A Comparative Analysis of Trajectory Similarity Measures: Recommendations for Selection and Use

Alan Both[1] Kevin Buchin[2], Matt Duckham[1], Patrick Laube[3], Dongliang Peng[4], Ross S. Purves[5], Stef Sijben[2], Rodrigo I. Silveira[6], Yaguang Tao[1], Kevin Toohey[7]

1 School of Science, RMIT University, Melbourne, Victoria, Australia

2 Department of Mathematics and Computer Science, TU Eindhoven, Netherlands

3 Institute of Natural Resource Sciences, Zürich University of Applied Sciences, Wädenswil, Switzerland

4 Institute of Computer Science, University of Würzburg, Würzburg, Germany

5 Department of Geography, University of Zurich, Zurich, Switzerland

6 Departament de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain

7 University of Melbourne, Melbourne, Australia

Authors Buchin, Duckham, Laube, Peng, Purves, Sijben, Silveira contributed equally to the design of this work.
Authors Both, Buchin, Tao, Toohey, Sijben contributed equally to the implementation of this work.
Authors Both, Buchin, Duckham, Laube, Purves, Sijben, Silveira contributed equally to the writing of this paper.

*** alan.both@rmit.edu.au**

Abstract

This paper describes an exploration and comparison of trajectory similarity measures, in order to make practical and generalizable recommendations for researchers seeking to use such measures. Specifically, we compare: dynamic time warping (DTW), edit distance (EDR), longest common subsequence (LCSS), discrete Fréchet distance (DFD), and Fréchet distance (FD). To clarify differences between measures, we calculate similarities for trajectories with contrasting properties: data about bus movements around a transportation network; and the movements of wading birds in a coastal environment. To examine the chosen similarity measures, four experiments were devised: creating a measurement baseline by comparing similarity measures to a single trajectory after applying various transformations; exploring the behavior of similarity measures on network constrained bus trajectories, grouping trajectories based on spatial and on temporal similarity; assessing similarity with respect to existing behavioral labels (flight and foraging of oystercatchers); and comparing bird and bus activity to examine whether they are distinguishable based solely on their movement patterns. Our results and conclusions provide practical advice and recommendations for selecting and using trajectory similarity measures across three complementary perspectives: conceptual, theoretical, and empirical.

1 Introduction

Trajectories—recording the evolving position of objects in geographic space and time—are fundamental building blocks of computational movement analysis [1]. Trajectories have become ubiquitous in a wide range of domains. These domains range across spatial and temporal scales, from analysis at the scale of micro-organisms in laboratory settings in the environmental sciences [2] to global species migrations and interactions [3, 4]. Further, the types of moving objects and movement spaces studied in computational movement analysis vary enormously. Trajectory analysis has been applied to the movement of "crisp" objects, such as the movement of birds, people, and vehicles [5–7], as well as ill-defined objects, such as hurricanes [8]. Trajectory analysis has also been applied to "unconstrained" movement, such as movement ships and aircraft [9], as well as movement within a transportation network, such as the movement of buses and cars [10].

Irrespective of these different settings, a fundamental operation for comparing two trajectories is the measurement of *trajectory similarity.* Measuring trajectory similarity is key to analysis tasks including search (find the most similar trajectory in a collection to a given trajectory, e.g., [11]), clustering (group trajectories with similar properties, e.g., [12]), classification (identifying trajectories associated with a known set of properties, e.g., [13]), and aggregation and characterization (identifying representative trajectories and their properties, e.g., [14]).

Given this wide range of applications, it is perhaps unsurprising that a plethora of methods have been developed with the aim of measuring trajectory similarity. These methods have arisen in parallel in a number of academic communities including (but not limited to) geographic information science [15], computational geometry [11], knowledge discovery and databases [16], movement ecology [17], and transport studies [18]. However, very few studies have, to our knowledge, sought to compare systematically the properties of different measures of trajectory similarity and to make general recommendations on their use.

Two welcome exceptions in this regard are the work of Magdy *et al.* [19] and of Wang *et al.* [20], who explored in an empirical setting the effectiveness of a range of trajectory similarity measures. However, though the latter compared measures, their conclusions are based on a small number of trajectories in a constrained network space, and lack a theoretical underpinning. The former paper briefly characterizes trajectories conceptually, but lacks empirical examples.

Our aim in this paper is to explore trajectory similarity measures from three complementary perspectives: conceptual, theoretical, and empirical. Based on the findings emerging from these perspectives, this paper aims to make useful and generalizable recommendations for researchers seeking to use such similarity measures.

More specifically, in this paper we:

- set out and explore a conceptual model of trajectory similarity, illustrated through a set of examples;
- populate our conceptual model with a set of algorithms and explore their theoretical properties from the perspective of computational geometry; and
- explore experimentally the different properties of selected algorithms through two contrasting datasets (constrained movement of vehicles on a network, and quasi-unconstrained movement of birds in a 2D space).

Through these three strands, the paper aims to not only explore the properties of different algorithmic approaches to calculating similarity, but also to characterize and explore the different ways in which such algorithms impact on analysis of real data.

The following three sections investigate trajectory similarity adopting the three perspectives above: conceptual, theoretical, and empirical. Based on these three perspectives we then set out our general recommendations for the use of trajectory similarity measures.

2 Conceptual modeling of trajectory similarity

2.1 Trajectories

A trajectory represents the path of an object's movement, in general as position in space as a continuous function of time. In practice, however, trajectories are usually captured as "fixes", which are discrete, granular measurements of location at given times. In such cases, both position and time may be regularly or irregularly sampled. In addition to the imprecision introduced through sampling, location in space and in time may be subject to inaccuracy. However, for simplicity in this paper we assume here that trajectory fixes are accurate.

Notation We first introduce notation that will be used throughout the paper. Let A and B be two trajectories of length n and m, respectively, with $A = ((s_1, a_1), \ldots, (s_n, a_n))$ and $B = ((t_1, b_1), \ldots, (t_m, b_m))$, where $a_i, b_i \in \mathbb{R}^2$ are two-dimensional locations and $s_i, t_i \in \mathbb{R}$ are the corresponding time stamps. Given a point $p \in \mathbb{R}^2$, we use $x(p)$ and $y(p)$ to denote the x and y coordinates of p, respectively.

For two points p, q in 2 dimensions, we use $\mathrm{dist}_2(p, q) = \sqrt{(x(p) - x(q))^2 + (y(p) - y(q))^2}$ to denote their Euclidean distance, and $dist_\infty(p, q) = \max(|x(p) - x(q)|, |y(p) - y(q)|)$ to denote their infinity or maximum norm. Finally, for a trajectory A, we use $A_{[i,j]}$ to refer to the sub-trajectory given by points $((s_i, b_i), \ldots, (s_j, b_j))$, for $1 \leq i \leq j \leq n$.

2.2 Similarity

Similarity measures aim to quantify the extent to which two trajectories resemble each other. Comparing two trajectories of the forms introduced above thus involves comparing both their spatial and temporal components. It is important to emphasize that we restrict our focus to spatiotemporal measures of similarity that compare the spatial component based on spatial distance; we do not consider similarity based on shape features like curvature or similarity measures solely using the direction of movement.

2.2.1 Metric versus non-metric

An important property of a similarity measure is whether it is a metric or not. A *metric* is a function that is zero only when the two objects are equal; is symmetric (i.e., distance from A to B equals the distance from B to A); and satisfies the triangle inequality (i.e., for any three trajectories A, B, C, the distance from A to B plus the distance from B to C must be at least as large as the distance from A to C). Metric properties are important for certain applications, such as indexing and clustering. However, not all distance measures are metric (e.g., travel time in transportation networks is a distance measure that is frequently not symmetric).

2.2.2 Discrete versus continuous

Similarity can be measured *continuously*, if the trajectory representation is also continuous and takes all the (infinite) trajectory points into account. Alternatively, similarity measures can be *discrete*, if they consider only a discrete subset of points in the trajectory, most commonly the measured data points (fixes). Hence, discrete measures use only the locations at certain times, ignoring the movement in-between. Continuous measures require interpolation between locations measured at a discrete set of times.

The decision as to whether to use a discrete or a continuous measure depends on several aspects, such as whether the sampling rates in the trajectories are expected to be similar (in density, meaning of the sampling points, etc.); whether interpolation between trajectory points is possible and meaningful; and the fact that discrete measures are typically simpler to implement.

2.2.3 Relative versus absolute time and space

In comparing two trajectories, one can consider space and time as either absolute (i.e., compared with an external spatial and/or temporal reference frame) or relative (i.e., intrinsic comparison, ignoring absolute times or positions). These distinctions are reflected in the behavior of the different trajectory similarity measures, and any transformations or preprocessing that may be applied to trajectory data before similarity analysis.

Absolute time and space Occasionally, it is desirable to compare trajectories that are proximal in both space and time. Such absolute trajectory comparison is quite restrictive, however, as two trajectories must have similar lengths, and be occurring in approximately the same space at the same time. For example, comparing the absolute trajectories of two runners in the same race may give insights to their relative performance.

In practice, it is relatively rare for trajectory similarity measures to be required for two trajectory instances that have such similar positions and speed throughout. Therefore, similarity measures that operate in absolute time and space are not useful for most application scenarios.

Relative time In most applications of trajectory similarity, the temporal references of trajectories are not as important as the spatial characteristics. In such cases, similarity measures are desired that only prioritize similarities in space between trajectories, and limit the influence of temporal differences. For example, in comparing an individual's travel from home to work over the working week, absolute time differences in the day of the week, or even the exact time the journey began, may not be as important as the relative spatial paths of the route taken.

Most of the trajectory similarity measures commonly in use support "local" time transformations, enabling movement at different time periods but across similar locations to be easily aligned and compared. However, even with transformations in time enabled, the sequence of fixes for a trajectory still strongly influences the results.

In such cases, two trajectories that follow spatially identical paths but move in opposite directions (e.g., a route from home to work, versus the same route from work to home) will be measured as dramatically different from each other, even by trajectory similarity measures that support local transformations in time.

Relative space The requirement that trajectories be close in absolute space can also be rather strict for some applications aspiring to mine general patterns from trajectories. For example, two objects do not have to be moving in the same area or even in the same direction to be considered similar if they are engaging in essentially the same behavior. Migration patterns of animals, for example, may exhibit meaningfully similar patterns even if they occur at dramatically different times, locations, and even scales.

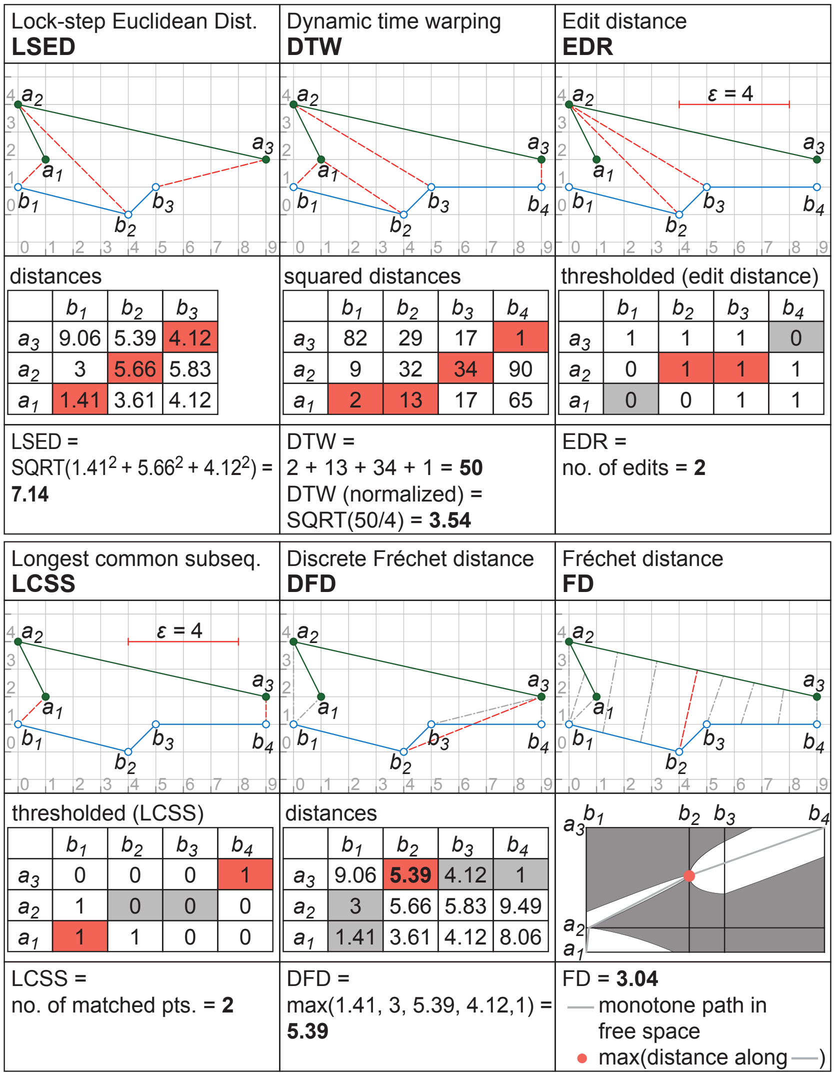
Transformations in space can be performed to align distal trajectories together before similarity measures are applied. Possible spatial transformations include but are not limited to translation, rotation, and scaling. For example, a translation may align trajectories so that they begin at the same point. Rotation can be used to ensure that the direction from the start point to the end point is the same for each trajectory. Additional scaling may also be used to align the start and end points of the trajectories. The type of transformations that are applicable to a specific application are dependent on the specific behaviors of the observed trajectories.

3 Theoretical analysis of similarity measures

For our analysis, we chose six of the most commonly-used trajectory similarity measures. Each measure supports relative time (i.e., is able to support local time transformations), but may treat space differently (i.e., relative or absolute). In addition, our discussion touches upon *lock-step Euclidean distance*, because of its simplicity and the fact that it has been well-tested in the literature (e.g., [21]). Lock-step Euclidean distance is the only measure covered here that does not support relative time, and instead assumes that trajectories occur at the same absolute times.

We begin in each of the following subsections by presenting the basic definition of each similarity measure. Except for unifying notation, we have tried to keep the definitions as close as possible to the variants most widely adopted. Additionally, Fig. 1

serves as a graphical summary of the computation of each measure.



distances

	b_1	b_2	b_3
a_3	9.06	5.39	4.12
a_2	3	5.66	5.83
a_1	1.41	3.61	4.12

squared distances

	b_1	b_2	b_3	b_4
a_3	82	29	17	1
a_2	9	32	34	90
a_1	2	13	17	65

thresholded (edit distance)

	b_1	b_2	b_3	b_4
a_3	1	1	1	0
a_2	0	1	1	1
a_1	0	0	1	1

LSED =
SQRT(1.41^2 + 5.66^2 + 4.12^2) =
7.14

DTW =
2 + 13 + 34 + 1 = **50**
DTW (normalized) =
SQRT(50/4) = **3.54**

EDR =
no. of edits = **2**

thresholded (LCSS)

	b_1	b_2	b_3	b_4
a_3	0	0	0	1
a_2	1	0	0	0
a_1	1	1	0	0

distances

	b_1	b_2	b_3	b_4
a_3	9.06	5.39	4.12	1
a_2	3	5.66	5.83	9.49
a_1	1.41	3.61	4.12	8.06

LCSS =
no. of matched pts. = **2**

DFD =
max(1.41, 3, 5.39, 4.12,1) =
5.39

FD = **3.04**
— monotone path in
free space
● max(distance along —)

Fig 1. Aligning two trajectories where n=3 and m=4 (except for LSED, where n=m=3) according to the various measures, along with a corresponding distance matrix or free-space diagram highlighting in red how each measure value is computed.

3.1 Lock-step Euclidean distance (LSED)

If $n = m$ we can interpret the trajectories as points in the Euclidean space $\mathbb{R}^{2n}$ and take their Euclidean distance.

Definition 1 *The lock-step Euclidean distance distance of A and B is defined as*

$$Eu(A, B) = \sqrt{\sum_{i=1}^{n} dist_2^2(a_i, b_i)} \ .$$

A frequently used variant is the average distance between corresponding measurements:

$$Eu'(A,B) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{dist}_2(a_i, b_i)\ .$$

Alternatively, the maximum instead of the average distance can be used.

If $s_i = t_i$ for all $1 \leq i \leq n = m$ then these distances measure how far the trajectories are apart at corresponding times. In particular, $Eu'(A,B)$ is then the average distance at corresponding times. If we assume uniform sampling in time, then the requirement $n = m$ corresponds to both trajectories having the same duration, i.e., $s_n - s_1 = t_n - t_1$. However, if both trajectories have the same duration but use different sampling—possibly non-uniform, then we can generalize these measures using interpolation:

$$Eu(A,B) = \frac{1}{n}\int_0^{s_n - s_1} \mathrm{dist}_2(A(t - s_1), B(t - t_1))dt\ ,$$

where $A(t)$ and $B(t)$ are the locations of A and B, respectively, obtained by interpolation. Most commonly linear interpolation is used for this, i.e., for $s_i \leq t \leq s_{i+1}$ we have

$$A(t) = a_{i+1}\frac{s_{i+1} - t}{s_{i+1} - s_i} + a_i\frac{t - s_i}{s_{i+1} - s_i}\ . \tag{1}$$

This interpolation assumes that the object moves between two measurements with constant speed along a straight line; an alternative is to bound these distances only assuming an upper bound on the speed of movement [22]. All the distances above are easily computed in $O(n + m)$ time by scanning over the data once.

The Euclidean distance between two trajectories and its variants is widely used, see for instance [23]. A key assumption underlying these measures is that the two trajectories should be aligned using a strict correspondence in time.

All of the following measures relax this condition: data points with different time stamps may be aligned as long as the alignment preserves the order of the points along the trajectories. For all of the measures the alignment is optimized according to certain criteria. The measures differ in the specific criteria.

3.2 Dynamic time warping (DTW)

Dynamic time warping is a classical dynamic-programming algorithm, originally used for speech recognition, which has been successfully applied to time series data since the work by Berndt and Clifford [24]. Later, it became one of the most common methods for measuring similarity between trajectories. The following definition follows the one presented by Chen et al. [25].

Definition 2 *The dynamic time warping distance of A and B is is defined as*

$$DTW(A,B) = \begin{cases} 0 & \textit{if } A \textit{ and } B \textit{ are empty} \\ \infty & \textit{if } A \textit{ or } B \textit{ are empty (not both)} \\ dist_2^2(a_1,b_1) + \min(& \\ DTW(A_{[2,n]}, B_{[2,m]}), & \\ DTW(A, B_{[2,m]}), & \\ DTW(A_{[2,n]}, B)) & \textit{otherwise} \end{cases}$$

Matrix formulation For this algorithm and several of the following ones, it will be insightful to interpret the distance definitions in terms of paths in the distance matrix between the trajectory points, illustrated in Fig. 1 for two sample trajectories A and B. In the figure, the rows and columns of the matrix are laid out such that the squared distance between the first two points is at the lower left and the last two points at the upper right corner of the matrix.

Dynamic time warping can be seen as selecting a minimum cost path in the distance matrix. More precisely, DTW selects a path from the lower left to the upper right corner of the distance matrix that minimizes the sum of squared distances. In the example, the resulting sum is $2 + 13 + 34 + 1 = 50$. DTW is based on defining a cost for aligning two data points, namely the squared Euclidean distance between them.

From the point of view of walking along this path, from the lower left to the upper right corner, at each step DTW considers three possible moves: horizontal, vertical or diagonal. More specifically, the options available are:

1. *Match current pair of points, and move diagonally*: The cost of this move is equal to the squared distance between the pair of points.

2. *Match current pair of points, and move up*: The cost is equal to the squared distance between the pair of points.

3. *Match current pair of points, and move right*: The cost is equal to the squared distance between the pair of points.

Another useful way to visualize the DTW approach is in terms of alignments. Each path in the distance matrix considered by DTW corresponds to an alignment between the points of the two trajectories (red dashed lines, Fig. 1). A path that goes through cell (i, j), for $1 \leq i \leq n$ and $1 \leq j \leq m$, is implicitly aligning a_i with b_j.

What characterizes a similarity measure like DTW is how the cost of a path is defined, since the cost of a path represents how well the two trajectories are aligned in that path. Following [25], we define the measure with squared Euclidean distances above. However, DTW is also frequently used with other costs, e.g., turning angles, discussed in more detail at the end of this section.

Therefore we can say that, in terms of alignments, DTW defines the cost of a path as the sum of the squared distances (or a different cost) between all pairs of aligned points. It is also common to enforce additional constraints on the path, for instance enforcing similar time-stamps between aligned measurements (see for example [26]).

Normalization The DTW distance corresponds to a sum of squared distances between data points and depends on the number of data points used. This makes it difficult to compare DTW distances between different numbers of data points in each trajectory. In the experiments we therefore divide the DTW distance by $\max(m, n)$, which is (in the matrix formulation) the smallest number of cells that need to be visited. To obtain a more intuitive measure, we additionally take the square root, that is, as normalized DTW distance we use $\sqrt{\frac{DTW(A,B)}{\max(m,n)}}$, which produces $\sqrt{\frac{50}{4}} = 3.54$ for the example in Fig. 1.

It might seem natural to normalize using the number of values in the sum (in terms of the matrix formulation: the number of cells visited) instead of $\max(m, n)$. This approach would however make the normalized distance dependent on the path in the matrix, assigning relatively smaller normalized distances to longer paths.

Algorithm The dynamic time warping distance is computed using dynamic programming, meaning that in terms of the formulation above one can compute for every cell (i, j) the cost of the best path to reach it. This computation requires constant time per cell, as a cell's cost can be computed based on the cost of the cell left, below, and diagonally (left-below), resulting in an overall quadratic, i.e., $O(nm)$, computation time. In practice, this can often be reduced to linear time, by carefully avoiding the computation for cells that have no influence on the final result [26].

3.3 Edit distance (EDR)

Another common choice for comparing trajectories are measures based on edit distances. Originally proposed to measure how similar two strings of characters are, edit distances have been successfully used for trajectory similarity. In this section we present the variant proposed by Chen et al. [25], known as *edit distance on real sequence (EDR).*

Definition 3 *The edit distance on real sequence (EDR) of A and B is is defined as*

$$EDR(A, B) = \begin{cases} n & \textit{if } B \textit{ is empty} \\ m & \textit{if } A \textit{ is empty} \\ \min(& \\ EDR(A_{[2,n]}, B_{[2,m]}) + penalty(a_1, b_1), & \\ EDR(A, B_{[2,m]}) + 1, & \\ EDR(A_{[2,n]}, B) + 1) & \textit{otherwise} \end{cases}$$

where $penalty(a_1, b_1)$ is 0 if $dist_\infty(a_1, b_1) < \epsilon$, or 1 otherwise.

The definition uses a parameter ϵ as a matching threshold distance (i.e., two points closer than ϵ are considered to match).

Matrix formulation Similar to DTW, EDR searches for a minimum cost path in the distance matrix, although it uses a matrix where the cost is defined differently. As in DTW, there are three possible ways to move in the path towards the upper right corner:

1. *Match current pair of points, and move diagonally*: The cost of this move depends on whether the current pair of points are considered to match (i.e., their distance

is smaller than ϵ). In the former case there is no cost, in the latter the cost is one unit.

2. *Move right*: This costs one unit.
3. *Move up*: The cost is one unit.

It is important to note that in EDR costs are thresholded to 0 if the current pair of points match, whereas in all other situations the cost is 1, irrespective of the distance between the current pair of points. This results in the *distance threshold matrix*, a Boolean matrix as shown in Fig. 1. However, non-thresholded versions also exist. For instance, EDR itself is an adaptation of a measure proposed by Cai and Ng [27], called *edit distance with real penalty* (ERP), where instead of penalizing with 1 every time two elements do not match, ERP penalizes with the squared distance between the non-matching elements.

In terms of alignments, EDR defines the cost of a path as the number of aligned points that are not considered a match.

Algorithm Computing edit distances can be implemented in the same way as DTW and therefore take quadratic time (i.e., $O(nm)$) in the worst case.

3.4 Longest common subsequence (LCSS)

Measures based on longest common subsequences try to capture how well two trajectories match by measuring the length of the longest subtrajectory in common. LCSS measures are closely related to edit distances, and have also received attention in the context of trajectory similarity. The following definition follows Vlachos et al. [23].

Definition 4 *The length of the longest common subsequence between A and B is defined as*

$$LCSS(A,B) = \begin{cases} 0 & \text{if } A \text{ or } B \text{ is empty} \\ 1 + LCSS(A_{[1,n-1]}, B_{[1,m-1]}) & \text{if } dist_\infty(a_n, b_m) < \epsilon \text{ and} \\ & |n-m| \leq \delta \\ \max(LCSS(A_{[1,n-1]}, B), & \\ LCSS(A, B_{[1,m-1]})) & \text{otherwise} \end{cases}$$

The definition uses two parameters, δ and ϵ. As in EDR, ϵ is a matching threshold distance (i.e., two points closer than ϵ are considered to match). Additionally, δ controls how far in time (specifically, in timesteps) two matching points can be, in order to align the trajectories in time. However, it should be noted that δ is not specific to LCSS, and could be added to any of the other measures.

Matrix formulation LCSS also looks for a path in its distance matrix (Fig. 1), although with a few differences with respect to the previous measures. Firstly, the path is searched in the opposite direction: from the upper right to the lower left corner.[1] Secondly, the goal is to find a path of *maximum* cost, as the objective is to maximize the number of matched points.

As before, there are three possible ways to advance:

1. *Match current pair of points, and move diagonally*: To be considered a match, the pair of points cannot be further than δ timesteps away. This movement gains (costs) one unit

2. *Move left*: This movement results in no gain (cost).

3. *Move down*: This movement results in no gain (cost).

Similar to EDR, LCSS is thresholded, meaning that what is relevant is whether the point pairs match or not, but not their actual distances. In terms of alignments, LCSS defines the value of a path as the number of alignments considered a match, making LCSS a measure that is somewhat complementary to EDR.

Algorithm As before, LCSS can be implemented using dynamic programming, and therefore takes quadratic time (i.e., $O(nm)$) in the worst case.

3.5 Discrete Fréchet distance (DFD)

Proposed by Eiter and Mannila [28], DFD can be seen as a version of DTW that takes the *maximum* distance between aligned points along the path. This is in contrast to DTW, which considers the *sum* of all squared distances.

[1] It is easy to modify the formula to go in the same direction as DTW and EDR, but we preferred to present the original formulation from [23].

Definition 5 *The discrete Fréchet distance of A and B is defined as*

$$DFD(A,B) = \begin{cases} 0 & \textit{if A and B are empty} \\ \infty & \textit{if A or B are empty (not both)} \\ \max(dist_2(a_1,b_1), \min(& \\ DFD(A_{[2,n]}, B_{[2,m]}), & \\ DFD(A, B_{[2,m]}), & \\ DFD(A_{[2,n]}, B)) & \textit{otherwise} \end{cases}$$

Matrix formulation Similar to DTW and EDR, DFD searches for a minimum cost path in the distance matrix, from the lower left to the upper right corner (Fig. 1). As in DTW, the cost of a pair is measured by taking the Euclidean distance, and there are three possible ways to advance:

1. *Match current pair of points, and move diagonally*: The cost is equal to the distance between the pair of points, only if it is greater than or equal to the cost of the rest of the path.

2. *Match current pair of points, and move up*: The cost is equal to the distance between the pair of points, only if it is greater than or equal to the cost of the rest of the path.

3. *Match current pair of points, and move right*: The cost is equal to the distance between the pair of points, only if it is greater than or equal to the cost of the rest of the path.

In terms of alignments, DFD defines the cost of a path as the maximum over the distances between all pairs of aligned points. Note that taking the squared distance instead of the distance would result in the same optimal paths. Essentially, DFD's difference from DTW is that it takes the maximum instead of the sum of the distances between all pairs of aligned points.

Algorithm As before, DFD can be implemented using dynamic programming, resulting in an $O(nm)$-time algorithm.

3.6 Fréchet distance (FD)

All the distance measures above are discrete, in the sense that they only align the measured locations a_i, b_i. This can potentially lead to problems for non-uniform sampling. In contrast, the Fréchet distance [29] is *continuous*, meaning that it considers alignments between all points in both trajectories, by using the interpolated trajectories $A(s), B(t)$ (defined as in Formula 1).

Definition 6 *The Fréchet distance between A and B is defined as*

$$F(A, B) = \inf_{\sigma} \max_{t \in [s_1, s_n]} dist_2(A(t), B(\sigma(t))),$$

where the infimum is taken over all continuous, strictly monotone-increasing functions $\sigma \colon [s_1, s_n] \to [t_1, t_m]$ *(i.e., all continuous alignments).*

This can be interpreted as the continuous version of the discrete Fréchet distance. Similarly, continuous versions of the other measures discussed above have been proposed.

Algorithm Most algorithms to compute the Fréchet distance solve the decision problem as a subroutine. The decision problem for the Fréchet distance is to decide whether the Fréchet distance is smaller than a given $\epsilon > 0$. Given an algorithm for the decision problem, the actual distance can be approximated using for instance a binary search over ϵ. A more complex search procedure, such as parametric search, can be used to compute the Fréchet distance exactly [29].

The decision problem can be solved by a simple dynamic programming algorithm. Consider the so-called *free-space diagram* in Fig. 1, which is the continuous analog to the distance threshold matrix that is used for the edit distance and longest common subsequence problem. In this diagram the vertical axis corresponds to the parameter space of A and the horizontal axis to the parameter space of B. Thus, the point (s, t) in the diagram corresponds to the pair of points $(A(s), B(t))$. The free-space diagram for a given $\epsilon > 0$ is the set of points (s, t) with the property that the distance between $A(s)$ and $B(t)$ is at most ϵ.

In the figure, the free-space diagram for $\epsilon \approx 3.04$ is the white-colored region, and the monotone path in free-space is grey. The Fréchet distance is at most ϵ if and only if there is a path from the lower-left corner to the upper-right corner that goes through the free-space and is monotonically increasing in both coordinates. To compute whether such a path exists we can incrementally compute the part of the free-space diagram that is reachable by such a path. This results in an $O(mn)$-time algorithm for the decision problem. Computing the exact Fréchet distance then requires an additional $\log(mn)$ factor for the parametric search [29]. In the example of Fig. 1 the exact Fréchet distance is approximately 3.04 as the white region would disconnect when ϵ is decreased further. The corresponding alignment is illustrated as a dashed red line.

3.7 Lessons from the conceptual and theoretical analysis

Following our pen-and-paper conceptual and theoretical analysis, this section summarizes the main aspects in which the measures differ.

3.7.1 Metric versus non-metric

When a similarity measure is used for clustering and indexing, it is desirable that the chosen measure is metric. Among the measures discussed, LSED, DFD, and FD are metrics, whereas the others are not because:

- DTW does not obey the triangle inequality;
- LCSS does not measure difference (instead measuring, to some extent, similarity), although variants that satisfy some weaker conditions can be defined [23]; and
- EDR does not fulfill two of the conditions, namely the identity of indiscernibles ($D(A, B) = 0$ if and only if $A = B$) and the triangle inequality ($D(A, B) + D(B, C) \geq D(A, C)$) are not preserved by EDR.

However, it should be noted that edit distance in general may be a metric, including some variants of edit distance used for time-series analysis, such as *edit distance with real penalty* [27].

3.7.2 Discrete versus continuous

As the name suggests, Fréchet distance (FD) is the continuous version of discrete Fréchet distance (DFD). It is this continuity that makes FD distinct from the other similarity measures considered here. The goal for FD is to find a continuous alignment; that is, one between the complete path of both trajectories, as opposed to an alignment between only the trajectory points. While implementing a continuous measure does present additional computational challenges, as opposed to the relative simplicity of a discrete measure, the worst-case running time of the FD is only slightly worse than of the other measures.

While we only discuss the FD as continuous version of the DFD, continuous versions of some other measures have also been defined. The so-called "partial Fréchet distance" [30] is the continuous analogue of LCSS. For a given $\epsilon > 0$, the partial Fréchet distance aligns two trajectories to maximize the parts that have distance at most ϵ, measuring the overall length of these parts. The summed or average Fréchet distance is a continuous version of dynamic time warping, and aligns the trajectories as to minimize the average distance between matched points [31]. Continuous versions of dynamic time warping using other measures for the pairwise distance between matched points have also been considered [32].

3.7.3 Relative versus absolute time

LSED is a similarity measure that only compares measurements at corresponding times. As such, LSED takes a very different approach from the other measures discussed, which aim at temporally aligning the two trajectories.

Common to all of the other discussed similarity measures, DTW, ED, LCSS, DFD, and FD, is the guiding principle of temporally aligning the two trajectories by aggregating the local costs (i.e., the cost of the temporal alignment between each pair of points). The key differences between measures often lie in the details of how this is done. For instance, DTW and DFD fundamentally differ only on whether to take the sum (DTW) or the maximum (DFD) of the local costs.

3.7.4 Relative versus absolute space

The distance measures as considered above align trajectories in time to minimize absolute Euclidean distances. However, depending on the application, relative distance may be more relevant. For instance, one might want to consider a trajectory and a shifted copy of it as essentially the same. This is addressed in two different ways.

The first approach is to take one of the measures above and optimize it under a suitable set of transformations, e.g., translations. That is, if $D(A, B)$ is a distance measure between trajectories A and B, one would consider $\min\{d(A, B + \tau) \mid \tau \in T\}$, where T is the set of two-dimensional translations. This minimization problem is typically computationally expensive (see for example [23]), and often solved by sampling the space of transformations [33].

The second approach is much simpler: instead of using Euclidean distances, an alternative measure that is invariant under a suitable set of transformations is used. Common choices for this measure include heading (translation-invariant) and turning angle (translation- and rotation-invariant), any of which can be applied to the similarity measures discussed above. For instance, one can use DTW with turning angles instead of squared Euclidean distances. To that end, the turning angle must be defined for each point a_i. This is usually done by considering points a_{i-1} and a_{i+1}, and computing the angle between vectors $\overrightarrow{a_{i-1}a_i}$ and $\overrightarrow{a_i a_{i+1}}$. Note that the use of measures such as heading or turning angle complicates the application of continuous similarity measures such as FD, since it would require to interpolate heading or turning angle between trajectory points. This is, in general, hard to do in a meaningful way; for instance, turning angle is always zero between points if linear interpolation is used.

3.7.5 Computational efficiency

Regarding efficiency, the simplest and fastest measure discussed is LSED, as it only requires processing the input trajectories once, which takes $O(n + m)$ time. All the dynamic programming-based measures, namely DTW, EDR, LCSS and DFD, require $O(nm)$ time, at least in their standard formulations. An advantage of the dynamic programming approach is that it is easy to implement, and it is almost identical for all four measures. Theoretical improvements for some of these measures are possible, see for example [34–36]. However, these are minor improvements on theoretical running times that come at the cost of an increase in the complexity of the implementation. In practice, it is possible to get better running times. For instance for DTW there is an algorithm that typically runs in linear time [26]. FD is the least efficient similarity measure, requiring $O(mn \log mn)$ time.

3.7.6 Outliers

One final difference between the various measures worth highlighting is the way in which they deal with outliers. Generally, measures that use the maximum distance between matched points (such as FD and DFD) emphasize large distances and are therefore more sensitive to outliers than measures that use the sum of distances (or the sum of squared distances). Thresholds (as used in the EDR and LCSS) can be useful for dealing with outliers as they allow for the assignment of a uniform cost to pairs that are matched but have a distance larger than the threshold. In this sense, LCSS can be interpreted as the measure that minimizes the number of points that need to be classified as outliers to perfectly align the remaining trajectories. This, however, comes at the cost of introducing the threshold as an additional parameter.

3.8 A key to choosing a suitable similarity measure

Summarizing our discussion of the theoretical and conceptual differences between each of the similarity measures is investigated above, Table 1 captures the salient differences between the measures. The table provides a visual guide as to whether each similarity measure is a metric; operates on continuous trajectories; temporally aligns trajectories; can be used following spatial transformation of trajectories (relative space following transformation); can be used with a distance measure that is invariant under transformation (relative space using invariant distance); compares the relative computational efficiency; and compares the measure robustness to outliers. Note that for the last two categories more stars indicate better performance.

4 Experiments

As highlighted in Section 2, our aim is not to identify a single similarity measure that fits all situations; rather our aim is to clarify the differences and similarities between measures. The decision on which similarity measure to apply depends on each individual definition of distance, with different applications placing the emphasis on different aspects of the trajectories they compare.

Having already explored these differences from a conceptual and theoretical

Table 1. Summary of differences in similarity measures

	LSED	DTW	EDR	LCSS	DFD	FD
Metric?	Yes	No	No, but see [27]	No	Yes	Yes
Continuous?	No	No, but see [31]	No	No, but see [30]	No	Yes
Relative time?	No	Yes	Yes	Yes	Yes	Yes
Relative space following transformation?	Yes	Yes	Yes	Yes	Yes	Yes
Relative space using invariant distance?	No	Yes	Yes	Yes	Yes	No
Computational efficiency?	⋆⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆
Outliers?	⋆⋆	⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆	⋆

standpoint, in this section we aim to explore these through real data, enabling us to distinguish between differences which may be theoretically important, but practically less relevant.

4.1 Datasets

To clarify the differences between measures, we calculated similarities for trajectories with contrasting properties: vehicle movements through a transportation network, and trajectories capturing the behavior of coastal wading birds.

Our first dataset used is the Dublin bus GPS sample data [37]. This dataset was selected to support experiments exploring the impacts spatial or temporal factors have on similarity measures independently. This dataset is suitable for separating spatial and temporal impacts because the trajectories can be fixed in one dimension and varied in the other. For example, a group of bus trajectories from the same bus route taken at different periods of a day can be identical in their spatial path, but have quite different temporal features. A representative subsample of bus trajectories were extracted from four weekdays (2nd, 3rd, 4th, and 7th of January 2013) and 8–9am, 1–2pm, and 8–9pm time blocks. This selection mechanism allows trajectories to be grouped based on temporal similarity as well as spatial similarity. Trajectory segments at the start or the end of a trajectory, where the bus was considered to be stationary, were removed as such a high level of similarity would likely dominate any similarity measures used. Two example pairs of trajectories are shown in Fig 2.

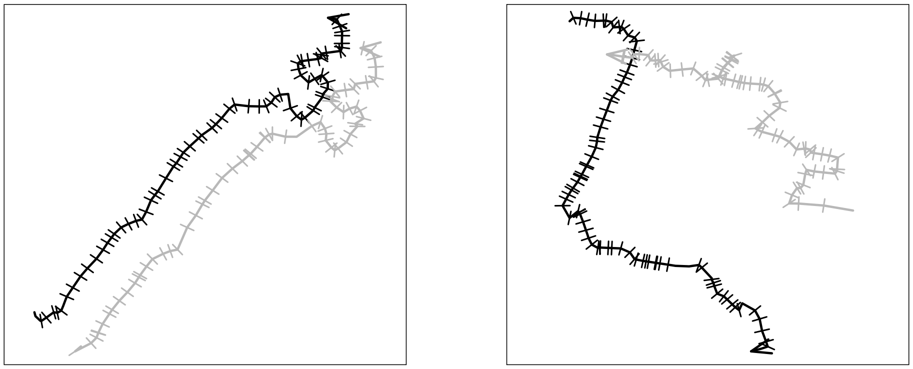


Fig 2. Example bus trajectories. The left pair shows trajectories of the same bus route collected at the same time but on different days; the right pair shows trajectories with different routes and different times. Grey trajectories have been displaced for clarity.

The second dataset used is GPS data on oystercatchers tagged with bird activities [38]. This dataset was selected to provide comparison cases between trajectories with known behavioral differences. Specifically, trajectories governed by the activities of flight and forging were extracted for analysis. A minimum trajectory length of three fixes was set to exclude trajectories that are too short to clearly indicate any embedded activity. This dataset is suitable for tests on similarity measures where transformation in time and space is required, as trajectories reflecting the same activity may occur in different locations, direction, and speeds. It is the features that are invariant under certain spatial or temporal transformations that make such trajectories distinguishable, and therefore represent the different underlying activities. For example, one such feature for oystercatcher trajectories is that birds are more likely to make sudden turns when they are foraging compared to cases when they are simply flying (Fig. 3).

4.2 Experiment 1

Our first experiment examined the similarity measures when applied to a single high resolution trajectory under different transformations (Fig. 4). A base trajectory was

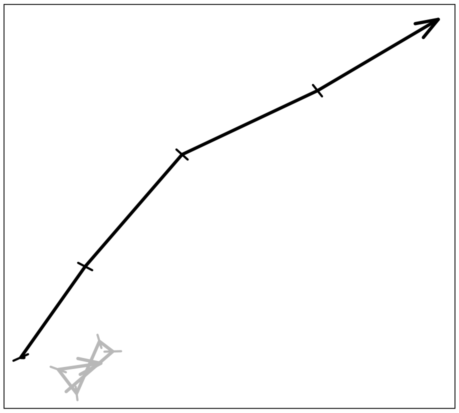


Fig 3. One trajectory of flight (black) and one trajectory of foraging (gray)

created by regular sub-sampling from the raw data (Fig. 4a). Obviously, we expect all trajectory measures to find maximum similarity when trajectories are identical, and such a comparison can be seen as a trivial verification of the implementation of our code. We then calculated similarity under the following conditions:

1. A temporal transformation, where points were sub-sampled from the original trajectory with an increasing temporal interval, clustering points towards the (temporal) beginning of the trajectory (Fig. 4b);

2. A spatial transformation where the base trajectory was rotated slightly about its origin (Fig. 4c); and

3. A spatiotemporal transformation where both temporal and spatial transformations above were applied (Fig. 4d).

In this experiment we are interested in the ordering of results within and between similarity measures. In other words, are the same trajectory pairs always more similar, irrespective of the similarity measure used, or, as we expect from our theoretical analysis, are some measures more sensitive to spatial or temporal transformations than others?

4.3 Experiment 2

In our second experiment, we aimed to explore the behavior of different similarity measures on real trajectories constrained in a network space. Here, we assumed that spatial behavior, while not identical, is very similar for repeated instances of the same route, whereas temporal behavior may vary greatly (i.e., from variations in traffic flow).

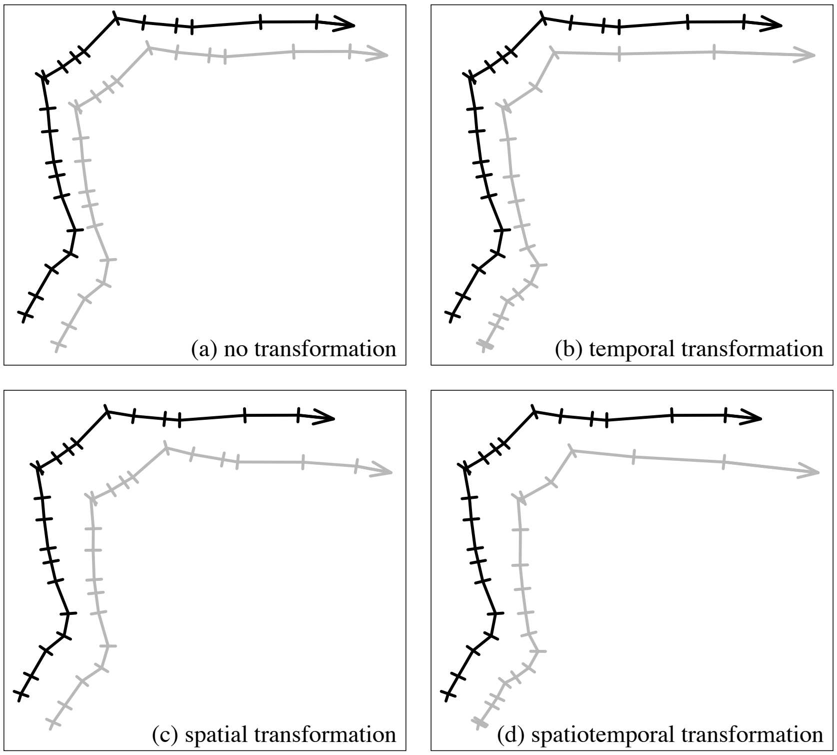

(a) no transformation
(b) temporal transformation
(c) spatial transformation
(d) spatiotemporal transformation

Fig 4. Trajectory comparisons between one bus trajectory and its variations

A key question then is: which similarity measures are better suited to discriminating between trajectories paired from different categories? We chose two dimensions along which to characterize trajectories: spatial similarity, where we select trajectories according to individual bus routes; and temporal similarity, where we select trajectories from the three sampled time periods (8–9am, 1–2pm, and 8–9pm, all on weekdays). These criteria were then used to randomly select 36 pairs of trajectories to test four scenarios:

- *SameSame*: Pairs of trajectories along the same route in the same temporal window.
- *SameRoute*: Pairs of trajectories along the same route in different temporal windows.
- *SameTime*: Pairs of trajectories along different routes in the same temporal window.
- *DiffDiff*: Pairs of trajectories along different routes in different temporal windows.

These four scenarios capture the essential spatial and temporal dimensions of empirical trajectory similarity in network space. We hypothesize that similarity measures will be able to discriminate between trajectories along these dimensions.

4.4 Experiment 3

In Experiment 3 we compared empirical trajectories of oystercatcher data. Here our aim was to assess trajectory similarity with respect to existing behavioral labels (flight and foraging). We hypothesized that similarity measures should be able to discriminate between such behaviors. In this experiment, pairs of trajectories were selected randomly from labeled foraging behavior and from all of the small number of available flight trajectories to build the following comparison sets:

- *FlyFly*: Pairs of trajectories labeled as flight.
- *FlyForage*: Pairs of trajectories labeled as flight and foraging.
- *ForageForage*: Pairs of trajectories labeled as foraging.

These comparisons are a typical example of trajectory similarity comparisons in a between-subjects experiment in ecology, where the aim is to describe animal behaviors using GPS tracks. Since such behavioral patterns need not take place within the same space, and we are interested in the general properties of the trajectories, a necessary first step is a spatial transformation (translation and rotation) to align trajectories. All trajectories were translated such that their origins were identical, and rotated so that the angle formed between the first and last point in every trajectory was 45 degrees. We hypothesized that similarity measures would capture differences between behavioral patterns through spatial behavior.

4.5 Experiment 4

In any experiments comparing methods, it is important to consider straightforward baselines that are easy to interpret. Since the two datasets used exhibit very different properties, another experiment with transformation in space allowed was designed to compare two more general activities—bird activity and bus activity. As the bird and bus trajectories lie far away from each other, transformation in space and time was utilized to enable comparison. The similarity measures were then performed on three types of trajectory pairs:

- *BirdBird*: Pairs of bird trajectories.
- *BusBird*: Pairs of bird and bus trajectories.
- *BusBus*: Pairs of bus trajectories.

Based on this pairing schema, a random set of both bird and bus trajectories were used to examine whether bus and bird are distinguishable based solely on their movement patterns.

4.6 Measure normalization

As the trajectories used in the experiments can vary dramatically in length, a direct comparison of similarity measures is not possible. In order for all similarity measures to be compared within the same categories, and between inter-category groups, LCSS, DTW, and EDR similarity values needed to first be normalized. LCSS was normalized by the shortest trajectory length while DTW and EDR were normalized by the longest trajectory length. As DFD and FD are essentially unaffected by length of trajectories, normalization was unnecessary. Additionally, the threshold value ϵ for LCSS and EDR was set to 100m for the first experiment while 50m was chosen for the later experiments.

5 Results and interpretation

5.1 Experiment 1

In our first experiment we sought to explore the difference between similarity measures under a range of transformations. As introduced above, our expectation is that some similarity measures are more sensitive to spatial or temporal transformations than others. Fig. 4 shows the trajectories compared, while Table 2 shows the values calculated for each similarity measure.

It is important to note that a physical interpretation of the meaning of the similarity measures in all cases except FD and DFD, which can be directly interpreted as a discrete physical distance, distance is difficult. EDR and LCSS are both normalized between 0–1, while DTW is normalized as a function of the number of points in the longest trajectory in a pair (see Section 4.6).

Table 2. Computed similarities between black and gray trajectories in Fig. 4. The ranks in parentheses indicate for every measure the relative order of the computed similarities.

Transformation	None (Fig. 4a)	Temporal (Fig. 4b)	Spatial (Fig. 4c)	Spatiotemporal (Fig. 4d)
LCSS Ratio	1 (1)	0.68 (2)	0.61 (3)	0.55 (4)
EDR Ratio	0 (1)	0.57 (3)	0.43 (2)	0.70 (4)
Fréchet (m)	0 (1)	163.61 (2)	497.84 (3)	497.84 (3)
Discrete Fréchet (m)	0 (1)	456.87 (2)	497.84 (3)	497.84 (3)
DTW (m)	0 (1)	179.64 (2)	270.19 (4)	259.10 (3)

A number of observations can be made from Table 2. Firstly, our implementations of the similarity measures deliver expected values for identical trajectories, a trivial but important sanity check.

Secondly, similarity values are indeed sensitive to the measures used, with variation in the ranking of trajectory similarity between measures as a function of transformations.

Thirdly, in this experiment a clear difference is visible in the distances associated with continuous and discrete Fréchet distance under temporal transformations. This difference arises since under FD distances are calculated between not only data points, but also interpolated segments between these points, and thus the influence of the temporal transformation of the data points is limited.

5.2 Experiment 2

In our second experiment we explored the sensitivity of the similarity measures to comparisons between the trajectories of buses in Dublin for four spatiotemporal scenarios. Our hypothesis was that different similarity measures should be able to discriminate between trajectories that differ spatially and temporally.

Fig. 5 shows box plots of the similarity measures for each case. The first observation that can be made is that spatial difference between trajectories dominates all similarity measures. This is visible in the clear differences across all measures between *SameSame* and *SameRoute* (which compare the same spatial trajectory paths) versus *SameTime* and *DiffDiff* (which compare different routes). By contrast, temporal differences appear to have little to no influence on the properties of the similarity measures.

This observation was confirmed using a Wilcoxon signed rank hypothesis test. The test revealed no significant differences at the 5% level between either the *SameSame* versus *SameRoute* or the *SameTime* versus *DiffDiff* across all measures tested. By contrast, the differences between *SameSame* versus *SameTime/DiffDiff* and between the *SameRoute* versus *SameTime/DiffDiff* are significant at the 5% level in all cases.

This result implies that differences between trajectories on a network, calculated using the measures we investigate, are largely the product of spatial differences. However, the fact that bus routes are oftentimes designed to be spatially different in order to cover more area and share less overlap may be a factor in the lack of similarity between different routes, when compared with different times. Further, bus trajectories collected at different time periods are not necessarily temporally distinct in the way illustrated by the temporal transformation of a trajectory in Experiment 1. Instead, there appeared to be limited difference in the proportion of points at each section of the trajectory. This is likely due to buses following fixed schedules, operating at similar speeds, and stopping with similar frequency.

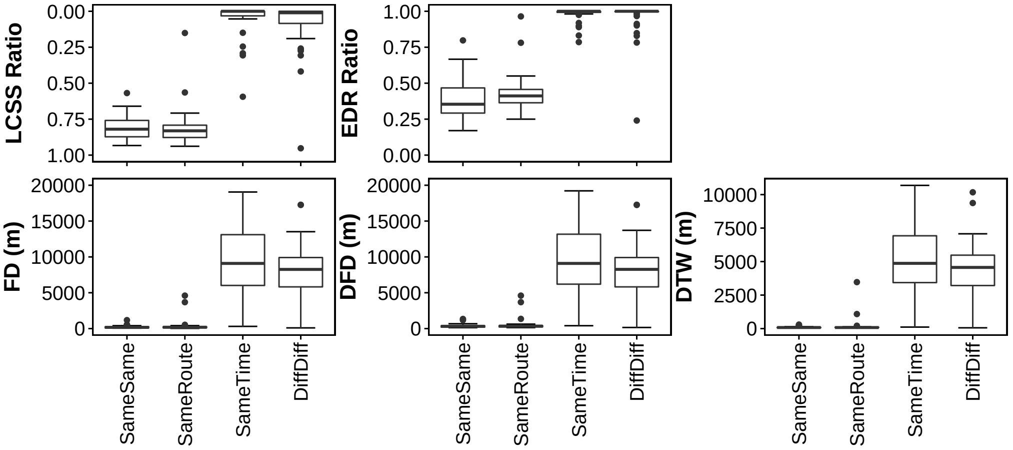


Fig 5. Box plots of bus trajectory similarity

5.3 Experiment 3

In our next set of experiments, we explored the ability of similarity measures to discriminate between different activities with very clear spatiotemporal signatures (cf. Fig. 3). Box plots showing the results for all five similarity measures are shown in Fig. 6. It was expected that trajectories governed by the same activity would be more alike, whereas trajectories for different activities would show greater difference.

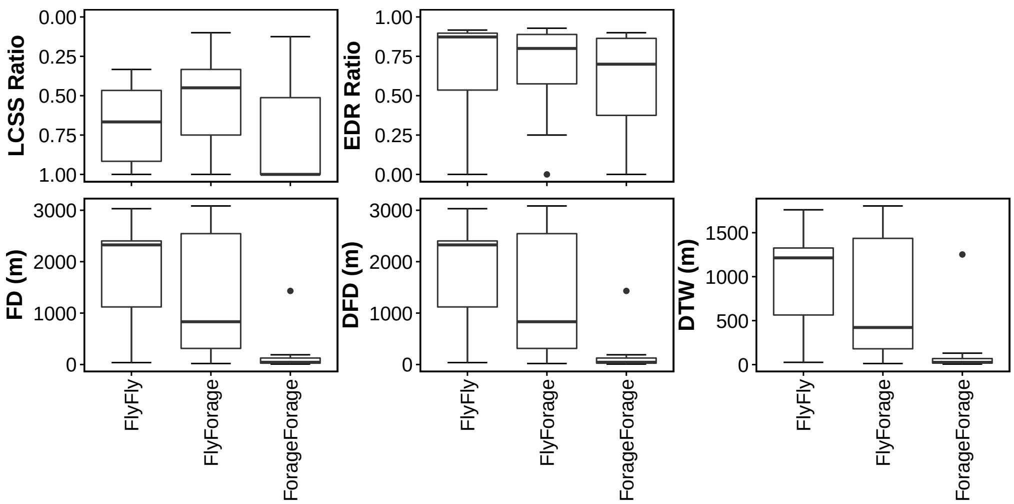


Fig 6. Box plots of bird activity trajectory similarity

In contrast to the previous experiment, the results do show a clear difference between the five similarity measures, with only Fréchet distance, DFD, and DTW showing obvious differences in similarity values. While pairs of foraging trajectories were ranked with higher similarity by Fréchet distance, DFD, and DTW, this was not the case for pairs of flight trajectories, exhibiting little difference from pairs of flying and foraging trajectories. These greater similarity values indicate that foraging trajectories share a large extent of common features that are invariant to transformation. Conversely, the weaker similarity values for the other two groups points to the explanation that one flight trajectory may as dissimilar from other flight trajectories, as it is from foraging trajectories. For the EDR Ratio, there appears to be little or no difference between any of the sets of pairs, indicating that EDR is unable to distinguish the exhibited movement patterns. The LCSS ratio however does appear to exhibit the expected signal (i.e., that pairs of flying and pairs of foraging trajectories have greater similarity than mixed pairs), albeit one that is very weak.

To test whether similarity measures could be treated as being drawn from different populations, according to the semantics of the comparisons, we performed a Kruskal-Wallis rank sum test (Table 3). As suggested by the box plots, we found significant differences ($p<0.05$) between the similarity values for Fréchet distance, DFD, and DTW. To further explore the nature of these differences, we then performed pairwise Wilcoxon signed rank tests to compare the (FlyFly with

FlyForage/ForageForage with FlyForage) (Table 4). We found significant differences (p<0.05) for both measures when comparing foraging behavior with mixed groups of trajectories, but were not able to distinguish between flying behavior from mixed groups. These results, given our previous experiment, imply that the form of trajectories has an influence on the sensitivity of measures to differences.

Table 3. P-values for Kruskal-Wallis test performed on the similarity distribution for analysis on Oystercatcher data.

	P-value	Significant at 5% level
LCSS Ratio	0.3389	
EDR Ratio	0.5583	
Fréchet	0.0057	*
Discrete Fréchet	0.0057	*
DTW	0.0075	*

Table 4. P-values for Wilcoxon signed rank tests for analysis on Oystercatcher data.

	Comparison groups	P-value	Significant at 5% level
FD, DFD	FlyVsFly and FlyVsForage	0.4375	
	ForageVsForage and FlyVsForage	0.0210	*
DTW	FlyVsFly and FlyVsForage	0.5625	
	ForageVsForage and FlyVsForage	0.0210	*

5.4 Experiment 4

In our fourth experiment we compared trajectories from our two datasets, to explore the ability of similarity measures to cluster fundamentally different types of behavior (buses moving in a structured network space versus birds free to move in a potentially unconstrained space). This experiment provides a baseline for all experiments by comparing trajectories from very different domains that are expected to be intrinsically very different (bus movement and bird movement).

Figure 7 shows box plots for trajectories selected from pairs of similar (*BusBus* and *BirdBird*) and dissimilar (*BusBird*) trajectories. Here we note that Fréchet distance, DFD, and DTW all appear to be able to discriminate between semantically similar and dissimilar objects, with largest values (and thus most dissimilar) trajectories associated with the *BusBird* pairs. However, LCSS and EDR, while finding the greatest similarity between *BirdBird* pairs, found either higher dissimilarity (LCSS) or comparably high dissimilarity (EDR) between *BusBus* and *BusBird* pairs. As for Experiment 3, pairwise Wilcoxon signed rank tests were performed in order to determine if there was a significant difference between the three groups of trajectory pairs. With the exception of the EDR ratio on *BusBird* and *BusBus* trajectory pairs, all other comparisons deferred exhibit significant differences at the 5% level.

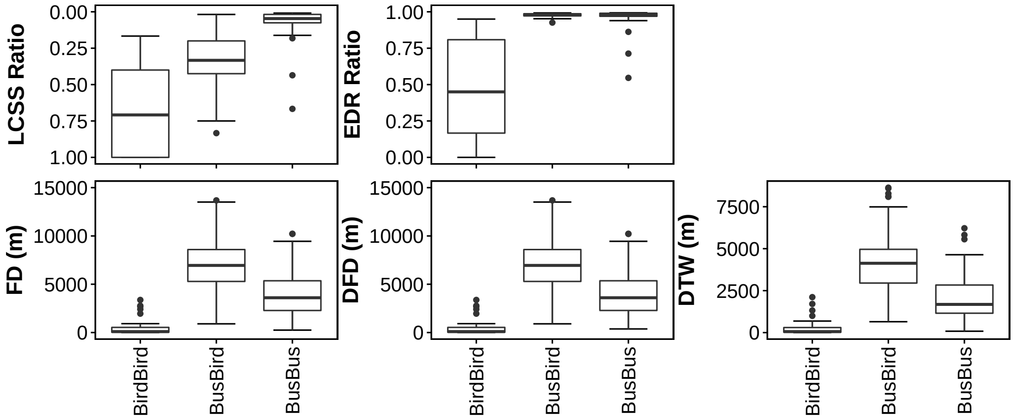


Fig 7. Box plots of bird and bus activity trajectory similarity

5.5 Summary of experimental results

In summary, based on the differences observed across our four experiments, we infer three general and empirical properties of the different similarity measures.

1. Differences in similarity values are sensitive to the choice of measure. The relative ordering of similarity for a fixed set of trajectories can be expected to differ under different similarity measures.

2. All tested similarity measures are better at distinguishing spatially dissimilar trajectories, when compared with temporally dissimilar trajectories. Relatively small spatial differences in trajectories tend to correspond to large differences in the magnitude of measured similarity than even relatively large temporal differences in trajectories.

3. Broadly speaking DTW, DFD, and FD tended to outperform LCSS and EDR in our experiments. In Experiment 3, LCSS and EDR both failed to distinguish trajectories that arose from quite different activities, and were at least visually quite distinct (Fig. 3). In Experiment 4, the similarity values for EDR even failed to enable discernment apart the dissimilarity between different bus trajectories when compared with the dissimilarity between bus and bird trajectories.

6 Conclusions and recommendations

We conclude with the following observations and recommendations on the use of different trajectory similarity measures.

Metric measures Some applications, such as indexing or clustering, rely on similarity measures that offer metric properties. In such cases only some of these similarity measures are suitable (LSED, DFD, FD, and possibly edit distance, although not EDR).

Discrete vs continuous measures The majority of the similarity measures evaluated were discrete (considering only the measured locations for each trajectory). Only Fréchet distance, and its interpolation between measured locations, can provide a measure of difference over continuous trajectory paths, although some continuous analogs of DTW and LCSS can also offer continuous measure properties.

Computational efficiency A major factor to consider when selecting a similarity measure is computational efficiency. In theory, FD is the least efficient measure; LSED the most efficient; with DTW, LCSS, EDR, DFD all underpinned by similar dynamic programming implementations. However, in practice throughout all of the experiments, little to no difference was found when comparing FD to its discrete counterpart. In all cases, the primary influence in execution time is the number of sample points in the trajectories, meaning that over-sampling should be avoided.

Maximum vs sum of distances Similarity measures at root measure either the *maximum* of distance between trajectories (i.e., FD, DFD), or the *sum* of all or a sample of distances between trajectories (LSED, DTW, EDR, LCSS). Different measures in this respect may lend themselves to different applications. As a direct consequence, those measures that are based on maximum distances are much more sensitive to outliers than those based on the sum of distances. That said, in our experiments FD, DFD, and DTW performed similarly, indicating that any outliers present in our datasets were not sufficiently significant to influence the results.

Spatial vs temporal similarity In all of the similarity measures tested, the spatial differences between trajectories were more important in determining the magnitude of measured similarity than temporal differences. This is particularly evident in Experiment 2. However, the precise magnitude of these differences is likely to depend strongly on the specific application.

Thresholds This exploration has not covered the selection of meaningful thresholds for similarity measures that require them, EDR and LCSS. Neither theory nor the experiments in this paper can offer insights into the right thresholds to choose. Thresholds are highly data dependent, and their selection needs to take into account the specific characteristics of the application, including noise, outliers, and constrained or unconstrained spaces for movement.

Bounded versus unbounded measures As noted among the five similarity measures, LCSS and ED can be expressed as ratios, bounded between 0 and 1. Fréchet distance, DFD, and DTW are unbounded positive numbers. Though bounded measures do enable similarity results to be compared across different datasets, they have low resolution when representing high dissimilarity. For example, while it is easy to define 0 in edit distance ratio as two trajectories that are identical, there is no situation where two trajectories are so different that they produce a value of 1. Additionally, this low resolution poses significant issues when different types of trajectories are compared as evidenced by LCSS and EDR ratio's inability to distinguish different movement patterns in Experiment 3.

Acknowledgments

This work was initiated as a result of a Lorentz Center Workshop, *Geometric Algorithms in the Field*, 23–27 June 2014, at Leiden University, Netherlands. Prof Duckham and Dr Both receive research funding support from the Cooperative Research Centre for Spatial Information (CRCSI) under the Rapid Spatial Analytics Program. R. Silveira is supported by projects MINECO MTM2015-63791-R, Gen. Cat. DGR2014SGR46, and by MINECO through the Ramón y Cajal program. P. Laube thanks the Institute of Natural Resource Sciences at the Zurich University of Applied Sciences (ZHAW) for supporting his research.

References

1. Laube P. Computational Movement Analysis. Springer Briefs in Computer Science. Springer; 2014. Available from: `http://dx.doi.org/10.1007/978-3-319-10268-9`.

2. Nathan R, Getz WM, Revilla E, Holyoak M, Kadmon R, Saltz D, et al. A movement ecology paradigm for unifying organismal movement research. Proceedings of the National Academy of Sciences. 2008;105(49):19052–19059. doi:10.1073/pnas.0800375105.

3. Horne JS, Garton EO, Krone SM, Lewis JS. Analyzing Animal Movements Using Brownian Bridges. Ecology. 2007;88(9):2354–2363. doi:10.1890/06-0957.1.

4. Andersson M, Gudmundsson J, Laube P, Wolle T. Reporting Leaders and Followers among Trajectories of Moving Point Objects. GeoInformatica. 2008;12(4):497–528. doi:10.1007/s10707-007-0037-9.

5. Fritz H, Said S, Weimerskirch H. Scale–dependent hierarchical adjustments of movement patterns in a long–range foraging seabird. Proceedings of the Royal Society of London B: Biological Sciences. 2003;270(1520):1143–1148. doi:10.1098/rspb.2003.2350.

6. González MC, Hidalgo CA, Barabási AL. Understanding individual human mobility patterns. Nature. 2008;453(7196):779. doi:10.1038/nature06958.

7. Liu Y, Kang C, Gao S, Xiao Y, Tian Y. Understanding intra-urban trip patterns from taxi trajectory data. Journal of Geographical Systems. 2012;14(4):463–483. doi:10.1007/s10109-012-0166-z.

8. Dodge S, Laube P, Weibel R. Movement similarity assessment using symbolic representation of trajectories. International Journal of Geographical Information Science. 2012;26(9):1563–1588. doi:10.1080/13658816.2011.630003.

9. Kaluza P, Kölzsch A, Gastner MT, Blasius B. The complex network of global cargo ship movements. Journal of The Royal Society Interface. 2010;7(48):1093–1103. doi:10.1098/rsif.2009.0495.

10. Tao Y, Both A, Duckham M. Analytics of movement through checkpoints. International Journal of Geographical Information Science. 2017;0(0):1–22. doi:10.1080/13658816.2017.1397675.

11. Buchin K, Buchin M, van Kreveld MJ, Luo J. Finding long and similar parts of trajectories. Comput Geom. 2011;44(9):465–476. doi:10.1016/j.comgeo.2011.05.004.

12. Zhang Z, Huang K, Tan T. Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. In: 18th International Conference on Pattern Recognition (ICPR 2006), 20-24 August 2006, Hong Kong, China; 2006. p. 1135–1138. Available from: `http://dx.doi.org/10.1109/ICPR.2006.392`.

13. Bashir FI, Khokhar AA, Schonfeld D. Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. IEEE Trans Image Processing. 2007;16(7):1912–1919. doi:10.1109/TIP.2007.898960.

14. Buchin K, Buchin M, van Kreveld MJ, Löffler M, Silveira RI, Wenk C, et al. Median Trajectories. Algorithmica. 2013;66(3):595–614. doi:10.1007/s00453-012-9654-2.

15. Dodge S, Laube P, Weibel R. Movement similarity assessment using symbolic representation of trajectories. International Journal of Geographical Information Science. 2012;26(9):1563–1588. doi:10.1080/13658816.2011.630003.

16. Pelekis N, Kopanakis I, Marketos G, Ntoutsi I, Andrienko GL, Theodoridis Y. Similarity Search in Trajectory Databases. In: 14th International Symposium on Temporal Representation and Reasoning (TIME 2007), 28-30 June 2007, Alicante, Spain; 2007. p. 129–140. Available from: `http://dx.doi.org/10.1109/TIME.2007.59`.

17. Demšar U, Buchin K, Cagnacci F, Safi K, Speckmann B, Van de Weghe N, et al. Analysis and visualisation of movement: an interdisciplinary review. Movement ecology. 2015;3(1):1.

18. Zhang J, Wang F, Wang K, Lin W, Xu X, Chen C. Data-Driven Intelligent Transportation Systems: A Survey. IEEE Trans Intelligent Transportation Systems. 2011;12(4):1624–1639. doi:10.1109/TITS.2011.2158001.

19. Magdy N, Sakr MA, Mostafa T, El-Bahnasy K. Review on trajectory similarity measures. In: Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on. IEEE; 2015. p. 613–619.

20. Wang H, Su H, Zheng K, Sadiq S, Zhou X. An effectiveness study on trajectory similarity measures. In: Proceedings of the Twenty-Fourth Australasian Database Conference-Volume 137. Australian Computer Society, Inc.; 2013. p. 13–22.

21. Nanni M, Pedreschi D. Time-focused clustering of trajectories of moving objects. Journal of Intelligent Information Systems. 2006;27(3):267–289.

22. Buchin M, Purves RS. Computing similarity of coarse and irregular trajectories using space-time prisms. In: Knoblock CA, Schneider M, Kröger P, Krumm J, Widmayer P, editors. 21st SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2013, Orlando, FL, USA, November 5-8, 2013. ACM; 2013. p. 446–449. Available from: http://doi.acm.org/10.1145/2525314.2525459.

23. Vlachos M, Gunopulos D, Kollios G. Discovering Similar Multidimensional Trajectories. In: Agrawal R, Dittrich KR, editors. Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002. IEEE Computer Society; 2002. p. 673–684. Available from: http://dx.doi.org/10.1109/ICDE.2002.994784.

24. Berndt DJ, Clifford J. Using Dynamic Time Warping to Find Patterns in Time Series. In: KDD Workshop; 1994. p. 359–370.

25. Chen L, Özsu MT, Oria V. Robust and Fast Similarity Search for Moving Object Trajectories. In: Proceedings of the ACM SIGMOD International Conference on

Management of Data, Baltimore, Maryland, USA, June 14-16, 2005; 2005. p. 491–502. Available from: http://doi.acm.org/10.1145/1066157.1066213.

26. Keogh E, Ratanamahatana CA. Exact indexing of dynamic time warping. Knowledge and information systems. 2005;7(3):358–386.

27. Cai Y, Ng RT. Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials. In: Weikum G, König AC, Deßloch S, editors. Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004. ACM; 2004. p. 599–610. Available from: http://doi.acm.org/10.1145/1007568.1007636.

28. Eiter T, Mannila H. Computing discrete Fréchet distance. Technische Universität Wien; 1994.

29. Alt H, Godau M. COMPUTING THE FRÉCHET DISTANCE BETWEEN TWO POLYGONAL CURVES. International Journal of Computational Geometry & Applications. 1995;05(01n02):75–91. doi:10.1142/S0218195995000064.

30. Buchin K, Buchin M, Wang Y. Exact Algorithm for Partial Curve Matching via the Fréchet Distance. In: Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA09); 2009. p. 645–654.

31. Buchin M. On the Computability of the Fréchet Distance Between Triangulated Surfaces. Free University Berlin, Institute of Computer Science; 2007.

32. Efrat A, Fan Q, Venkatasubramanian S. Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. Journal of Mathematical Imaging and Vision. 2007;27(3):203–216.

33. Alt H, Scharf L. Shape Matching by Random Sampling. Theoretical Computer Science. 2012;442:2–12.

34. Agrawal R, Dittrich KR, editors. Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002. IEEE Computer Society; 2002.

35. Buchin K, Buchin M, Meulemans W, Mulzer W. Four Soviets Walk the Dog - with an Application to Alt's Conjecture. In: 25th; 2014. p. 1399–1413.

36. Masek WJ, Paterson MS. A faster algorithm computing string edit distances. Journal of Computer and System Sciences. 1980;20(1):18 – 31. doi:http://dx.doi.org/10.1016/0022-0000(80)90002-1.

37. Dublin City Council. Dublin Bus GPS sample data from Dublin city Council (Insight Project); 2013. Available from: http://dublinked.com/datastore/datasets/dataset-304.php.

38. Shamoun-Baranes J, Bom R, van Loon EE, Ens BJ, Oosterbeek K, Bouten W. From sensor data to animal behaviour: an oystercatcher example. PloS one. 2012;7(5):e37997.