# Computing the similarity between moving curves

# Computing the Similarity Between Moving Curves

Kevin Buchin[a], Tim Ophelders[a], Bettina Speckmann[a]

[a]*Department of Mathematics and Computer Science, TU 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 retreating 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 generally NP-hard, 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 a relation between obstacles.

## 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 objects 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 $X \to \mathbb{R}^n$.

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

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* $\mu$ 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 $\mu$.

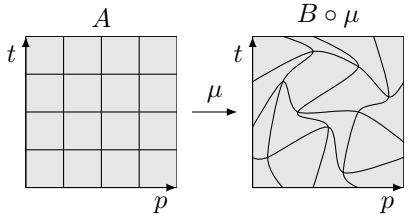


Figure 1: A matching $\mu$ between surfaces $A$ and $B$ drawn as a homeomorphism between their parameter spaces.

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, 14]. 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, 19], and approximation algorithms [4, 17] 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. Computing the Fréchet distance between surfaces is NP-hard [18], even for terrains [10]. For general surfaces the Fréchet distance was only known to be semi-computable [1, 13]. Very recently twofold progress was made: the first computability results were found for surfaces of genus 0 [22] and the problem

of computing the Fréchet distance between spheres in $\mathbb{R}^1$ was shown to be in NP [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 [16].

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, 20]. 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.

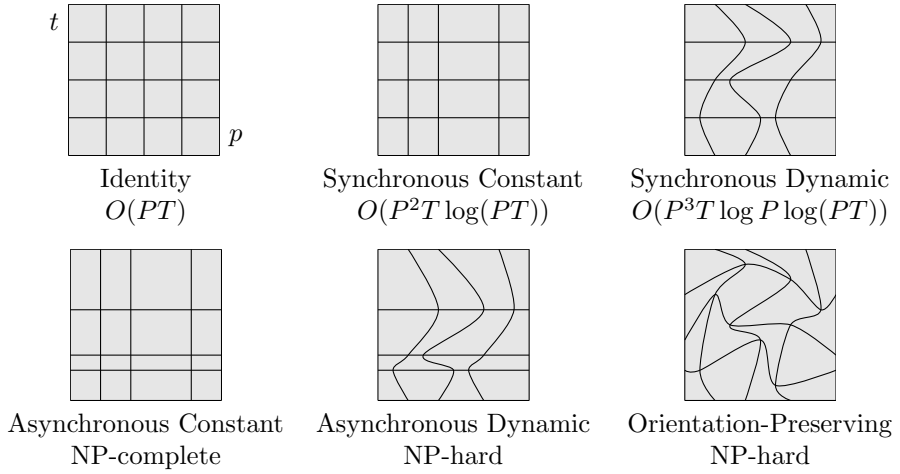



Figure 2: The time complexities of the considered classes of matchings.

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 Section 2.

We discuss the *synchronous constant* Fréchet distance in Section 3. 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 minimizing the Fréchet distance.

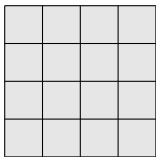
The *synchronous dynamic* Fréchet distance considered in Section 4 also assumes a predefined matching of timestamps, but does not have the constraint of the *synchronous constant* class that the matching of positions remains constant over time. Instead, the positional matching may change continuously over time.

Finally, in Section 5, 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, matchings range over orientation preserving homeomorphisms between parameter spaces, given that the corners of the parameter spaces 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 even under these very restricted classes of matchings. 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. In fact, we refer to time and position parameters solely to distinguish the two dimensions; in practice, they can represent any two parameters that constitute a surface. 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 or data sampled at a fixed rate. The synchronous dynamic Fréchet distance is better suited for sensors with different sampling frequencies, placed on curve-like moving objects.

## 2. Identity Matchings



Suppose we are given a single predefined matching $\mu$ between the moving curves $A$ and $B : [0, P] \times [0, T] \to \mathbb{R}^n$. We can compute the Fréchet distance under this matching if we can find the corresponding points of $A$ and $B$ that are furthest apart, see Equation 2.

$$D_\mu(A, B) = \sup_{x \in [0,P] \times [0,T]} \|A(x) - B(\mu(x))\| \tag{2}$$

For quadrilateral meshes $A$ and $B$ of size $P \times T$, the identity matching $\mu(p, t) = (p, t)$ allows us to simplify $A(x) - B(\mu(x))$ into $C(x) = A(x) - B(x)$ where $C$ is again a quadrilateral mesh of size $P \times T$. The Fréchet distance for the identity matching depends only on the point on $C$ that is furthest from the origin. To see this, consider a single quadrilateral; the point furthest from the origin must be one of its four corners since all points on the quadrilateral lie within the convex hull of its four corner points. Hence, it suffices to check only the distance to the origin for the $O(PT)$ vertices of $C$, see Equation 3. The Fréchet distance under the identity matching can then be computed in $O(PT)$ time.

$$D_{id}(A,B) = \sup_{x\in[0,P]\times[0,T]} \| \underbrace{A(x) - B(x)}_{C(x)} \| = \sup_{x\in\{0,\dots,P\}\times\{0,\dots,T\}} \|C(x)\| \qquad (3)$$

Meshes of different size can be compared after introducing $O(PT)$ dummy vertices, such that the meshes have equal dimensions and each vertex has an aligned vertex on the other mesh. For this, it is important to note that any quadrilateral can be subdivided into an equivalent mesh of four quadrilaterals with any point on the original quadrilateral as their shared corner.

To extend this further, consider the case where the matching $\mu(p,t) = (\pi(p), \tau(t))$ is defined by two piecewise-linear functions $\pi$ and $\tau$ of $|\pi|$ and $|\tau|$ vertices. This allows comparing moving curves under predefined realigments of timestamps as well as positions. For such a matching, the surface $C(x) = A(x) - B(\mu(x))$ is a quadrilateral mesh of $O((P+|\pi|)(T+|\tau|))$ vertices. We illustrate this in Figure 3 for $t = 4$ and $|\tau| = 3$ with vertices $(\tau(0), \tau(1.5), \tau(4)) = (0, 2.5, 4)$. In such case $O(T+|\tau|)$ timestamps of $A$ are matched with $O(T+|\tau|)$ timestamps of $B$.

The same can be done given a piecewise-linear reparameterization of positions. As a result, the Fréchet distance under a predefined piecewise-linear reparameterization of timestamps and positions can be computed in $O((P + |\pi|)(T + |\tau|))$ time.

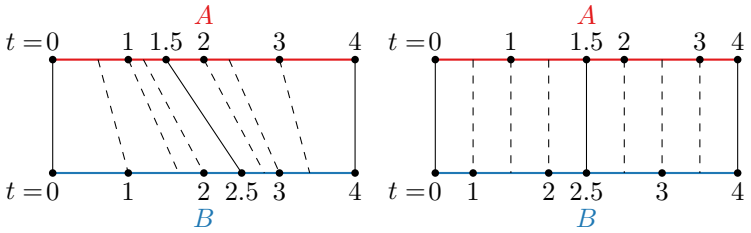
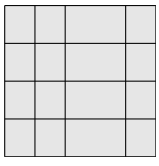



Figure 3: A piecewise linear matching of timestamps. Left, the matching drawn with the original time axes. Right, the matching drawn after warping the time axes.

## 3. Synchronous Constant Matchings



In this section, we consider the class of *synchronous constant matchings* where the matching $\mu(p,t) = (\pi(p), t)$ assumes no realignments of time and a constant reparameterization $\pi$ of positions. Thus, the goal is to find a continuous nondecreasing surjection $\pi$ on $[0, P]$, such that the Fréchet distance is minimized.

Before we present the algorithm to find synchronous constant matchings, we refer to an existing algorithm [3] that computes the Fréchet distance between static curves. This classic algorithm makes use of a data structure called the

$\varepsilon$-freespace diagram, which is the set of parameter pairs for which the represented points are at most $\varepsilon$ apart. A freespace diagram for two static curves is illustrated in Figure 4.

The matching $\pi(x) = y$ between the parameter spaces of static curves $A$ and $B$ can be embedded as a bimonotone path $\{(x, y) \mid \pi(x) = y\}$ in the freespace. Therefore, the Fréchet distance is at most $\varepsilon$ if and only if there exists a bimonotone path from the bottom-left to the top-right corner of the $\varepsilon$-freespace. Using freespace diagrams, the decision problem for static curves can be solved in $O(P^2)$ time.

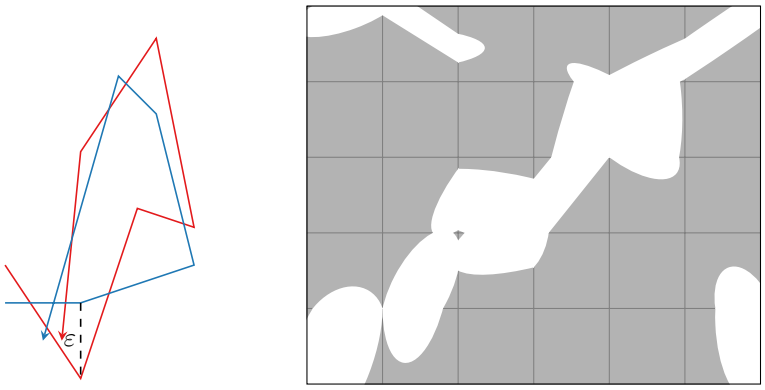



Figure 4: Left: Curves $A$ (red) and $B$ (blue) with Fréchet distance $\varepsilon$. Right: In white, their freespace diagram $\mathcal{F}_\varepsilon = \{(x, y) \in [0, 6] \times [0, 5] \mid \|A(x) - B(y)\| \leq \varepsilon\}$. If we draw the freespace diagram for any smaller value of $\varepsilon$, no bimonotone path through the freespace connects the bottom-left corner to the top-right corner.

$$\mathcal{F}_\varepsilon = \{(x, y, t) \in [0, P] \times [0, P] \times [0, T] \mid \|A(x, t) - B(y, t)\| \leq \varepsilon\} \tag{4}$$

We extend this approach to compute matchings between moving curves. For this, consider the 3D freespace defined by Equation 4. Since any synchronous constant matching $\mu(p, t) = (\pi(p), t)$ is defined by a bimonotone path $\pi : [0, P] \to [0, P]$, the matching $\mu$ is embedded in the freespace diagram as the surface $\mu = \{(x, y) \mid \pi(x) = y\} \times [0, T]$. Such a matching yields a Fréchet distance of at most $\varepsilon$ if and only if $\mu \subseteq \mathcal{F}_\varepsilon$. The 3D freespace consists of $O(P^2T)$ cells $C_{x,y,t} = \mathcal{F}_\varepsilon \cap ([x, x + 1] \times [y, y + 1] \times [t, t + 1])$ for $(x, y, t) \in \mathbb{N}^3$. We can simplify the three-dimensional freespace ($\mathcal{F}_\varepsilon$) into a two-dimensional one ($\mathcal{F}_\varepsilon^{\text{2D}}$) with

$$\mathcal{F}_\varepsilon^{\text{2D}} = \{(x, y) \in [0, P] \times [0, P] \mid \text{for all } t \in [0, T], \|A(x, t) - B(y, t)\| \leq \varepsilon\} \tag{5}$$

and find the path $\pi$ defining $\mu$ in it. To simplify Equation 5 we prove the following lemma.

**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.*

*Proof.* Each cell in the freespace is the freespace induced by a pair of quadrilaterals, so consider two quadrilaterals $A(x,t)$ and $B(y,t)$. These quadrilaterals are bilinear interpolations between four corner points. Hence, $A(x,t) - B(y,t)$ is an affine map for each fixed $t$. Likewise, $A(x,t) - B(y,t)$ is an affine map for each fixed pair $(x,y)$. Because the preimage of a convex norm ball under an affine map is convex, the intersection of a freespace cell with lines parallel to the $t$-axis or the $xy$-plane forms a convex set. □

**Lemma 2.** $\mathcal{F}_\varepsilon^{2D} = \{(x,y) \in [0,P] \times [0,P] \mid \textit{for all } t \in \{0,\ldots,T\} \ \|A(x,t) - B(y,t)\| \leq \varepsilon\}$.

*Proof.* By Lemma 1, the intersection of a cell of $\mathcal{F}_\varepsilon$ with a line parallel to the $t$-axis is convex. Hence, if a point $(x,y,t) \notin \mathcal{F}_\varepsilon$, then either $(x,y,\lfloor t \rfloor) \notin \mathcal{F}_\varepsilon$ or $(x,y,\lceil t \rceil) \notin \mathcal{F}_\varepsilon$. Therefore the internal structure of freespace cells can safely be ignored. □

We denote the $O(P^2)$ cells of the 2D freespace by $C_{x,y} = ([x,x+1] \times [y,y+1]) \cap \mathcal{F}_\varepsilon^{2\mathrm{D}}$ where $(x,y) \in \mathbb{N}^2$.

**Lemma 3.** *Every cell $C_{x,y}$ in the 2D freespace $\mathcal{F}_\varepsilon^{2D}$ is convex.*

*Proof.* By Lemma 1, $C_{x,y}$ is an intersection of convex sets, so $C_{x,y}$ is convex. □

Figure 5 illustrates (in white) what the 2D freespace $\mathcal{F}_\varepsilon^{2\mathrm{D}}$ might look like for two moving curves. As before, any smaller value of $\varepsilon$ would disconnect the bottom-left from the top-right corner. For a given $\varepsilon$, for each of the $O(P^2)$ cells, the boundary intervals can be computed in $O(T)$ time. Therefore, finding an $xy$-monotone path takes $O(P^2T)$ time, solving the decision problem for the Fréchet distance under synchronous constant matchings in $O(P^2T)$ time.

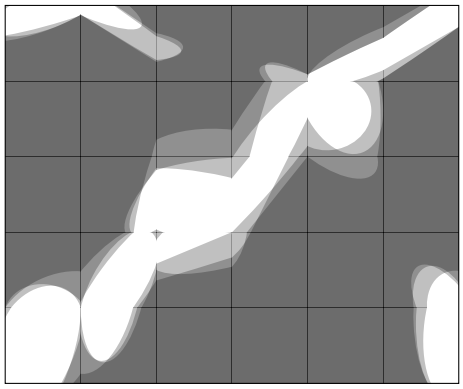


Figure 5: In white, $\mathcal{F}_\varepsilon^{2\mathrm{D}}$ for two moving curves. Darker shaded areas are in fewer layers of $\mathcal{F}_\varepsilon$.

### *3.1. Parametric Search*

We use the decision problem in a parametric search for the minimum $\varepsilon$ admitting a matching. When increasing $\varepsilon$ starting from $\varepsilon = 0$, there are three types of critical values of $\varepsilon$ for which a passage might open in the freespace. Due to Lemma 1, these critical values correspond exactly to those of the known algorithm [3] for computing the Fréchet distance between two curves:

a) The minimal $\varepsilon$ with $(0,0) \in \mathcal{F}_\varepsilon^{2\mathrm{D}}$ and $(P,P) \in \mathcal{F}_\varepsilon^{2\mathrm{D}}$.
b) One of the four boundaries of a cell $C_{x,y}$ becomes nonempty.
c) The lower (or left) endpoint on the boundary of cell $C_{x,y}$ aligns with the upper (or right) endpoint on the boundary of $C_{x+i,y}$ (or $C_{x,y+j}$).

Note that all critical values occur when two endpoints align (or an endpoint aligns with a gridpoint). For each of the $O(T)$ layers defining $\mathcal{F}_\varepsilon^{2\mathrm{D}}$, we use a function of $\varepsilon$ to represent the $x$- or $y$-position of such an endpoint (or gridpoint). This amounts to a total of $O(P^2T)$ functions, and each critical value occurs when two of them intersect. Since the intersections between any pair of these functions can be computed in constant time, we can apply a parametric search [21].

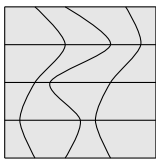
We use Cole's variant [15] of the parametric search to find the desired critical value in $O((k+time_{\mathrm{dec}})\log k)$ time. Here $k = O(P^2T)$ is the number of functions to which the parametric search is applied and $time_{\mathrm{dec}} = O(P^2T)$ is the running time for the decision problem. We obtain the running time of Theorem 4.

**Theorem 4.** *The synchronous constant Fréchet distance between quadrilateral meshes can be computed in $O(P^2T\log(PT))$ time.*

**Remark 1.** To compare quadrilateral meshes under a piecewise linear realignment of timestamps, we can subdivide the quadrilateral meshes as explained in Section 2. Although for simplicity we have assumed that the meshes are of equal size, when computing the Fréchet distance between meshes of different sizes, $P \times T$ and $Q \times T$, the 2D freespace has only $O(PQ)$ cells. Thus the decision problem is solved in $O(PQT)$ time and the exact computation takes $O(PQT\log(PQT))$ time.

**Remark 2.** If the inputs of the algorithm are two sequences of $O(T)$ curves without predefined interpolations to obtain quadrilateral meshes, we can still measure their Fréchet distance for the optimal (but unknown) linear interpolation. Due to the convexity of freespace cells, the Fréchet distance is the minimum Fréchet distance between two curves $A_t$ and $B_t$, which can be computed in $O(TP^2\log P)$ time by running the original algorithm $O(T)$ times.

## 4. 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.

### *4.1. Freespace Partitions in 2D*

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

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

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 $\varepsilon$ for which an $xy$-monotone path (representing $\mu$) from $(0,0)$ to $(P,P)$ through the freespace exists.

We use a relation between *obstacles* 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 6. 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.

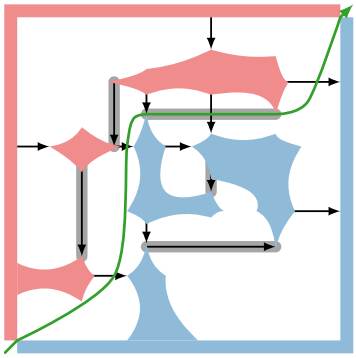


Figure 6: A matching (green) in the 2D freespace (white).

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, as proven in a generalized setting in Lemma 6. The figure shows a few such edges in black. If all obstacles were slightly larger, an edge could connect the lower-right (blue) boundary blue obstacle to the upper-left (red) boundary obstacle, forming the path connecting the boundaries highlighted in gray.

### *4.2. 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 $\mathcal{F}_\varepsilon$ (see Equation 6).

$$(x,y,t) \in \mathcal{F}_\varepsilon \text{ if and only if } \|A(x,t) - B(y,t)\| \leq \varepsilon \tag{6}$$

Such a surface again divides the obstacles in the freespace into 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.

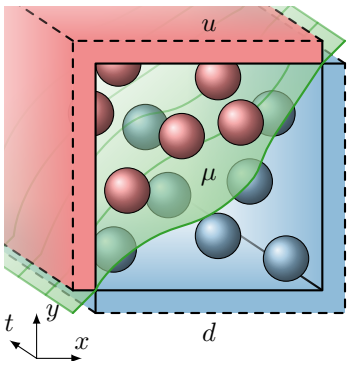



Figure 7: $\mu$ separates $u$ and $d$.

For $x, y, t \in \mathbb{N}$, the cell $C_{x,y,t}$ of the 3D freespace is the set $\mathcal{F}_\varepsilon \cap ([x, x+1] \times [y, y+1] \times [t, t+1])$. The property of Lemma 1 holds for all such cells. We divide the set of points not in $\mathcal{F}_\varepsilon$ into a set $\Gamma$ 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 $\mathcal{F}_\varepsilon$. Symmetrically, let $d$ represent the bottom and right boundary, see Figure 7. Denote by $\Gamma' \subset \Gamma$ the obstacles between the boundaries.

$\Gamma = \{u, d\} \cup \Gamma'$ with $\bigcup \Gamma' = ([0, P]^2 \times [0, T]) \setminus \mathcal{F}_\varepsilon$;

$u = \{(x, y, t) \mid (x < 0 \text{ and } y > 0) \text{ or } (x < P \text{ and } y > P)\}$;

$d = \{(x, y, t) \mid (x > 0 \text{ and } y < 0) \text{ or } (x > P \text{ and } y < P)\}$.

For a given matching $\mu$, let $D_\mu \subseteq \Gamma$ be the set of obstacles below it, then $u \notin D_\mu$ and $d \in D_\mu$. 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 $\mu$ cannot intersect obstacles, a single obstacle cannot lie on both sides of the same matching. Because all matchings have $u \notin D_\mu$ and $d \in D_\mu$, a matching exists if and only if $\neg(d \in D_\mu \Rightarrow u \in D_\mu)$.

We compute a relation $\triangleright$ of elementary dependencies between obstacles, such that its transitive closure $\circledcirc$ has $d \circledcirc u$ if and only if $d \in D_\mu \Rightarrow u \in D_\mu$. 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 5 and 6 that this choice of $\triangleright$ satisfies the required properties and in Theorem 7 that we can use the transitive closure $\circledcirc$ of $\triangleright$ to solve the decision problem of the Fréchet distance.

**Lemma 5.** *If $a \circledcirc b$, then $a \in D_\mu \Rightarrow b \in D_\mu$ for every matching $\mu$.*

*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 $\mu$ with $a \in D_\mu$, then $(x_a, y_\mu, t) \in \mu$ for some $y_\mu > y_a$. Similarly, if $b \notin D_\mu$, 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 $\mu$ that we can pick $y'_\mu$ 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, $a \in D_\mu \Rightarrow b \in D_\mu$ whenever $a \triangleright b$ and therefore whenever $a \circledcirc b$. □

**Lemma 6.** *If $d \in D_\mu \Rightarrow u \in D_\mu$ for all $\mu$, then $d \circledcirc u$.*

*Proof.* Suppose $d \in D_\mu \Rightarrow u \in D_\mu$ but not $d \circledcirc u$. Then no matchings exist that separate $u$ from $d$, and no path from $d$ to $u$ exists in the directed graph $G = (\Gamma, \triangleright)$. Pick as $D$ the set of obstacles reachable from $d$ in $G$, then $D$ does not contain $u$. Let $\mu$ be the boundary of the *hull* of $D$, where the hull is the set of points below and to the right of points of $D$; that is, so $\mu$ is the boundary of $\{(x', y', t) \mid (x, y, t) \in d \in D, x' \geq x, y' \leq y\}$. Then $D_\mu \supseteq D$, and we show that $\mu$ is a matching that separates $u$ from $d$ (and does not intersect any obstacles). Let $\mu$ restricted to any timestamp $t$ is an $x$- and $y$-monotone path from $(0, 0, t)$ to $(P, P, t)$, so $u \notin D_\mu$. We show that $\mu$ does not *properly* intersect any obstacles of $\Gamma$ by showing it does not intersect any obstacle $o \in \Gamma \setminus D$. Indeed, otherwise some point $(x_o, y_o, t) \in o$ lies below $\mu$, and because $D$ satisfies $\triangleright$ it follows from the definition of $\triangleright$ that $o' \triangleright o$ for some $o' \in D$. Hence, such an obstacle $o$ cannot exist. So $\mu$ does not intersect any obstacles, contradicting $d \in D_\mu \Rightarrow u \in D_\mu$. □

**Theorem 7.** *The Fréchet distance is greater than $\varepsilon$ if and only if for this value of $\varepsilon$, we have $d \circledcirc u$.*

*Proof.* We have for every matching $\mu$ that $u \notin D_\mu$ and $d \in D_\mu$. Therefore it follows from Lemma 5 that no matching exists if $d \circledcirc u$ for $\varepsilon$. In that case, the Fréchet distance is greater than $\varepsilon$. Conversely, if $\neg(d \circledcirc u)$ there is a set $D_\mu$ satisfying $\circledcirc$ with $u \notin D_\mu$ and $d \in D_\mu$. In that case, a matching exists by Lemma 6, and the Fréchet distance is at most $\varepsilon$. □

Denote by $\overline{\mathcal{F}_\varepsilon}$ the complement $([0,P]^2 \times [0,T]) \setminus \mathcal{F}_\varepsilon$ of the freespace. We choose the set of obstacles $\Gamma'$ such that $\bigcup \Gamma' = \overline{\mathcal{F}_\varepsilon}$ and the relation $\triangleright$ is easily computable. Note that due to Lemma 1, each connected component of $\overline{\mathcal{F}_\varepsilon}$ contains a corner of a cell, so any cell in the freespace contains constantly many (up to eight) components of $\bigcup \Gamma'$. We use grid points $(x,y,t) \in \mathbb{N}^3$ to index the obstacles $o_{x,y,t}$ of $\Gamma' = \bigcup_{(x,y,t)} \{o_{x,y,t}\}$. If $(x,y,t) \in \overline{\mathcal{F}_\varepsilon}$, define obstacle $o_{x,y,t}$ to be the connected component of $([x-1,x+1] \times [y-1,y+1] \times [t-1,t+1]) \cap \overline{\mathcal{F}_\varepsilon}$ containing $(x,y,t)$; that is, of the restriction of $\overline{\mathcal{F}_\varepsilon}$ to the (up to eight) cells adjacent to $(x,y,t)$. If $(x,y,t) \notin \overline{\mathcal{F}_\varepsilon}$, we define $o_{x,y,t}$ to be an empty (dummy) obstacle. Note that $\bigcup \Gamma' = \overline{\mathcal{F}_\varepsilon}$ and that obstacles need not be 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 $(a,b) \in \Gamma \times \Gamma$, 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 $d \circledcirc u$ is equivalent to testing whether a path from $d$ to $u$ exists in the directed graph $(\Gamma, \triangleright)$, 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 4.4 we show that a smaller set $E$ of size $O(P^3T)$ with the same transitive closure $\circledcirc$ is computable in $O(P^3T \log P)$ time, so the decision algorithm takes only $O(P^3T \log P)$ time.

### *4.3. 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 8. 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.

We are looking for the minimum value of $\varepsilon$ for which a matching exists. When increasing the value of $\varepsilon$, the relation $\triangleright$ becomes sparser since obstacles shrink. Critical values of $\varepsilon$ occur when $\triangleright$ changes. By Theorem 7, all critical values involve at most two obstacles. By Lemma 1, all critical

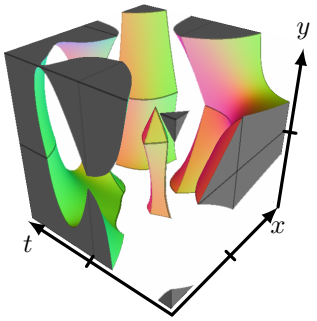


Figure 8: $[0,2]^3 \setminus \mathcal{F}_\varepsilon$

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 $\varepsilon$ such that $(0,0,t) \in \mathcal{F}_\varepsilon$ and $(P,P,t) \in \mathcal{F}_\varepsilon$ 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 orthogonally onto the $yt$- or $xt$-plane.

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 $\varepsilon_{\text{abcd}}$ of type a), b), c) or d) for which a matching exists. This takes $O((P^2T + \mathit{time}_{\text{dec}}) \log(PT))$ time.

We illustrate the need for critical values of type e) in Figure 9, 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 $\varepsilon_{\text{e}}$ of type e) for which a matching exists. Finding the critical value $\varepsilon_{\text{e}}$ then takes $O((P^3T + \mathit{time}_{\text{dec}}) \log(PT))$ time. The synchronous dynamic Fréchet distance is then the minimum of $\varepsilon_{\text{abcd}}$ and $\varepsilon_{\text{e}}$. This results in the following running time.

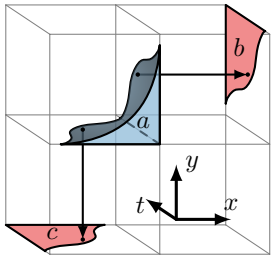


Figure 9: $a \triangleright b$ and $a \triangleright c$

**Theorem 8.** *The synchronous dynamic Fréchet distance can be computed in* $O((P^3T + \mathit{time}_{dec}) \log(PT))$ *time.*

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

### *4.4. 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 4.3. 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 $d \circledcirc u$. Let $V = \Gamma = \{u, d\} \cup \Gamma'$ be the vertices as before (we will include dummy obstacles for grid points 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 $E^{\mathrm{d}}_k$ 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 $E^{\mathrm{d}}_0$ 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 $E^{\mathrm{d}}_{-2}$, $E^{\mathrm{d}}_{-1}$, $E^{\mathrm{d}}_1$ and $E^{\mathrm{d}}_2$. 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 $E^{\mathrm{d}}_0$ induces a path from $o_{x+i,y-j,t+k}$ to $o_{x+i',y-j',t+k}$ 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 $j \leq j'$, 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 $E^{\mathrm{d}}_k$ ($k \in \{-2, -1, 1, 2\}$) 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 $E^{\mathrm{d}}_k$. 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 (see the definition below) of those obstacles $b$ contains only $O(P)$ obstacles, see the grid of fictional values $t^-(b)$ in Figure 10. Theorem 11 of Section 4.5 shows 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.

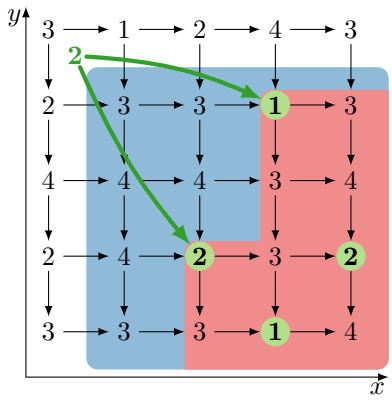



Figure 10: Two edges (green) cover (red) all four obstacles $b$ (green) within the query rectangle (blue) with values $t^-(b) \leq t^+(a) = 2$.

**Definition (Pareto frontier).** Given a partially ordered set $(\mathcal{S}, \succeq)$, the Pareto frontier of a subset $S \subseteq \mathcal{S}$ is the unique minimal subset $S' \subseteq S$ such that for all $s \in S$, there is some $s' \in S'$ for which $s' \succeq s$.

As a result, we can compute all $O(P^3T)$ edges of $E_k^{\mathrm{d}}$ in $O(P^3T \log P)$ time. By Theorem 9, the decision problem for the synchronous dynamic Fréchet distance is solvable in $O(P^3T \log P)$ time.

**Theorem 9.** *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 $\circledcirc$, 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 8 and 9.

**Corollary 10.** *The synchronous dynamic Fréchet distance can be computed in $O(P^3T \log P \log(PT))$ time.*

### *4.5. Pareto Frontier Queries*

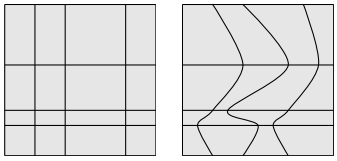
Suppose we are given a matrix $M$ with $m$ columns and $n$ rows, and we want to efficiently query submatrices for the Pareto frontier of numbers that are at most a given threshold value, $t$. A query specifies the threshold $t$, and two coordinates $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ of the query rectangle $R = \{x_{\min}, \ldots, x_{\max}\} \times \{y_{\min}, \ldots, y_{\max}\}$. Let $C_t(R) = \{(x, y) \in R \mid M[x, y] \leq t\}$ denote the coordinates of the query rectangle that must be dominated by the Pareto frontier $F_t(R) \subseteq C_t(R)$. That is, if $(x, y) \in C_t(R)$, then some $(x', y') \in F_t(R)$ with $x' \leq x \wedge y' \leq y$ exists.

We preprocess each of the $O(n)$ rows of $M$ in $O(m)$ time by storing their cells as the leaves of an augmented binary tree, whose internal nodes store the minimum value over its subtrees. Then queries for the index of the leftmost element with value at most $t$ in a range $\{x_{\min}, \ldots, x_{\max}\}$ of that row can be answered in $O(\log m)$ time. We can compute $F_t$ by including for each row, the element with minimum $x$-coordinate and value at most $t$ in the query range. So using $O(n)$ queries, we compute a set $F_t(R)$ of size $O(n)$ in $O(n \log m)$ time, using $O(nm)$ preprocessing time for the matrix $M$.

In this case, $F_t(R)$ is not actually the Pareto frontier since some of its elements might be dominated by other elements. With a slight modification, we can make $F_t(R)$ the actual Pareto frontier (of minimum size). If the query with range $\{x_{\min}, \ldots, x_{\max}\}$ returns $x^*$ with $x_{\min} \leq x^* \leq x_{\max}$ for some row, then the query range for subsequent rows can be restricted to $\{x_{\min}, \ldots, x^* - 1\}$, so no unnecessary values are generated.

**Theorem 11.** *For an $n \times m$ matrix, using $O(nm)$ preprocessing time, the Pareto frontier of values at least $t$ of any submatrix can be computed in $O(n \log m)$ time.*

## 5. Hardness



We extend the synchronous constant and synchronous dynamic classes of matchings (of Sections 3 and 4) 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 $\pi$ and $\tau$ 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 $\pi_t$ changes over time. We first prove that the asynchronous constant Fréchet distance is in NP.

**Theorem 12.** *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 $\varepsilon$, we can derive—due to Lemma 1—a piecewise-linear matching $\tau^*$ in $O(PT)$ time, such that a matching $\mu^*(p,t) = (\pi^*(p), \tau^*(t))$ with Fréchet distance at most $\varepsilon$ exists. We can realign the quadrilateral meshes $A$ and $B$ under $\tau^*$ to obtain meshes $A^*$ and $B^*$ of polynomial size. Now the polynomial-time decision algorithm for synchronous constant matchings is applicable to $A^*$ and $B^*$. □

Due to critical values of type e), it is unclear whether each asynchronous dynamic matching admits a piecewise-linear matching $\tau^*$ 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 11. These represent quadrilateral meshes embedded in $\mathbb{R}^1$ and correspond to a single clause of a 3-CNF formula of four variables. Both moving curves are zig-zags whose peaks change height slightly over time.

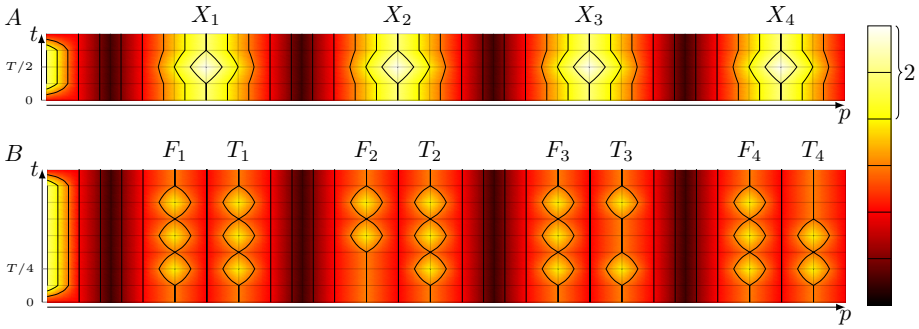



Figure 11: Two quadrilateral meshes $A$ and $B$ embedded in $\mathbb{R}^1$ (indicated by color and isolines). Their Fréchet distance is 2 isolines if the clause $(X_2 \vee \neg X_3 \vee \neg X_4)$ is satisfiable and 3 isolines otherwise.

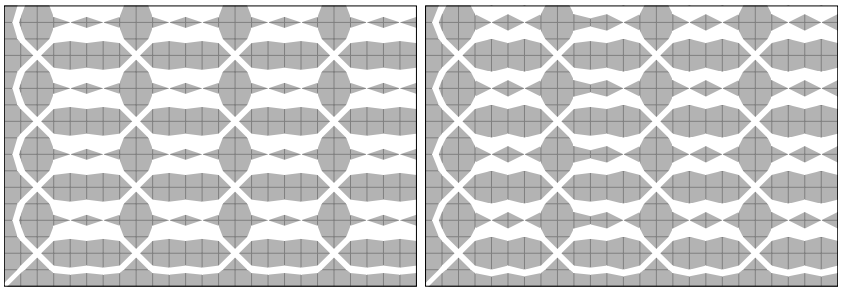


Figure 12: The freespace $\mathcal{F}_{\varepsilon=2}$ of $(A, B)$ at times $(0, 0)$ (left) and $(T/2, T/4)$ (right).

We distinguish dark valleys (height 0), peaks (white on $A$ (height 6), yellow on $B$ (height 4)) and ridges (denoted $X_i$, $F_i$ and $T_i$) with height 5 on $A$ and height 3 on $B$. In addition, $B$ has shallow valleys of height 2 separating each pair of ridges $F_i$ 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 dynamic matchings, if $X_i$ is matched to $F_i$, it cannot be matched to $T_i$ with $\varepsilon < 3$ 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 12, we give two 2D slices (with $(t_A, t_B) = (0, 0)$ and $(T/2, T/4)$, respectively) 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 $X_2 = \textit{true}$.

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 15 where we prove that it is NP-hard to approximate the Fréchet distance within a factor 1.5.

**Lemma 13.** *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 less 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. □

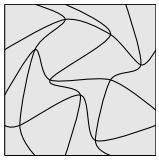
**Lemma 14.** *The Fréchet distance between two such moving curves is at most* 2 *if the corresponding 3-CNF formula is satisfiable.*

*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 11, the remaining parts of the curves can be matched with $\varepsilon = 2$. Therefore this yields a Fréchet distance of at most 2. □

**Theorem 15.** *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 13 and 14, 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. □

### *5.1. Orientation-Preserving Homeomorphisms*



Previous results [1, 10] have shown that computing the Fréchet distance between surfaces under orientation-preserving homeomorphisms is NP-hard for surfaces embedded in $\mathbb{R}^2$. We will refer to this variant as the *orientation-preserving* Fréchet distance. The prior results hold for triangular meshes, which are a degenerate case of quadrilateral meshes. We consider the case where the corners of the meshes are matched with each other.

The prior NP-hardness constructions for the orientation-preserving Fréchet distance seem unnecessarily complex. Therefore we extend our results for the asynchronous dynamic Fréchet distance of Section 5 to obtain a new hardness construction for the orientation-preserving Fréchet distance for surfaces embedded in $\mathbb{R}^2$. Note that we cannot directly apply the previous construction because there was only a single matching of timestamps for asynchronous dynamic matchings. Thus, we do not preserve the property that for a clause, the timestamp with peaks of $A$ maps to a single timestamp of $B$.

We prevent this by means of a second dimension in which we enforce that a row of peaks of $A$ maps to a single row of peaks of $B$. In addition to the embedding in $\mathbb{R}^1$ of Figure 11, which defines one coordinate for every point on a clause of a quadrilateral mesh, we define a second coordinate using Figure 13, yielding an embedding in $\mathbb{R}^2$.

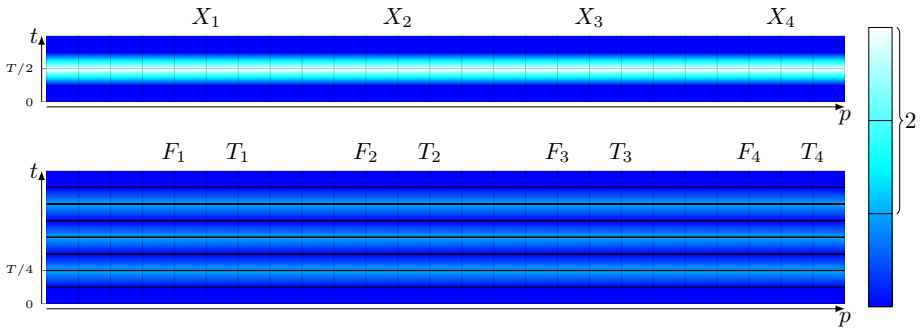



Figure 13: The second coordinate of a clause gadget of $A$ (top) and $B$ (bottom) embedded in $\mathbb{R}^2$, the first coordinate is given in Figure 11.

Now, under the maximum norm ($L^\infty$), a row of peaks of $A$ can only be matched to a single row of peaks on $B$ for $\varepsilon < 3$. Conversely, we can still match the meshes of two satisfiable formulas with $\varepsilon = 2$. Hence, Theorem 16 follows. This result extends to triangular meshes since all quadrilaterals lie in the plane, and can thus be represented by a pair of triangles. For norms other than the maximum norm, the problem is still NP-hard, but our bound on the approximation factor is smaller than 1.5.

**Theorem 16.** *Unless P=NP, no polynomial time algorithm can approximate the orientation-preserving Fréchet distance between two quadrilateral meshes embedded in* $\mathbb{R}^2$ *under the maximum norm within a factor* 1.5*.*

**Remark 3.** We recently were able to improve upon the results presented in this paper by showing that the Fréchet distance between surfaces is NP-hard to approximate within a factor 2 even for surfaces in $\mathbb{R}^1$ [12].

## 6. Conclusion

Based on the Fréchet distance, we presented several similarity measures between moving curves, together with efficient algorithms for computing some measures, while proving NP-hardness for computing others.

Although many algorithmic solutions to variants of the Fréchet distance between static curves also apply to the synchronous constant Fréchet distance, extending the synchronous dynamic class is more complex. For example, computing the synchronous dynamic Fréchet distance between moving closed curves (with a cylinder instead of square as parameter space) remains an open problem.

Finally, the performance of the algorithm for the synchronous dynamic Fréchet distance seems undesirably slow and we have not been able to prove tight lower bounds on this running time yet.

**Acknowledgements.** We thank the reviewers for their helpful comments.

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).

[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] K. Buchin, T. Ophelders, and B. Speckmann. Computing the Fréchet distance between real-valued surfaces. In *Proc. 28th Sympos. Discrete Algorithms*, 2017. To appear.

[13] M. Buchin. *On the Computability of the Fréchet Distance Between Triangulated Surfaces*. PhD thesis, Free University Berlin, Institute of Computer Science, 2007.

[14] M. Buchin, S. Dodge, and B. Speckmann. Context-aware similarity of trajectories. In *Proc. 6th Intern. Conf. GIS*, pages 43–56, 2012.

[15] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. ACM*, 34(1):200–208, 1987.

[16] 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.

[17] 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.

[18] M. Godau. *On the complexity of measuring the similarity between geometric objects in higher dimensions*. PhD thesis, Berlin, Freie Universität Berlin, 1999.

[19] S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms*, 10(1):3, 2014.

[20] 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.

[21] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.

[22] A. Nayyeri and H. Xu. On Computing the Fréchet Distance Between Surfaces. In *Proc. 32nd Sympos. Comput. Geom.*, volume 51, pages 55:1–55:15, 2016.