Computing the similarity between moving curves

Citation for published version (APA):
Buchin, K., Ophelders, T. A. E., & Speckmann, B. (2015). Computing the similarity between moving curves. In N. Bansal, & I. Finocchi (Eds.), *Proc. 23rd Annual European Symposium on Algorithms (ESA)* (pp. 928-940). (Lecture Notes in Computer Science; Vol. 9294). Springer. https://doi.org/10.1007/978-3-662-48350-3_77

DOI:
10.1007/978-3-662-48350-3_77

Document status and date:
Published: 01/01/2015

Document Version:
Accepted manuscript including changes made at the peer-review stage

Computing the Similarity Between Moving Curves

Kevin Buchin, Tim Ophelders, and Bettina Speckmann

Eindhoven University of Technology
Eindhoven, the Netherlands
[k.a.buchin|t.a.e.ophelders|b.speckmann]@tue.nl

Abstract. In this paper we study similarity measures for moving curves which can, for example, model changing coastlines or glacier termini. Points on a moving curve have two parameters, namely the position along the curve as well as time. We therefore focus on similarity measures for surfaces, specifically the Fréchet distance between surfaces. While the Fréchet distance between surfaces is not even known to be computable, we show for variants arising in the context of moving curves that they are polynomial-time solvable or NP-complete depending on the restrictions imposed on how the moving curves are matched. We achieve the polynomial-time solutions by a novel approach for computing a surface in the so-called free-space diagram based on max-flow min-cut duality.

1 Introduction

Over the past years the availability of devices that can be used to track moving objects has increased dramatically, leading to an explosive growth in movement data. Naturally the goal is not only to track objects but also to extract information from the resulting data. Consequently recent years have seen a significant increase in the development of methods extracting knowledge from moving object data.

Tracking an object gives rise to data describing its movement. Often the scale at which the tracking takes place is such that the tracked objects can be viewed as point objects. Cars driving on a highway, birds foraging for food, or humans walking in a pedestrian zone: for many analysis tasks it is sufficient to consider them as moving points. Hence the most common data sets used in movement data processing are so-called trajectories: sequences of time-stamped points.

However, not all moving objects can be reasonably represented as points. A hurricane can be represented by the position of its eye, but a more accurate description is as a 2-dimensional region which represents the hurricanes extent. When studying shifting coastlines, reducing the coastline to a point is obviously unwanted: one is actually interested in how the whole coast line moves and changes shape over time. The same holds true when studying the terminus of a glacier. In such cases, the moving object is best represented as a polyline rather than by a single point. In this paper we hence go beyond the basic setting of moving point objects and study moving complex, non-point objects. Specifically, we focus on similarity measures for moving curves, based on the Fréchet distance.

Definitions and Notation. The Fréchet distance is a well-studied distance measure for shapes, and is commonly used to determine the similarity between two curves A and $B : [0,1] \to \mathbb{R}^n$. A natural generalization to more complex shapes uses the definition of Equation 1 where the shapes A and B have type $X \to \mathbb{R}^n$.

$$D_{\mathrm{fd}}(A,B) = \inf_{\mu : X \to X} \sup_{x \in X} \|A(x) - B(\mu(x))\| \tag{1}$$

Here, $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a norm such as the Euclidean norm (L^2) or the Manhattan norm (L^1). The *matching* μ ranges over orientation-preserving homeomorphisms (possibly with additional constraints) between the parameter spaces of the shapes compared; as such, it defines a correspondence between the points of the compared shapes. A matching between surfaces with parameters p and t is illustrated in Figure 1. Given one such matching we obtain a distance between A and B by taking the largest distance between any two corresponding points of A and B. The Fréchet distance is the infimum of these distances taken over all possible matchings. For moving points or static curves, we have as parameter space $X = [0,1]$ and for moving curves or static surfaces, we have $X = [0,1]^2$. We can define various similarity measures between shapes by imposing further restrictions on μ.

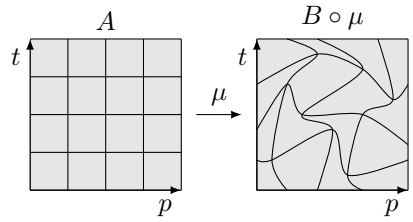


Fig. 1. A matching μ between surfaces A and B drawn as a homeomorphism between their parameter spaces.

In practice a curve is generally represented by a sequence of $P+1$ points. Assuming a linear interpolation between consecutive points, this results in a polyline with P segments. Analogously, a moving curve is a sequence of $T+1$ polylines, each of P segments. We also interpolate the polylines linearly, yielding a bilinear interpolation, or a quadrilateral mesh of $P \times T$ quadrilaterals.

Related Work. The Fréchet distance or related measures are frequently used to evaluate the similarity between point trajectories [8, 7, 13]. The Fréchet distance is also used to match point trajectories to a street network [2, 5]. The Fréchet distance between polygonal curves can be computed in near-quadratic time [3, 6, 9, 17], and approximation algorithms [4, 15] have been studied.

The natural generalization to moving (parameterized) curves is to interpret the curves as surfaces parameterized over time and over the curve parameter. The Fréchet distance between surfaces is NP-hard [16], even for terrains [10]. In terms of positive algorithmic results for general surfaces the Fréchet distance is only known to be semi-computable [1, 12]. Polynomial-time algorithms have been given for the so called weak Fréchet distance [1] and for the Fréchet distance between simple polygons [11] and so called folded polygons [14].

When interpreting moving curves as surfaces it is important to take the different roles of the two parameters into account: the first is inherently linked to time and the other to space. This naturally leads to restricted versions of the Fréchet distance of surfaces. For curves, restricted versions of the Fréchet distance were considered [7, 18]. For surfaces we are not aware of similar results.

1.1 Results

We refine the Fréchet distance between surfaces to meaningfully compare moving curves. To do so, we restrict matchings to be one of several suitable classes. Representative matchings for the considered classes together with the running times of our results are illustrated in Figure 2.

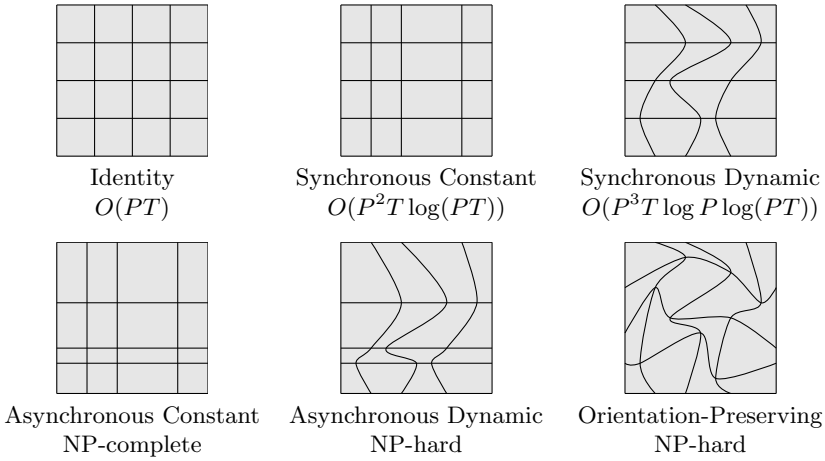


Fig. 2. The time complexities of the considered classes of matchings.

The simplest class of matchings consists of a single predefined *identity* matching $\mu(p,t) = (p,t)$. Hence, to compute the *identity* Fréchet distance, we need only determine a pair of matched points that are furthest apart. It turns out that one of the points of a furthest pair is a vertex of a moving curve (i.e. quadrilateral mesh), allowing computation in $O(PT)$ time. See the full paper for more details.

We also discuss the *synchronous constant* Fréchet distance in the full paper. Here we assume that the matching of timestamps is known in advance, and the matching of positions is the same for each timestamp, so it remains constant. Our algorithm computes the positional matching that minimizes the Fréchet distance.

The *synchronous dynamic* Fréchet distance considered in Section 2 also assumes a predefined matching of timestamps, but does not have the constraint of the *synchronous constant* class that the matching of positions remains constant over time. Instead, the positional matching may change continuously over time.

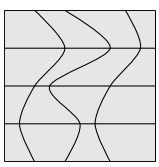
Finally, in Section 3, we consider several cases where neither positional nor temporal matchings are predefined. The three considered cases are the *asynchronous constant*, *asynchronous dynamic*, and *orientation-preserving* Fréchet distance. The asynchronous constant class of matchings consists of a constant (but not predefined) matching of positions, as well as timestamps whereas in the asynchronous dynamic class of matchings, the positional matching may change continuously. In the orientation-preserving class (see the full paper), matchings range over orientation preserving homeomorphisms between the parameter spaces, given that the corners of the parameter space are aligned.

The last three classes are quite complex, and we give constructions proving that approximating the Fréchet distance within a factor 1.5 is NP-hard under

these classes. For the asynchronous constant and asynchronous dynamic classes of matchings, this result holds even for moving curves embedded in $\mathbb{R}^1$ whereas the result for the orientation-preserving case holds for embeddings in $\mathbb{R}^2$.

Although we do not discuss classes where positional matchings are known in advance, these symmetric variants can be obtained by interchanging the time and position parameters for the discussed classes. Deciding which variant is appropriate for comparing two moving curves depends largely on how the data is obtained, as well as the use case for the comparison. For instance, the synchronous constant variant may be used on a sequence of satellite images which have associated timestamps. The synchronous dynamic Fréchet distance is better suited for sensors with different sampling frequencies, placed on curve-like moving objects.

2 Synchronous Dynamic Matchings



Synchronous dynamic matchings align timestamps under the identity matching, but the matching of positions may change continuously over time. Specifically, the matching is defined as $\mu(p,t) = (\pi_t(p), t)$. Here, $\mu(p,t) : [0,P] \times [0,T] \to [0,P] \times [0,T]$ is continuous, and for any t the matching $\pi_t : [0,P] \to [0,P]$ between the two curves at that time is a nondecreasing surjection.

2.1 Freespace Partitions in 2D

The freespace diagram $\mathcal{F}_\varepsilon$ is the set pairs of points that are within distance ε of each other.

$$(x,y) \in \mathcal{F}_\varepsilon \Leftrightarrow \|A(x) - B(y)\| \leq \varepsilon$$

If A and B are curves with parameter space $[0,P]$, then their freespace diagram is two-dimensional, and the Fréchet distance is the minimum value of ε for which an xy-monotone path (representing μ) from $(0,0)$ to (P,P) through the freespace exists.

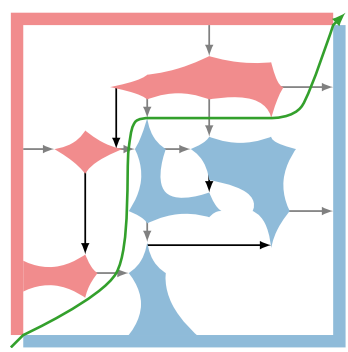


Fig. 3. A matching (green) in the 2D freespace (white).

We use a variant of the max-flow min-cut duality to determine whether a matching through the freespace exists. Before we present the 3D variant for moving curves with synchronized timestamps, we illustrate the idea in the fictional 2D freespace of Figure 3. Here, any matching—such as the green path—must be an x- and y-monotone path from the bottom left to the top right corner and this matching must avoid all obstacles (i.e. all points not in $\mathcal{F}_e$). Therefore each such matching divides the obstacles in two sets: those above, and those below the matching.

Suppose we now draw a directed edge from an obstacle a to an obstacle b if and only if any matching that goes over a must necessarily go over b. The key

observation is that a matching exists unless such edges can form a path from the lower-right boundary to the upper-left boundary of the freespace. In the example, a few trivial edges are drawn in black and gray. If all obstacles were slightly larger, an edge could connect a blue obstacle with a red obstacle, connecting the two boundaries by the edges drawn in black.

2.2 Freespace Partitions in 3D

In contrast to the 2D freespace where the matching is a path, matchings of the form $\mu(p,t) = (\pi_t(p), t)$ form surfaces in the 3D freespace $\mathcal{F}_\varepsilon^{3\mathrm{D}}$ (see Equation 2). Such a surface again divides the obstacles in the freespace in two sets and can be punctured by a path connecting two boundaries. We formalize this concept for the 3D freespace and give an algorithm for deciding the existence of a matching.

$$(x,y,t) \in \mathcal{F}_\varepsilon^{3\mathrm{D}} \text{ if and only if } \|A(x,t) - B(y,t)\| \leq \varepsilon \tag{2}$$

For $x, y, t \in \mathbb{N}$, the cell $C_{x,y,t}$ of the 3D freespace is the set $\mathcal{F}_\varepsilon^{3\mathrm{D}} \cap ([x, x+1] \times [y, y+1] \times [t, t+1])$. The property of Lemma 1 holds for all such cells.

Lemma 1. *A cell $C_{x,y,t}$ of the freespace has a convex intersection with any line parallel to the xy-plane or the t-axis.*

We divide the set of points not in $\mathcal{F}_\varepsilon^{3\mathrm{D}}$ into a set O of so-called obstacles, such that each individual obstacle is a connected point set. Let u be the open set of points representing the left and top boundary of $\mathcal{F}_\varepsilon^{3\mathrm{D}}$. Symmetrically, let d represent the bottom and right boundary, see Figure 4. Denote by $O' \subset O$ the obstacles between the boundaries.

$$\begin{aligned} O &= \{u, d\} \cup O' \text{ with } \bigcup O' = ([0,P]^2 \times [0,T]) \setminus \mathcal{F}_\varepsilon^{3\mathrm{D}}; \\ u &= \{(x,y,t) \mid (x < 0 \wedge y > 0) \vee (x < P \wedge y > P)\}; \\ d &= \{(x,y,t) \mid (x > 0 \wedge y < 0) \vee (x > P \wedge y < P)\}. \end{aligned}$$

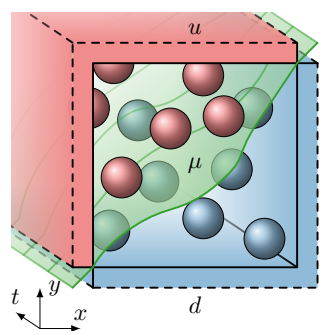


Fig. 4. μ separates u and d.

Given a matching μ, let $D \subseteq O$ be the set of obstacles below it, then $u \notin D$ and $d \in D$. Here, we use axes (x, y, t) and say that a point is below some other point if it has a smaller y-coordinate. Because each obstacle is a connected set and μ cannot intersect obstacles, a single obstacle cannot lie on both sides of the same matching. Because all matchings have $u \notin D$ and $d \in D$, a matching exists if and only if $\neg(d \in D \Rightarrow u \in D)$.

We compute a relation $\triangleright$ of elementary dependencies between obstacles, such that its transitive closure $\trianglerighteq$ has $d \trianglerighteq u$ if and only if $d \in D \Rightarrow u \in D$. Let $a \triangleright b$ if and only if $a \cup b$ is connected (a touches b) or there exists some point $(x_a, y_a, t_a) \in a$ and $(x_b, y_b, t_b) \in b$ with $x_a \leq x_b$, $y_a \geq y_b$ and $t_a = t_b$. We prove in Lemmas 2 and 3 that this choice of $\triangleright$ satisfies the required properties and in Theorem 4 that we can use the transitive closure $\trianglerighteq$ of $\triangleright$ to solve the decision problem of the Fréchet distance.

Lemma 2. *If $a \circledcirc b$, then $a \in D \Rightarrow b \in D$.*

Proof. Assume that $a \triangleright b$, then either a touches b and no matching can separate them, or there exists some $(x_a, y_a, t) \in a$ and $(x_b, y_b, t) \in b$ with $x_a \leq x_b$, $y_a \geq y_b$. If there were some matching μ with $a \in D$, then $(x_a, y_\mu, t) \in \mu$ for some $y_\mu > y_a$. Similarly, if $b \notin D$, then $(x_b, y'_\mu, t) \in \mu$ for some $y'_\mu < y_b$. We can further deduce from $x_a \leq x_b$ and monotonicity of μ that we can pick y'_μ such that $y_a < y_\mu \leq y'_\mu < y_b$. However, this contradicts $y_a \geq y_b$, so such a matching does not exist. Hence, $a \in D \Rightarrow b \in D$ whenever $a \triangleright b$ and therefore whenever $a \circledcirc b$. □

Lemma 3. *If $d \in D \Rightarrow u \in D$, then $d \circledcirc u$.*

Proof. Suppose $d \in D \Rightarrow u \in D$ but not $d \circledcirc u$. Then no matching exists, and no path from d to u exists in the directed graph $G = (O, \triangleright)$. Pick as D the set of obstacles reachable from d in G, then D does not contain u. Pick the *tightest* matching μ such that D lies below it, we define μ in terms of matchings $\pi_t \subseteq \mathbb{R}^2 \times \{t\}$ in the plane at each timestamp t.

$$(x, y, t) \in \pi_t \text{ if and only if } (x' > x \wedge y' < y) \Rightarrow \neg m(x', y', t) \wedge m(x, y, t) \text{ where}$$

$$m(x, y, t) \text{ if and only if } \{(x', y', t) \mid x' \leq x \wedge y' \geq y\} \cap \bigcup D = \emptyset$$

Because $u \notin D$, this defines a monotone path π_t from $(0, 0)$ to (P, P) at each timestamp t. Suppose that π_t *properly* intersects some $o \in O$, such that some point of $(x_o, y_o, t) \in o$ lies below π_t. It follows from the definition of $\triangleright$ and $\neg m(x_o, y_o, t)$ that $d \triangleright o$ for some $d \in D$. However, such obstacle o cannot exist because D satisfies $\triangleright$. As a result, no path π_t intersects any obstacle and we can connect the paths π_t to obtain a continuous matching μ without intersecting any obstacles. So μ does not intersect obstacles in $O \setminus D$, contradicting $d \in D \Rightarrow u \in D$. □

Theorem 4. *The Fréchet distance is greater than ε if and only if $d \circledcirc u$ for ε.*

Proof. We have for every matching that $u \notin D$ and $d \in D$. Therefore it follows from Lemma 2 that no matching exists if $d \circledcirc u$ for ε. In that case, the Fréchet distance is greater than ε. Conversely, if $\neg(d \circledcirc u)$ there is a set D satisfying $\circledcirc$ with $u \notin D$ and $d \in D$. In that case, a matching exists by Lemma 3, and the Fréchet distance is less than ε. □

We choose the set of obstacles O' such that $\bigcup O' = ([0, P]^2 \times [0, T]) \setminus \mathcal{F}_\varepsilon$ and the relation $\triangleright$ is easily computable. Note that due to Lemma 1, each connected component contains a corner of a cell, so any cell in the freespace contains constantly many (up to eight) components of $\bigcup O'$. Moreover, we can index each obstacle in O' by a grid point $(x, y, t) \in \mathbb{N}^3$.

Let $o_{x,y,t} \subseteq ([0, P]^2 \times [0, T]) \cap ([x-1, x+1] \times [y-1, y+1] \times [t-1, t+1]) \setminus \mathcal{F}_\varepsilon$ be the maximal connected subset of the cells adjacent to (x, y, t), such that $o_{x,y,t}$ contains (x, y, t). Now, the obstacle $o_{x,y,t}$ is not well-defined if $(x, y, t) \in \mathcal{F}_\varepsilon$, in which case we define $o_{x,y,t}$ to be an empty (dummy) obstacle. We have $O' = \bigcup_{(x,y,t)} \{o_{x,y,t}\}$ and we remark that obstacles are not necessarily disjoint.

Each of the $O(P^2T)$ obstacles is now defined by a constant number of vertices. We therefore assume that for each pair of obstacles $(a, b) \in O^2$, we can decide in constant time whether $a \triangleright b$; even though this decision procedure depends on the chosen distance metric. For each obstacle $a = o_{x,y,t}$, there are $O(P^2)$ obstacles $b = o_{x',y',t'}$ for which $a \triangleright b$, namely because $t - 2 \leq t' \leq t + 2$ if $a \triangleright b$. Furthermore, u and d contribute to $O(P^2T)$ elements of the relation. Therefore we can compute the relation $\triangleright$ in $O(P^4T)$ time.

Testing whether $d \trianglerighteq u$ is equivalent to testing whether a path from d to u exists in the directed graph $(O, \triangleright)$, which can be decided using a depth first search. So we can solve the decision problem for the Fréchet distance in $O(P^2T + |\triangleright|) = O(P^4T)$ time. However, the relation $\triangleright$ may yield many unnecessary edges. In Section 2.4 we show that a smaller set E of size $O(P^3T)$ with the same transitive closure $\trianglerighteq$ is computable in $O(P^3T \log P)$ time, so the decision algorithm takes only $O(P^3T \log P)$ time.

2.3 Parametric Search

To give an idea of what the 3D freespace looks like, we have drawn the obstacles of the eight cells of the freespace between two quadrilateral meshes of size $P \times T = 2 \times 2$ in Figure 5. Cells of the 3D freespace lie within cubes, having six faces and twelve edges. We call such edges x-, y- or t-edges, depending on the axis to which they are parallel.

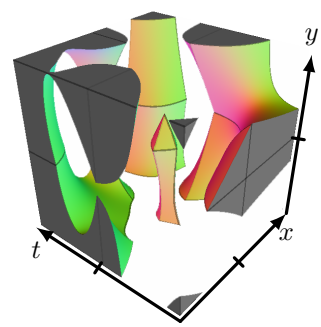


Fig. 5. $[0, 2]^3 \setminus \mathcal{F}_\varepsilon^{3\mathrm{D}}$

We are looking for the minimum value of ε for which a matching exists. When increasing the value of ε, the relation $\triangleright$ becomes sparser since obstacles shrink. Critical values of ε occur when $\triangleright$ changes. Due to Lemma 1, all critical values involve an edge or an xt-face or yt-face of a cell, but never the internal volume, so the following critical values cover all cases.

a) The minimal ε such that $(0, 0, t) \in \mathcal{F}_\varepsilon^{3\mathrm{D}}$ and $(P, P, t) \in \mathcal{F}_\varepsilon^{3\mathrm{D}}$ for all t.
b) An edge of $C_{x,y,t}$ becomes nonempty.
c) Endpoints of y-edges of $C_{x,y,t}$ and $C_{x+i,y,t}$ align in y-coordinate, or endpoints of x-edges of $C_{x,y,t}$ and $C_{x,y-j,t}$ align in x-coordinate.
d) Endpoints of a t-edge of $C_{x,y,t}$ and a t-edge of $C_{x+i,y-j,t}$ align in t-coordinate.
e) An obstacle in $C_{x,y,t}$ stops overlapping with an obstacle in $C_{x+i,y,t}$ or $C_{x,y-j,t}$ when projected along the x- or y-axis.

The endpoints involved in the critical values of type a), b), c) and d) can be captured in $O(P^2T)$ functions. We apply a parametric search for the minimum critical value $\varepsilon_{\mathrm{abcd}}$ of type a), b), c) or d) for which a matching exists. This takes $O((P^2T + time_{\mathrm{dec}}) \log(PT))$ time.

We illustrate the need for critical values of type e) in Figure 6, here obstacle a overlaps with both obstacles b and c while the overlap in edges does not contribute to $\triangleright$. It is unclear how critical values of type e) can be incorporated in the parametric search directly. Instead, we enumerate and sort the $O(P^3T)$ critical values of type e) in $O(P^3T\log(PT))$ time. Using $O(\log(PT))$ calls to the decision algorithm, we apply a binary search to find the minimum critical value ε_{e} of type e) for which a matching exists. Finding the critical value ε_{e} then takes $O((P^3T + time_{\mathrm{dec}})\log(PT))$ time.

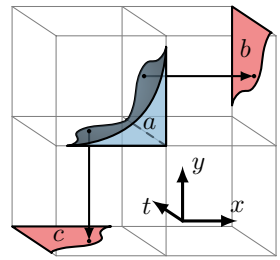


Fig. 6. $a \triangleright b$ and $a \triangleright c$

The synchronous dynamic Fréchet distance is then the minimum of $\varepsilon_{\mathrm{abcd}}$ and ε_{e}. This results in the following running time.

Theorem 5. *The synchronous dynamic Fréchet distance can be computed in* $O((P^3T + time_{dec})\log(PT))$ *time.*

Before stating the final running time, we present a faster algorithm for the decision algorithm.

2.4 A Faster Decision Algorithm

To speed up the decision procedure we distinguish the cases for which two obstacles may be related by $\triangleright$, these cases correspond to the five types of critical values of Section 2.3. Critical values of type a) and b) depend on obstacles in single cells, so there are at most $O(P^2T)$ elements of $\triangleright$ arising from type a) and b). Critical values of type c) and e) arise from pairs of obstacles in cells in the same row or column, so there are at most $O(P^3T)$ of them. In fact, we can enumerate the edges of type a), b), c), and e) of $\triangleright$ in $O(P^3T)$ time. On the other hand, edges of type d) arise between two cells with the same value of t, so there can be $O(P^4T)$ of them.

We compute a smaller directed graph (V, E) with $|E| = O(P^3T)$ that has a path from d to u if and only if $d \trianglerighteq u$. Let $V = O = \{u, d\} \cup O'$ be the vertices as before (we will include dummy obstacles for grid points in that lie in the freespace) and transfer the edges in $\triangleright$ except those of type d) to the smaller set of edges E. We must still induce edges of type d) in E, but instead of adding $O(P^4T)$ edges, we use only $O(P^3T)$ edges. The edges of type d) can actually be captured in the transitive closure of E using only $O(P)$ edges per obstacle in E.

Using an edge from $o_{x,y,t}$ to $o_{x+1,y,t}$ and to $o_{x,y-1,t}$, we construct a path from $o_{x,y,t}$ to any obstacle $o_{x+i,y-j,t}$. The sole purpose of the dummy obstacles is to construct these paths effectively. For obstacles whose gridpoints have the same t-coordinates, it then takes a total of $O(P^2T)$ edges to include the obstacles overlapping in t-coordinate related by type d), this is valid because $(x, y, t) \in o_{x,y,t}$ for non-dummy obstacles.

Denote by E_k^{d} the edges of type d) of the form $(a, b) = (o_{x,y,t_a}, o_{x+i,y-j,t_b})$ where $t_b = t_a + k$, then the set E_0^{d} of $O(P^2T)$ edges is the one we just constructed. Now it remains to induce paths with $t_a \neq t_b$, that still overlap in t-coordinates,

i.e. the sets E^{d}_{-2}, E^{d}_{-1}, E^{d}_{1} and E^{d}_{2}. Denote by $t^-(a)$ and $t^+(a)$ the minimum and maximum t-coordinate over points in an obstacle a. For each obstacle, both the $t^-(a)$ and the $t^+(a)$ coordinates are an endpoint of a t-edge in a cell defining the obstacle due to Lemma 1, and therefore computable in constant time.

Our savings arise from the fact that if $o_{x,y,t} \triangleright o_{x+i,y-j,t+k}$ and $o_{x,y,t} \triangleright o_{x+i',y-j',t+k}$ with $i \leq i'$ and $j \leq j'$, then E^{d}_0 induces a path from $o_{x+i,y-j,t+k}$ to $o_{x+i',y-j',t+k}$, so we do not need an additional edge to induce a path to the latter obstacle. To avoid degenerate cases, we start by exhaustively enumerating edges of E^{d}_k ($k \in \{-2,-1,1,2\}$) for which $i \leq 1$ or $j \leq 1$ in $O(P^3T)$ time so we need only consider edges with $i \geq 2 \wedge j \geq 2$.

For these remaining cases, we have $a \triangleright b$ if and only if $t^+(a) \geq t^-(b) \wedge t_b = t_a + k$, and $t^-(a) \leq t^+(b) \wedge t_b = t_a - k$ for positive k. From this we can derive the edges of E^{d}_k. Although for each a, there may be $O(P^2)$ obstacles b such that $a \triangleright b$ with $t_b = t_a + k$, the Pareto frontier of those obstacles b contains only $O(P)$ obstacles, see the grid of fictional values $t^-(b)$ in Figure 7. In the full paper, we show how to find these Pareto frontiers in $O(P \log P)$ time per obstacle a, using only $O(P^2T)$ preprocessing time for the complete freespace.

As a result, we can compute all $O(P^3T)$ edges of E^{d}_k in $O(P^3T \log P)$ time. By Theorem 6, the decision problem for the synchronous dynamic Fréchet distance is solvable in $O(P^3T \log P)$ time.

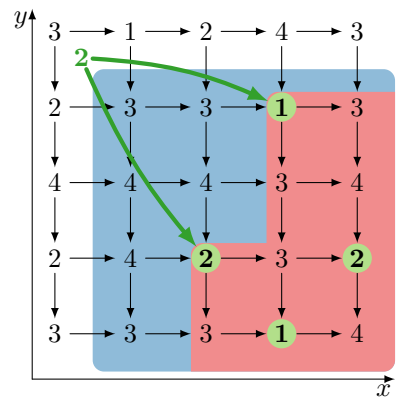


Fig. 7. Two edges (green) cover (red) all four obstacles b (green) within the query rectangle (blue) with values $t^-(b) \leq t^+(a) = 2$.

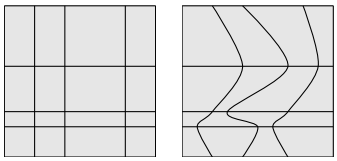
Theorem 6. *The decision problem for the synchronous dynamic Fréchet distance is solvable in $O(P^3T \log P)$ time.*

Proof. The edges E of types other than d) are enumerated in $O(P^3T)$ time, and using constantly many Pareto frontier queries for each obstacle, $O(P^3T)$ edges of type d) in E are computed in $O(P^3T \log P)$ time. Given the set E of edges, deciding whether a path between two vertices exists takes $O(|E|) = O(P^3T)$ time. The transitive closure of E equals $\trianglerighteq$, so a path from d to u exists in E if and only if there was such a path in $\triangleright$. Since we compute E in $O(P^3T \log P)$ time, the decision problem is solved in $O(P^3T \log P)$ time. □

The following immediately follows from Theorems 5 and 6.

Corollary 7. *The synchronous dynamic Fréchet distance can be computed in $O(P^3T \log P \log(PT))$ time.*

3 Hardness



We extend the synchronous constant and synchronous dynamic classes of matchings to asynchronous ones. For this, we allow realignments of timestamps, giving rise to the asynchronous constant and asynchronous dynamic classes of matchings. The asynchronous constant class ranges over matchings of the form $\mu(p,t) = (\pi(p), \tau(t))$ where the π and τ are matchings of positions and timestamps. The asynchronous dynamic class of matchings has the form $\mu(p,t) = (\pi_t(p), \tau(t))$ for which the positional matching π_t changes over time. We first prove that the asynchronous constant Fréchet distance is in NP.

Theorem 8. *Computing the Fréchet distance is in NP for the asynchronous constant class of matchings.*

Proof. Given any matching $\mu(p,t) = (\pi(p), \tau(t))$ with a Fréchet distance of ε, we can derive—due to Lemma 1—a piecewise-linear matching τ^* in $O(T)$ time, such that a matching $\mu^*(p,t) = (\pi^*(p), \tau^*(t))$ with Fréchet distance at most ε exists. We can realign the quadrilateral meshes A and B under τ^* to obtain meshes A^* and B^* of polynomial size. Now the polynomial-time decision algorithm for synchronous constant matchings (see full paper) is applicable to A^* and B^*. $\square$

Due to critical values of type e), it is unclear whether each asynchronous dynamic matching admits a piecewise-linear matching τ^* of polynomial size, which would mean that the asynchronous dynamic Fréchet distance is also in NP.

We show that computing the Fréchet distance is NP-hard for both classes by a reduction from 3-SAT. The idea behind the construction is illustrated in the two height maps of Figure 8. These represent quadrilateral meshes embedded in $\mathbb{R}^1$ and correspond to a single clause of a 3-CNF formula of four variables.

We distinguish valleys (dark), peaks (white on A, yellow on B) and ridges (denoted X_i, F_i and T_i). An important observation is that in order to obtain a low Fréchet distance of $\varepsilon < 3$, the n-th valley of A must be matched with the n-th valley of B. Moreover, each ridge X_i must be matched with F_i or T_i and each peak of A must be matched to a peak of B. Note that even for asynchronous

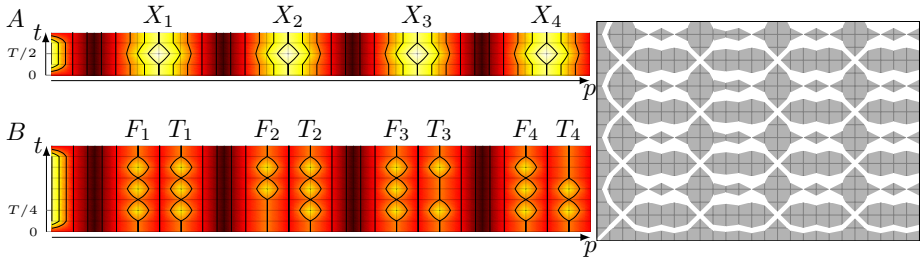


Fig. 8. Two quadrilateral meshes A and B embedded in $\mathbb{R}^1$ (indicated by color and isolines). Their Fréchet distance is 2 isolines if the clause $(X_2 \vee \neg X_3 \vee \neg X_4)$ is satisfiable and 3 isolines otherwise. The freespace $\mathcal{F}_{\varepsilon=2}$ of (A, B) at times $(T/2, T/4)$ on the right.

dynamic matchings, if X_i is matched to F_i, it cannot be matched to T_i and vice-versa because the (red) valley separating F_i and T_i has distance 3 from X_i.

The aforementioned properties are reflected more clearly in the 2D freespace between the curves at aligned timestamps t and $\tau(t)$. In Figure 8, we give a 2D slice (at $t_A = T/2$, $t_B = T/4$) of the 4D-freespace diagram with $\varepsilon = 2$ for the shown quadrilateral meshes. In this diagram with $\varepsilon = 2$, only 2^3 monotone paths exist (up to directed homotopy) whereas for $\varepsilon = 3$ there would be 2^4 monotone paths (one for each assignment of variables). For $\varepsilon = 2$, the peak of X_2 cannot be matched to F_2 at $t = T/4$ of B, corresponding to an assignment of $X_2 = \mathit{true}$.

Consider a 3-CNF formula with n variables and m clauses, then A and B consist of m clauses along the t-axis and n variables ($X_1 \dots X_n$ and $F_1, T_1 \dots F_n, T_n$) along the p-axis. The k-th clause of A is matched to the k-th clause of B due to the elevation pattern on the far left ($p = 0$). This means that the peaks of A are matched with peaks of the same clause on B and all these peaks have the same timestamp because $\tau(t)$ is constant (independent of p).

For each clause, there are three rows (timestamps) of B with peaks on the ridges. On each such timestamp, exactly one ridge (depending on the disjuncts of the clause) does not have a peak. Specifically, if a clause has X_i or $\neg X_i$ as its k-th disjunct, then the k-th row of that clause has no peak on ridge F_i or T_i, respectively. We use these properties in Theorem 11 where we prove that it is NP-hard to approximate the Fréchet distance within a factor 1.5.

Lemma 9. *The Fréchet distance between two such moving curves is at least* 3 *if the corresponding 3-CNF formula is unsatisfiable.*

Proof. Consider a matching yielding a Fréchet distance smaller than 3 given an unsatisfiable formula, then the peaks of A (of the k-th clause) are matched with peaks of B (of a single row of the k-th clause). Assign the value *true* to variable X_i if ridge X_i is matched with T_i and *false* if it is matched with F_i. Then for every clause $(V_i \vee V_j \vee V_k)$ with $V_i \in \{X_i, \neg X_i\}$, there is a peak at $\pi(X_i)$, $\pi(X_j)$ or $\pi(X_k)$ for that clause. Such a matching cannot exist because then the 3-CNF formula would be satisfiable, so the Fréchet distance is at least 3. □

Lemma 10. *The Fréchet distance between two such moving curves is at most* 2 *if the corresponding 3-CNF formula is satisfiable.*

Proof. Consider a satisfying assignment to the 3-CNF formula. Match X_i with the center of F_i or T_i, if X_i is *false* or *true*, respectively. For every clause, the timestamp with peaks of A can be matched with a row of peaks on B. As was already hinted at by Figure 8, the remaining parts of the curves can be matched with $\varepsilon = 2$. Therefore this yields a Fréchet distance of at most 2. □

Theorem 11. *It is NP-hard to approximate the asynchronous constant or asynchronous dynamic Fréchet distance for moving curves in* $\mathbb{R}^1$ *within a factor* 1.5.

Proof. By Lemmas 9 and 10, the asynchronous constant or asynchronous dynamic Fréchet distance between two quadrilateral meshes embedded in $\mathbb{R}^1$ is at least 3 or at most 2, depending on whether a 3-CNF formula is satisfiable. □

Acknowledgements. K. Buchin, T. Ophelders, and B. Speckmann are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 612.001.207 (K. Buchin) and no. 639.023.208 (T. Ophelders & B. Speckmann).

References

1. H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete Comput. Geom.*, 43(1):78–99, 2010.
2. H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. In *Proc. 14th Sympos. Discrete Algorithms*, pages 589–598, 2003.
3. H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Intern. J. Comput. Geom. Appl.*, 5(01n02):75–91, 1995.
4. B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet Distances for Curves, Revisited. In *Proc. 14th European Sympos. Algorithms*, pages 52–63, 2006.
5. S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proc. 31st Intern. Conf. VLDB*, pages 853–864, 2005.
6. K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Foundations of Computer Science, 2014*, pages 661–670, 2014.
7. K. Buchin, M. Buchin, and J. Gudmundsson. Constrained free space diagrams: a tool for trajectory analysis. *Intern. J. GIS*, 24:1101–1125, 2010.
8. K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Intern. J. Comput. Geom. Appl.*, 21(3):253–282, 2011.
9. K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog-with an application to Alt's conjecture. In *Proc. 25th Sympos. Discrete Algorithms*, pages 1399–1413, 2014.
10. K. Buchin, M. Buchin, and A. Schulz. Fréchet distance of surfaces: Some simple hard cases. In *Proc. 18th European Sympos. Algorithms*, pages 63–74, 2010.
11. K. Buchin, M. Buchin, and C. Wenk. Computing the Fréchet distance between simple polygons. *Comput. Geom. Theory Appl.*, 41(1–2):2–20, 2008.
12. M. Buchin. *On the Computability of the Fréchet Distance Between Triangulated Surfaces.* PhD thesis, Free University Berlin, Institute of Computer Science, 2007.
13. M. Buchin, S. Dodge, and B. Speckmann. Context-aware similarity of trajectories. In *Proc. 6th Intern. Conf. GIS*, pages 43–56, 2012.
14. A. F. Cook IV, A. Driemel, S. Har-Peled, J. Sherette, and C. Wenk. Computing the Fréchet distance between folded polygons. In *Algorithms and Data Structures*, pages 267–278, 2011.
15. A. Driemel, S. Har-Peled, and C. Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete Comput. Geom.*, 48(1):94–127, 2012.
16. M. Godau. *On the complexity of measuring the similarity between geometric objects in higher dimensions.* PhD thesis, Berlin, Freie Universität Berlin, Diss., 1998, 1999.
17. S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms*, 10(1):3, 2014.
18. A. Maheshwari, J.-R. Sack, K. Shahbaz, and H. Zarrabi-Zadeh. Fréchet distance with speed limits. *Comput. Geom. Theory Appl.*, 44(2):110–120, 2011.