803, 2004.

[6] L. Chen, M. T. Özsu, and V. Oria. Robust and fast similarity search for moving object trajectories. In *Proc. ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 491–502, New York, NY, USA, 2005. ACM.

[7] S. Dodge. *Exploring Movement Using Similarity Analysis*. PhD thesis, Universität Zürich, 2011.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. *SIGMOD Rec.*, 23(2):419–429, May 1994.

[9] M. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo*, 22(1):1–72, 1906.

[10] J. Gudmundsson, P. Laube, and T. Wolle. Movement patterns in spatiotemporal data. In *Encyclopedia of GIS*, pages 726–732. Springer US, 2008.

[11] J. Gudmundsson, P. Laube, and T. Wolle. Computational movement analysis. In *Springer handbook of geographic information*, pages 423–438. Springer, 2012.

[12] J. Haase and U. Brefeld. Finding similar movements in positional data streams. In *Proc. ECML/PKDD Workshop on Machine Learning and Data Mining for Sports Analytics*, 2013.

[13] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proc. 6th ACM International Conference on Knowledge Discovery and Data Mining*, pages 285–289, 2000.

[14] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[15] P. Laube and R. S. Purves. How fast is a cow? Cross-scale analysis of movement data. *Transactions in GIS*, 15(3):401–418, 2011.

[16] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan. Crowded scene analysis: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, (99):1–20, 2014.

[17] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš. A trajectory-based analysis of coordinated team activity in a basketball game. *Computer Vision and Image Understanding*, 113(5):612–621, 2009. Computer Vision Based Analysis in Sport Environments.

[18] P. Ranacher and K. Tzavella. How to compare movement? A review of physical movement similarity measures in geographic information science and beyond. *Cartography and Geographic Information Science*, 41(3):286–307, 2014.

[19] K. Toohey. *SimilarityMeasures: Trajectory Similarity Measures*, 2015. R package version 1.4, `http://CRAN.R-project.org/package=SimilarityMeasures`.

[20] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proc. 18th International Conference on Data Engineering (ICDE)*, pages 673–684, 2002.

[21] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou. An effectiveness study on trajectory similarity measures. In *Proc. 24th Australasian Database Conference*, volume 137, pages 13–22, 2013.

[22] Y. Yuan. *Image-based gesture recognition with support vector machines*. ProQuest, 2008.

[23] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. 18th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 1135–1138, 2006.