Data Structures for Approximate Fréchet Distance for Realistic Curves

Ivor van der Hoog
DTU Compute, Technical University of Denmark, Lyngby, Denmark

Eva Rotenberg
DTU Compute, Technical University of Denmark, Lyngby, Denmark

Sampson Wong
Department of Computer Science, University of Copenhagen, Denmark

Abstract

The Fréchet distance is a popular distance measure between curves P and Q. Conditional lower bounds prohibit $(1+\varepsilon)$-approximate Fréchet distance computations in strongly subquadratic time, even when preprocessing P using any polynomial amount of time and space. As a consequence, the Fréchet distance has been studied under *realistic* input assumptions, for example, assuming both curves are *c-packed*.

In this paper, we study c-packed curves in Euclidean space $\mathbb{R}^d$ and in general geodesic metrics $\mathcal{X}$. In $\mathbb{R}^d$, we provide a nearly-linear time static algorithm for computing the $(1+\varepsilon)$-approximate continuous Fréchet distance between c-packed curves. Our algorithm has a linear dependence on the dimension d, as opposed to previous algorithms which have an exponential dependence on d.

In general geodesic metric spaces $\mathcal{X}$, little was previously known. We provide the first data structure, and thereby the first algorithm, under this model. Given a c-packed input curve P with n vertices, we preprocess it in $O(n \log n)$ time, so that given a query containing a constant ε and a curve Q with m vertices, we can return a $(1+\varepsilon)$-approximation of the discrete Fréchet distance between P and Q in time polylogarithmic in n and linear in m, $1/\varepsilon$, and the realism parameter c.

Finally, we show several extensions to our data structure; to support dynamic extend/truncate updates on P, to answer map matching queries, and to answer Hausdorff distance queries.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, data structures, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2024.56

Related Version *Full Version*: https://arxiv.org/abs/2407.05114 [44]

Funding *Ivor van der Hoog*: Supported by Independent Research Fund Denmark grant 2020-2023 (9131-00044B) "Dynamic Network Analysis".
Eva Rotenberg: Partially supported by Independent Research Fund Denmark grant 2020-2023 (9131-00044B) "Dynamic Network Analysis" and the Carlsberg Young Researcher Fellowship CF21-0302 "Graph Algorithms with Geometric Applications".

1 Introduction

The Fréchet distance is a popular metric for measuring the similarity between (polygonal) curves P and Q. We assume that P has n vertices and Q has m vertices and that they reside in some geodesic metric space $\mathcal{X}$. The Fréchet distance is often intuitively defined through the following metaphor: suppose that we have two curves that are traversed by a person and their dog. Consider the length of their connecting leash, measured over the metric $\mathcal{X}$. What is the minimum length of the connecting leash over all possible traversals by the person and the dog? The Fréchet distance has many applications; in particular in the analysis and visualization of movement data [11, 14, 35, 47]. It is a versatile measure that can be used for a variety of objects, such as handwriting [41], coastlines [37], outlines of shapes in geographic information systems [20], trajectories of moving objects, such as vehicles, animals or sports players [40, 42, 7, 14], air traffic [6] and protein structures [34].

Alt and Godau [2] compute the continuous Fréchet distance in $\mathbb{R}^2$ under the L_2 metric in $O(mn \log(n+m))$ time. This was later improved by Buchin et al. [12] to $O(nm(\log\log nm)^2)$ time. Eiter and Manila [26] showed how to compute the discrete Fréchet distance in $\mathbb{R}^2$ in $O(nm)$ time, which was later improved by Agarwal et al. [1] to $O(nm(\log\log nm)/\log nm)$ time. Typically, the quadratic $O(nm)$ running time is considered costly. Bringmann [8] showed that, conditioned on the Strong Exponential Time Hypothesis (SETH), one cannot compute a $(1+\varepsilon)$-approximation of the continuous Fréchet distance between curves in $\mathbb{R}^2$ under the L_1, L_2 or L_∞ metric faster than $\Omega((nm)^{1-\delta})$ time for any $\delta > 0$. This lower bound was extended by Buchin, Ophelders and Speckmann [13] to intersecting curves in $\mathbb{R}^1$. Driemel, van der Hoog and Rotenberg [25] extended the lower bound to paths P and Q in a weighted planar graph under the shortest path metric.

Well-behaved curves. Previous works have circumvented lower bounds by assuming that *both* curves come from a well-behaved class. A curve P in a geodesic metric space $\mathcal{X}$ is any sequence of points where consecutive points are connected by their shortest path in $\mathcal{X}$. For a ball B in $\mathcal{X}$, let $P \cap B$ denote all (maximal) segments of P contained in B. A curve P is:

- **κ-straight** (by Alt, Knauer and Wenk [3]) if for every i, j the length of the subcurve from p_i to p_j is $\ell(P[i,j]) \leq \kappa \cdot d(p_i, p_j)$,
- **c-packed** (by Driemel, Har-Peled and Wenk [22]) if for every ball B in the geodesic metric space $\mathcal{X}$ with radius r: the length $\ell(P \cap B) \leq c \cdot r$.
- **ϕ-low-dense** (by van der Stappen [45]; see also [20, 22, 39]) if for every ball B in $\mathcal{X}$ with radius r, there exist at most ϕ edges of length r intersecting B.
- **backbone** (by Aronov et al. [5]) if consecutive vertices have distance between c_1 and c_2 for some constants c_1, c_2, and if non-consecutive vertices have distance at least 1.

Any c-straight curve is also $O(c)$-packed. Parametrized by ε, $\phi \in O(1)$, c and $\kappa = O(c)$, Driemel, Har-Peled and Wenk [22] compute a $(1+\varepsilon)$-approximation of the continuous Fréchet distance between a pair of realistic curves in $\mathbb{R}^d$ under the L_1, L_2, L_∞ metric for constant d in $O(\frac{c(n+m)}{\varepsilon} + c(n+m)\log n)$ time. Their result for c-packed and c-straight curves was improved by Bringmann and Künnemann [10] to $O(\frac{c(n+m)}{\sqrt{\varepsilon}} \log \varepsilon^{-1} + c(n+m)\log n)$, which matches the conditional lower bound for c-packed curves. In particular, Bringmann [8] showed that under SETH, for dimension $d \geq 5$, there is no $O((c(n+m)/\sqrt{\varepsilon})^{1-\delta})$ time algorithm for computing the Fréchet distance between c-packed curves for any $\delta > 0$. Realistic input assumptions have been applied to other geometric problems, e.g. for robotic navigation in ϕ-low-dense environments [45], and map matching of ϕ-low-dense graphs [16] or c-packed graphs [30].

Deciding versus computing. We make a distinction between two problem variants: the decision variant, the optimisation variant. For the decision variant, we are given a value ρ and two curves P and Q and we ask whether the Fréchet distance $D_{\mathcal{F}}(P,Q) \leq \rho$. This variant often solved through navigating an n by m "free space diagram". In the optimization variant, the goal is to output the Fréchet distance $D_{\mathcal{F}}(P,Q)$. To convert any decision algorithm into a optimization algorithm, two techniques are commonly used. The first is binary search over what we will call $\text{TADD}(P,Q)$:

▶ **Definition 1.** *Given two sets of points P, Q in a geodesic metric space $\mathcal{X}$, we define a Two-Approximate Distance Decomposition of P denoted by **TADD(P, Q)** as a set of reals T_{PQ} where for every pair $(p_i, q_j) \in P \times Q$ there exist $a, b \in T_{PQ}$ with $a \leq d(p_i, q_j) \leq b \leq 2a$.*

Essentially a TADD is a two-approximation of the set of all pairwise distances in $P \times Q$ and it can be used to determine, approximately, the (Fréchet) distance values for when the simplification of the input curve changes, or when the reachability of the free space matrix changes. It is known how to compute a TADD from a Well-Separated Pair Decomposition (WSPD) in time linear in the size of the WSPD [22, Lemma 3.8]. A downside of this approach [22] is that, it is only known how to compute a WSPD for doubling metrics [48]. Moreover, for non-constant (doubling) dimensions d, computing the WSPD (and therefore the TADD) takes $O(2^d n + dn \log n)$ time [33, 48], which dominates the running time.

The second technique, deployed when for example TADDs cannot be computed, is parametric search [38]. For decision variants that have a sublinear running or query time of T, the running time of parametric search is commonly $O(T^2)$ [46, 31].

Data structures for Fréchet distance. An interesting question is whether we can store P in a data structure, for efficient (approximate) Fréchet distance queries for any query Q. This topic received considerable attention throughout the years [24, 31, 27, 19, 21, 15, 30]. A related field is nearest neighbor data structures under the Fréchet distance metric [9, 23, 18, 4, 28]. Recently, Gudmundsson, Seybold and Wong [30] answer this question negatively for arbitrary curves in $\mathbb{R}^2$: showing that even with polynomial preprocessing space and time, we cannot preprocess a curve P to decide the continuous Fréchet distance between P and a query curve Q in $\Omega((nm)^{1-\delta})$ time for any $\delta > 0$. Surprisingly, even in very restricted settings data structure results are difficult to obtain. De Berg et al. [19] present an $O(n^2)$ size data structure that restricts the orientation of the query segment to be horizontal. Queries are supported in $O(\log^2 n)$ time, and even subcurve queries are allowed (in that case, using $O(n^2 \log^2 n)$ space). At ESA 2022, Buchin et al. [15] improve these result to using only $O(n \log^2 n)$ space, where queries take $O(\log n)$ time. For arbitrary query segments, they present an $O(n^{4+\delta})$ size data structure that supports (subcurve) queries to arbitrary segments in $O(\log^4 n)$ time. Gudmundsson et al. [31] extend de Berg et al.'s [19] data structure to handle subcurve queries, and to handle queries where the horizontal query segment is translated in order to minimize its Fréchet distance. Driemel and Har-Peled [21] create a data structure to store any curve in $\mathbb{R}^d$ for constant d. They preprocess P in $O(n \log^3 n)$ time and $O(n \log n)$ space. For any query (Q, ε, i, j) they can create a $(3 + \varepsilon)$-approximation of $D_{\mathbb{F}}(P[i, j], Q)$ in $O(m^2 \log n \log(m \log n))$ time.

We state existing data structures for the discrete Fréchet distance. Driemel, Psarros and Schmidt [24] fix ε and an upper bound M beforehand where for all queries Q, they demand that $|Q| \leq M$. They store any curve P in $\mathbb{R}^d$ for constant d using $O((M \log \frac{1}{\varepsilon})^M)$ space and preprocessing, to answer $(1 + \varepsilon)$-approximate Fréchet distance queries in $O(m^2 + \log \frac{1}{\varepsilon})$ time. Filtser [29] gives the corresponding data structure for the discrete Fréchet distance. At SODA 2022, Filtser and Filtser [27] study the same setting: storing P in $O((\frac{1}{\varepsilon})^{dM} \log \frac{1}{\varepsilon})$ space, to answer $(1 + \varepsilon)$-approximate Fréchet distance queries in $\tilde{O}(m \cdot d)$ time.

Contributions. We provide four contributions.

(1) A 1-TADD technique. A crucial step in computing the Fréchet distance is to turn a decision algorithm into an optimization algorithm. TADD(P, Q) is commonly used when approximating the Fréchet distance. Our 1-TADD technique shows a new argument where we map P and Q to curves in $\Lambda \subset \mathbb{R}^1$ and compute only TADD(Λ, Λ) in $O(n \log n)$ time.

Table 1 Our results for computing a $(1+\varepsilon)$-approximation of the Fréchet distance between P and Q. All settings assume that P is a realistic curve, except for [27] who assume an upper bound on $|Q|$. T_ε denotes the query time of a $(1+\varepsilon)$-approximate oracle. The tilde hides lower order factors in terms of n, m and ε. Under the continuous Fréchet distance, we require that Q is also realistic.

Domain	Previous result			Our result		
	Preprocess	Query time	Ref.	Preprocess	Query time	Ref.
$\mathbb{R}^d$ L_p metric	static	$\tilde{O}(2^d n + d^{\frac{c(n+m)}{\sqrt{\varepsilon}}}$ $+d^2 \cdot c(n+m))$	[10]	static	$\tilde{O}(d^{\frac{c(n+m)}{\varepsilon}})$	Thm. 2
Discrete Fréchet distance only:						
$\mathbb{R}^d$ $\|Q\| \leq M$	$\tilde{O}((\frac{1}{\varepsilon})^{dM})$	$\tilde{O}(md)$	[27]	$O(n \log n)$	$\tilde{O}(d \frac{cm}{\varepsilon} \log n)$	Cor. 7
Planar $G = (V, E)$	static	$\tilde{O}(\|V\|^{1+o(1)} + \frac{c(n+m)}{\varepsilon})$	[25]	$\tilde{O}(\|V\|^{1+o(1)})$	$\tilde{O}(\frac{cm}{\varepsilon} \log n)$	Cor. 7
Graph $G = (V, E)$	static	$\tilde{O}(\|V\|^{1+o(1)} + \|E\| \log \|E\|$ $+\frac{c(n+m)}{\varepsilon})$	[25]	$\tilde{O}(\|V\|^{1+o(1)})$	$\tilde{O}(\frac{cm}{\varepsilon} \log n)$	Cor. 7
Simple Polygon P	static	$O(nm \log(n+m))$	[2]	$\tilde{O}(\|P\| + n)$	$\tilde{O}(\frac{cm}{\varepsilon} \log \|P\|)$	Cor. 7
Any geodesic $\mathcal{X}$ with $(1+\varepsilon)$-oracle	static	$O(T_\varepsilon \cdot nm \log n)$	[26]	$O(n \log n)$	$\tilde{O}(T_\varepsilon \frac{cm}{\varepsilon} \log n)$	Thm. 5
Map Matching $G = (V, E)$	$\tilde{O}(c^2 \varepsilon^{-4} n^2)$	$O(m \log m \cdot$ $(\log^4 n + c^4 \varepsilon^{-8} \log^2 n))$	[30]	$\tilde{O}(c^2 \varepsilon^{-4} n^2)$	$O(m(\log n + \log \varepsilon^{-1}) \cdot$ $(\log^2 n + c^2 \varepsilon^{-4} \log n))$	[44]

In Euclidean $\mathbb{R}^d$ this allows us to approximate the discrete and continuous Fréchet distances in time that is linear in d, whereas previous approaches required an exponential dependence on d. In general geodesic metric spaces $\mathcal{X}$, our 1-TADD technique allows us to approximate the discrete Fréchet distance when $\text{TADD}(P, Q)$ cannot be efficiently computed.

(2) Allowing approximate oracles under the discrete Fréchet distance. Many ambient spaces (e.g., Euclidean spaces under floating point arithmetic, and $\mathcal{X}$ as a weighted graph under the shortest path metric.) do not allow for efficient exact distance computations. Thus, we revisit and simplify the argument by Driemel, Har-Peled and Wenk [22]. We assume access to a $(1+\alpha)$-approximate distance oracle with T_α query time. We generalize the previous argument to approximate the discrete Fréchet distance between two curves in any geodesic metric space with approximate distance oracles. For contributions *(1)* and *(2)*, we do not require the curves P and Q to be c-packed.

(3) A data structure under the discrete Fréchet distance. Under the discrete Fréchet distance, we show how to store a c-packed or c-straight curve P with n vertices in *any* geodesic ambient space $\mathcal{X}$. Our solution uses $O(n)$ space and $O(n \log n)$ preprocessing time. For *any* query curve Q, any $0 < \varepsilon < 1$, and any subcurve P^* of P, we can compute a $(1+\varepsilon)$-approximation the discrete Fréchet distance $D_{\mathbb{F}}(P^*, Q)$ using $O(\frac{c \cdot m}{\varepsilon} \log n (T + \log \frac{c \cdot m}{\varepsilon} + \log n))$ time. Here, T is the time required to perform a distance query in the ambient space (e.g., $O(\log n)$ for geodesic distances in a polygon). All times are deterministic and worst-case. This is the Fréchet distance first data structure for realistic curves that avoids spending query time linear in n. Our solution improves various recent results [10, 25, 27, 30] (see Table 1).

(4) Extensions. In the full version of this paper [44], we provide several extensions to our results. We modify our data structure to support updates that truncate the curve P, or extend P, we apply our algorithmic skeleton to map matching queries, and we study Hausdorff distance queries.

2 Preliminaries

Let $\mathcal{X}$ denote some geodesic metric space (e.g., $\mathcal{X}$ is some weighted graph). For any $a, b \in \mathcal{X}$ we denote by $d(a, b)$ their distance in $\mathcal{X}$. A *curve* P in $\mathcal{X}$ is any ordered sequence of points in $\mathcal{X}$, where consecutive points are connected by their shortest path in $\mathcal{X}$. We refer to such points as *vertices*. For any curve P with n vertices, for any integers $i, j \in [n]$ with $i < j$ we denote by $P[i, j]$ the subcurve from p_i to p_j. We denote by $|P[i, j]| = (j - i + 1)$ the size of the subcurve and by $\ell(P[i, j]) := \sum_{k=i}^{j-1} d(p_k, p_{k+1})$ its length. We receive as preprocessing input a curve P where for each pair (p_i, p_{i+1}) we are given $d(p_i, p_{i+1})$.

Distance and distance oracles. Throughout this paper, we assume that for any $\alpha > 0$ we have access to some $(1 + \alpha)$-approximate distance oracle. This is a data structure $\mathcal{D}_{\mathcal{X}}^{\alpha}$ that for any two $a, b \in \mathcal{X}$ can report a value $d^\circ(a, b) \in [(1 - \alpha)d(a, b), (1 + \alpha)d(a, b)]$ in $O(T_\alpha)$ time. To distinguish between inaccuracy as a result of our algorithm and as a result of our oracle, we refer to $d^\circ(a, b)$ as the *perceived value* (as opposed to an approximate value).

Discrete Fréchet distance. Given two curves P and Q in $\mathcal{X}$, we denote by $[n]\times[m]$ the n by m integer lattice. We say that an ordered sequence F of points in $[n] \times [m]$ is a *discrete walk* if for every consecutive pair $(i, j), (k, l) \in F$, we have $k \in \{i-1, i, i+1\}$ and $l \in \{j-1, j, j+1\}$. It is furthermore *xy-monotone* when we restrict to $k \in \{i, i + 1\}$ and $l \in \{j, j + 1\}$. Let F be a discrete walk from $(1, 1)$ to (n, m). The *cost* of F is the maximum over $(i, j) \in F$ of $d(p_i, q_j)$. The discrete Fréchet distance is the minimum over all xy-monotone walks F from $(1, 1)$ to (n, m) of its associated cost: $D_{\mathbb{F}}(P, Q) := \min_F \text{cost}(F) = \min_F \max_{(i,j)\in F} d(p_i, q_j)$. In this paper we, given a $(1 + \alpha)$-approximate distance oracle, define the *perceived* discrete Fréchet distance as $D_{\mathbb{F}}{}^\circ$, obtained by replacing in the above definition $d(p_i, q_j)$ by $d^\circ(p_i, q_j)$.

Free space matrix (FSM). The FSM for a fixed $\rho^* \geq 0$ is a $|P| \times |Q|$, $(0, 1)$-matrix where the cell (i, j) is zero if and only if the distance between the i'th point in P and the j'th point in Q is at most ρ^*. Per definition, $D_{\mathbb{F}}(P, Q) \leq \rho^*$ if and only if there exists an xy-monotone discrete walk F from $(1, 1)$ to (n, m) where for all $(i, j) \in F$: the cell (i, j) is zero.

Continuous Fréchet distance and Free Space Diagram (FSD). We define the continuous Fréchet distance in a geodesic metric space. Given a curve P, we consider P as a continuous function mapping at time $t \in [0, 1]$ to a point $P(t)$ in $\mathcal{X}$. The continuous Fréchet distance is $D_{\mathcal{F}}(P, Q) := \inf_{\alpha,\beta} \max_{t\in[0,1]} d(P(\alpha(t)), Q(\beta(t)))$, where $\alpha, \beta : [0, 1] \to [0, 1]$ are non-decreasing surjections. For a fixed ρ^*, we can define the Free Space Diagram of (P, Q, ρ^*) to be a $[0, 1] \times [0, 1]$, (0,1)-matrix where the point (t, t') is zero if and only if the distance between $P(t)$ and $Q(t')$ is at most ρ^*. The diagram consists of nm cells corresponding to all pairs of edges of P and Q. A cell is *reachable* if there exists an xy-monotone curve from $(0, 0)$ to a point in the cell where all points (t, t') on the curve are zero. The continuous Fréchet distance is at most ρ^* if and only if there exists an xy-monotone curve from $(0, 0)$ to $(1, 1)$ where all points (t, t') on the curve are zero.

Defining discrete queries. Our data structure input is a curve $P = (p_1, p_2, \ldots, p_n)$. The number of vertices m of Q is part of the query input and may vary. Let $D_{\mathbb{F}}(P, Q)$ denote the discrete Fréchet distance between P and Q. We distinguish between four types of (approximate) queries. The input pararmeters are given at query time:

- **A-decision($\boldsymbol{Q, \varepsilon, \rho}$):** for $\rho \geq 0$ and $0 < \varepsilon < 1$ outputs a Boolean concluding either $D_{\mathbb{F}}(P, Q) > \rho$, or $D_{\mathbb{F}}(P, Q) \leq (1 + \varepsilon)\rho$ (these two options are not mutually exclusive).
- **A-value($\boldsymbol{Q, \varepsilon}$):** for $0 < \varepsilon < 1$ outputs a value in $[(1 - \varepsilon)D_{\mathbb{F}}(P, Q), (1 + \varepsilon)D_{\mathbb{F}}(P, Q)]$.
- **Subcurve-decision($\boldsymbol{Q, \varepsilon, \rho, i, j}$):** for $\rho \geq 0$ and $0 < \varepsilon < 1$ outputs a Boolean concluding either $D_{\mathbb{F}}(P[i, j], Q) > \rho$, or $D_{\mathbb{F}}(P[i, j], Q) \leq (1 + \varepsilon)\rho$.
- **Subcurve-value($\boldsymbol{Q,\ \varepsilon,\ i,\ j}$):** for $0 < \varepsilon < 1$ outputs a value in $[(1 - \varepsilon)D_{\mathbb{F}}(P[i, j], Q), (1 + \varepsilon)D_{\mathbb{F}}(P[i, j], Q)]$.

We want a solution that is efficient in time and space, where time and space is measured in units of ε, n, m, ρ and the distance oracle query time T_α.

Previous works: $\boldsymbol{\mu}$-simplifications. Driemel, Har-Peled and Wenk [22] , for a parameter $\mu \in \mathbb{R}$, construct a curve P^μ as follows. Start with the initial vertex p_1, and set this as the current vertex p_i. Next, scan the polygonal curve to find the first vertex p_j such that $d(p_i, p_j) > \mu$. Add p_j to P^μ, and set p_j as the current vertex. Continue this process until we reach the end of the curve. Finally, add the last vertex p_n to P^μ. Driemel, Har-Peled and Wenk [22] observe any μ-simplified curve P^μ can be computed in linear time and $D_{\mathcal{F}}(P^\mu, P) \leq \mu$. This leads to the following approximate decision algorithm:

1. Given P, ε, Q and ρ, construct $P^{\frac{\varepsilon\rho}{4}}$ and $Q^{\frac{\varepsilon\rho}{4}}$ in $O(n + m)$ time.
2. Denote by X the reachable cells in the FSD of $(P^{\frac{\varepsilon\rho}{4}}, Q^{\frac{\varepsilon\rho}{4}}, \rho^* = (1 + \varepsilon/2)\rho)$.
3. Iterating over X, doing $O(|X|)$ distance computations, test if $D_{\mathcal{F}}(P^{\frac{\varepsilon\rho}{4}}, Q^{\frac{\varepsilon\rho}{4}}) \leq \rho^*$.
   - They prove that: if yes then $D_{\mathcal{F}}(P, Q) \leq (1 + \varepsilon)\rho$. If no then $D_{\mathcal{F}}(P, Q) > \rho$.
   - If P and Q are c-packed, they upper bound $|X|$ by $O(\frac{c(n+m)}{\varepsilon})$.

This scheme is broadly applicable to various domains, see [25, 17]. In this paper, we apply this technique to answer value queries at the cost of a factor $O(\log n + \log \varepsilon^{-1})$. Under the discrete Fréchet distance, we extend the analysis to work with approximate distance oracles. Finally, we show a data structure to execute step 3 in time independent of $|P| = n$. We also show that under the discrete Fréchet distance, it suffices to assume that only P is c-packed.

2.1 Results

(1) A 1-TADD technique for the Fréchet distance. For any $\mu > 0$, we denote by P^μ and Q^μ their μ-simplified curves according to our new definition of μ-simplification. We show in Section 4 that our new definition allows us to efficiently transform existing decision algorithms into approximation algorithms in $\mathbb{R}^d$ [22]. We assume access to exact $O(d)$-time distance oracle in $\mathbb{R}^d$ and prove:

▶ **Theorem 2.** *We can preprocess a pair of c-packed curves (P, Q) in $\mathbb{R}^d$ under any L_p metric with $|P| = n \geq |Q| = m$ in $O(n \log n)$ time s.t.: given any ε and an exact distance oracle, we can compute a $(1 + \varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$ in $O(d\frac{c(n+m)}{\varepsilon} \cdot (\log n + \log \varepsilon^{-1}))$ time.*

(2) Allowing approximate oracles under the discrete Fréchet distance. In Section 5, we show that (for computing the discrete Fréchet distance) it suffices to have access to a $(1 + \alpha)$-approximate distance oracle. This will enable us to approximate $D_{\mathbb{F}}(P, Q)$ in ambient spaces such as planar graphs and simple polygons. Formally, we show:

▶ **Lemma 3.** *For any $\rho > 0$ and $0 < \varepsilon < 1$, choose $\rho^* = (1 + \frac{1}{2}\varepsilon)\rho$ and $\mu \leq \frac{1}{6}\varepsilon\rho$. Let $\mathcal{X}$ be any geodesic metric space and $\mathcal{D}_{\mathcal{X}}^{\varepsilon/6}$ be a $(1 + \frac{1}{6}\varepsilon)$-approximate distance oracle. For any curve $P = (p_1, \ldots, p_n)$ in $\mathcal{X}$ and any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$:*
- *If for the discrete Fréchet distance, $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q) \leq \rho^*$ then $D_{\mathbb{F}}(P, Q) \leq (1 + \varepsilon)\rho$.*
- *If for the discrete Fréchet distance, $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q) > \rho^*$ then $D_{\mathbb{F}}(P, Q) > \rho$.*

We note that for ambient spaces such as planar graphs and simple polygons, there is no clear way to define a continuous μ-simplification (or even continuous Fréchet distance).

(3) An efficient data structure under the discrete Fréchet distance. Finally, in Section 6, we study computing the discrete Fréchet distance in a data structure setting. We show that under the discrete Fréchet distance, it suffices to assume that only P is c-packed or c-straight. Moreover, we can store P in a data structure such that we can answer approximate Fréchet value queries $D_{\mathbb{F}}(P, Q)$ in time linear in m and polylogarithmic in n:

▶ **Theorem 4.** *Let $\mathcal{X}$ be any geodesic space and $\mathcal{D}_{\mathcal{X}}^{\alpha}$ be a $(1+\alpha)$-approximate distance oracle with $O(T_\alpha)$ query time. Let $P = (p_1, \ldots, p_n)$ be any c-packed curve in $\mathcal{X}$. We can store P using $O(n)$ space and preprocessing, such that for any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$ and any $\rho > 0$ and $0 < \varepsilon < 1$, we can answer A-decision(Q, ε, ρ) for the discrete Fréchet distance in:*

$$O\left(\frac{c \cdot m}{\varepsilon} \cdot (T_{\varepsilon/6} + \log n)\right) \text{ time.}$$

We may apply the proof of Theorem 2 to Theorem 4 to answer A-value(Q, ε) in any geodesic metric space by increasing the running time by a factor $O(\log n + \log \varepsilon^{-1})$. However, under the discrete Fréchet distance we can be more efficient:

▶ **Theorem 5.** *Let $\mathcal{X}$ be a geodesic metric space and $\mathcal{D}_{\mathcal{X}}^{\alpha}$ be a $(1 + \alpha)$-approximate distance with $O(T_\alpha)$ query time. Let $P = (p_1, \ldots, p_n)$ be any c-packed curve in $\mathcal{X}$. We can store P using $O(n)$ space and $O(n \log n)$ preprocessing time, such that for any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$ and any $0 < \varepsilon < 1$, we can answer A-Value(Q, ε) for the discrete Fréchet distance in:*

$$O\left(\frac{c \cdot m}{\varepsilon} \cdot \log n \cdot \left(T_{\varepsilon/6} + \log \frac{c \cdot m}{\varepsilon} + \log n\right)\right) \text{ time.}$$

The advantage of our result is that it applies to a variety of metric spaces $\mathcal{X}$, while also improving upon previous *static* algorithms in those spaces. Here, static refers to solutions that do not require preprocessing or building a data structure. For a complete overview of the improvements we make to the state-of-the-art, we refer to Table 1.

Our result does not rely upon complicated techniques such parametric search [31, 30], higher-dimensional envelopes [15], or advanced path-simplification structures [21, 24, 27]. Our techniques are not only generally applicable, but also appear implementable (e.g., the authors of [15] mention that their result is un-implementable).

2.2 Corollaries

Theorems 4+5 are the first data structures for computing the Fréchet distance between c-packed curves. Our construction has two novelties: first, we consider c-packed curves in *any* metric space $\mathcal{X}$, and only require access to *perceived* distances (distance oracles that can report a $(1 + \alpha)$-approximation in $O(T_\alpha)$ time). Second, we propose a new 1-TADD technique that can be used to compute the discrete Fréchet distance independently of the geodesic ambient space $\mathcal{X}$. These two novelties imply improvements to previous results, even static results for computing the Fréchet distance:

Applying perceived distances. Our analysis allows us to answer the decision variant for the discrete Fréchet distance for any geodesic metric space for which there exist efficient $(1+\alpha)$-approximate distance oracles (See Oracles 12). Combining Theorem 4 with Oracles 12 we obtain:

▶ **Corollary 6.** *Let P be a c-packed curve in a metric space $\mathcal{X}$.*

- *For $\mathcal{X} = \mathbb{R}^d$ under L_1, L_2, L_∞ metric in the real-RAM model, we can store P using $O(n)$ space and preprocessing, to answer A-decision(Q, ε, ρ) in $O\left(\frac{cm}{\varepsilon} \cdot (d + \log n)\right)$ time.*
  - *Improving the static $O(d \cdot \frac{cn}{\sqrt{\varepsilon}} + d \cdot cn \log n)$ algorithm by Bringmann and Künnemann [10, IJCGA'17]: making it faster when $n > m$ and making it a data structure.*
- *For $\mathcal{X} = \mathbb{R}^d$ under L_2 in word-RAM, we can store P using $O(n)$ space and $O(n \log n)$ preprocessing, to answer A-decision(Q, ε, ρ) in $O\left(\frac{cm}{\varepsilon} \cdot \left(d \log \varepsilon^{-1} + \log n\right)\right)$ worst case time.*
  - *Improving the static expected $O(d^2 \cdot \frac{cn}{\sqrt{\varepsilon}} + d^2 \cdot cn \log n)$ algorithm by Bringmann and Künnemann [10, IJCGA'17]: saving a factor d and obtaining deterministic guarantees.*
- *For $\mathcal{X}$ an N-vertex planar graph under the shortest path metric, we can store P using $O(N^{1+o(1)})$ space and preprocessing, to answer A-decision(Q, ε, ρ) in $O\left(\frac{cm}{\varepsilon} \cdot \log^{2+o(1)} N\right)$ time.*
  - *Generalizing the static $O\left(N^{1+o(1)} + \frac{c \cdot m}{\varepsilon} \log^{2+o(1)} N\right)$ algorithm by Driemel, van der Hoog and Rotenberg [25, SoCG'22] to a data structure.*
- *For $\mathcal{X}$ an N-vertex graph under the shortest path metric, we can fix ε and store P using $O(\frac{N}{\varepsilon} \log N)$ space and preprocessing, to answer A-decision(Q, ε, ρ) in $O\left(\frac{cm}{\varepsilon} \cdot (\varepsilon^{-1} + \log n)\right)$ time.*
  - *Generalizing the static $O\left(N^{1+o(1)} + \frac{c \cdot m}{\varepsilon} \log^{2+o(1)} N\right)$ algorithm by Driemel, van der Hoog and Rotenberg [25, SoCG'22] to a data structure.*

Applying the 1-TADD. We can transform the decision variant of the Fréchet distance to the optimization variant, by using only a well-separated pair decomposition of P (mapped to $\mathbb{R}^1$) with itself. This allows us to answer the optimization variant for the discrete Fréchet distance without an exponential dependence on the dimension (speeding up even static algorithms). In full generality, combining Theorem 4 with Oracles 12 implies:

▶ **Corollary 7.** *Let P be a c-packed curve in a metric space $\mathcal{X}$.*

- *For $\mathcal{X} = \mathbb{R}^d$ in the real-RAM model, we can store P using $O(n)$ space and $O(n \log n)$ preprocessing, to answer A-value(Q, ε) in $O\left(\frac{cm}{\varepsilon} \log n (d + \log \frac{c \cdot m}{\varepsilon} + \log n\right)$ time.*
  - *Improving the static $O(2^d n + d \cdot \frac{cn}{\sqrt{\varepsilon}} + d^2 \cdot cn \log n)$ algorithm by Bringmann and Künnemann [10, IJCGA'17]: removing the exponential dependency on the dimension.*
  - *Improving the dynamic $O(\left(O(\frac{1}{\varepsilon})\right)^{dm} \log \frac{1}{\varepsilon})$ space solution with $\tilde{O}(m \cdot d)$ query time by Filtser and Filtser [27, SODA'21]: allowing Q to have arbitrary length, and using linear as opposed to exponential space and preprocessing time. This applies only when P is c-packed.*
- *For $\mathcal{X} = \mathbb{R}^d$ under the L_2 metric in word-RAM, we can store P using $O(n)$ space and $O(n \log n)$ preprocessing, to answer A-value(Q, ε) in $O\left(\frac{cm}{\varepsilon} \log n (d \log n + \log \frac{c \cdot m}{\varepsilon}\right)$ time.*
  - *Improving upon the static expected $O(2^d n + d^2 \cdot \frac{cn}{\sqrt{\varepsilon}} + d^3 \cdot cn \log n)$ algorithm by Bringmann and Künnemann [10, IJCGA'17]: saving a factor $d2^d$ with deterministic guarantees.*
- *For $\mathcal{X}$ an N-vertex planar graph under the SP metric, we can store P using $O(N^{1+o(1)})$ space and preprocessing, to answer A-value(Q, ε) in $O\left(\frac{c \cdot m}{\varepsilon} \log n \cdot (\log^{2+o(1)} N + \log \frac{c \cdot m}{\varepsilon})\right)$ time.*
  - *Improving the static $O\left(N^{1+o(1)} + |E| \log |E| + \frac{c \cdot m}{\varepsilon} \log^{2+o(1)} N \log |E|\right)$ algorithm by Driemel, van der Hoog and Rotenberg [25, SoCG'22]: making it a data structure.*

- *For $\mathcal{X}$ an N-vertex graph under the SP metric, we can fix ε and store P using $O(\frac{N}{\varepsilon}\log N)$ space and preprocessing, to answer A-value(Q, ε) in $O\left(\frac{cm}{\varepsilon} \cdot \log n \cdot (\varepsilon^{-1} + \log \frac{c \cdot m}{\varepsilon} + \log n)\right)$ time.*
  - *Same improvement as above, except that this result is not adaptive to ε.*
- *For $\mathcal{X}$ an N-vertex simple polygon under geodesics, we can store P using $O(N \log N + n)$ space and preprocessing, to answer A-value(Q, ε) in $O\left(\frac{cm}{\varepsilon} \cdot \log n \cdot (\log N + \log \frac{c \cdot m}{\varepsilon} + \log n)\right)$ time.*
  - *No realism-parameter algorithm was known in this setting, because no TADD can be computed in this setting.*

We briefly note that all our results are also immediately applicable to subcurve queries:

▶ **Corollary 8.** *All results obtained in Section 6 can answer the subcurve variants of the A-decision and A-value queries for any $i, j \in [n]$ at no additional cost.*

3 Simplification and a data structure

To facilitate computations in arbitrary geodesic metric spaces, we modify the definition of μ-simplifications. Our modified definition has the same theoretical guarantees as the previous definition, but works in arbitrary metrics. Formally, we say that henceforth the μ-simplification is a curve obtained by starting with p_1, and recursively adding the first p_j such that the *length* of the subtrajectory $\ell(P[i, j]) > \mu$, where p_i is the last vertex added to the simplified curve. This way, our μ-simplifications (and their computation) are independent of the ambient space and only depend on the edge lengths.
We construct a data structure such that for any value μ, we can efficiently obtain P^μ:

▶ **Definition 9.** *For any curve P in $\mathcal{X}$ with n vertices, for each $1 < i \leq n$ we create a half-open interval $(\ell(P[1, i-1]), \ell(P[1, i])]$ in $\mathbb{R}^1$. This results in an ordered set of $O(n)$ disjoint intervals on which we build a balanced binary tree in $O(n)$ time.*

Our new definition and data structure allow us to obtain P^μ at query time:

▶ **Lemma 10.** *Let $P = (p_1, \ldots, p_n)$ be a curve in $\mathcal{X}$ stored in the data structure of Definition 9. For any value $\mu \geq 0$, any pair (i, j) with $i < j$, and any integer N we can report the first N vertices of the discrete μ-simplification $P[i, j]^\mu$ in $O(N \log n)$ time.*

Proof. The first vertex of $P[i, j]^\mu$ is p_i. We inductively add subsequent vertices. Suppose that we just added p_x to our output. We choose the value $a = \ell(P[1, x]) + \mu$. We binary search in $O(\log n)$ time for the point p_y where the interval $(\ell(P[1, y-1]), \ell(P[1, y])]$ contains a. Per definition: the length $\ell(P[x, y]) \geq \mu$. Moreover, for all $z \in (x, y)$ the length $\ell(P[x, z]) < \mu$. Thus, p_y is the successor of p_x and we recurse if necessary. ◀

4 The 1-TADD technique

Let P and Q be curves in $\mathbb{R}^d$ under the Euclidean metric. In [22], they show an algorithm that (given the μ-simplified curves P^μ and Q^μ for $\mu = \varepsilon\rho/4$) they can decide whether $D_{\mathcal{F}}(P, Q) > \rho$ or $D_{\mathcal{F}}(P, Q) \leq (1 + \varepsilon)\rho$ in $O(d^{\frac{c(n+m)}{\varepsilon}})$ time. They then compute a $(1 + \varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$ through a binary search over TADD(P, Q). This approach scales poorly with the dimension d because computing TADD(P, Q) has an exponential dependency on d. We alleviate this through our 1-TADD definiton:

▶ **Definition 11.** *Given P, map each vertex p_i to $\lambda_i = \ell(P[1,i])$. Denote by $\Lambda = \{\lambda_i\}_{i=1}^n$. We define 1-TADD(P) as TADD(Λ, Λ).*

Our 1-TADD can be computed in $O(n \log n)$ time using $O(n)$ space [22].

▶ **Theorem 2.** *We can preprocess a pair of c-packed curves (P, Q) in $\mathbb{R}^d$ under any L_p metric with $|P| = n \geq |Q| = m$ in $O(n \log n)$ time s.t.: given any ε and an exact distance oracle, we can compute a $(1+\varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$ in $O(d\frac{c(n+m)}{\varepsilon} \cdot (\log n + \log \varepsilon^{-1}))$ time.*

Proof. With slight abuse of notation, we say that $\mathrm{A}(P, Q, \varepsilon, \rho)$ is an algorithm that takes as input $P^{\varepsilon\rho/4}$ and $Q^{\varepsilon\rho/4}$ and outputs either $D_{\mathcal{F}}(P, Q) > \rho$ or $D_{\mathcal{F}}(P, Q) \leq (1+\varepsilon)\rho$. We briefly note given our new definition of μ-simplification, [22] present an $\mathrm{A}(P, Q, \varepsilon, \rho)$ algorithm with a runtime of $O(d\frac{c(n+m)}{\varepsilon})$ under any L_p metric (as Lemma 4.4 in [22] immediately works for our simplification definition). We use $\mathrm{A}(P, Q, \varepsilon, \rho)$ to approximate $D_{\mathcal{F}}(P, Q)$.

We preprocess P and Q by computing T_P = 1-TADD(P) and T_Q = 1-TADD(Q). We denote by I the set of intervals obtained by taking for each $a \in T_P \cup T_Q$ the interval $[4\varepsilon^{-1}a, 8\varepsilon^{-1}a]$ (we add the interval $[0, 0]$ to I). Given $\mathrm{A}(P, Q, \varepsilon, \rho)$ that runs in $O(d\frac{c(n+m)}{\varepsilon})$ time, we do binary search over I. Specifically, we iteratively select an interval $[a, b] \in I$ and run $\mathrm{A}(P, Q, \varepsilon, \rho)$ for ρ equal to either endpoint.

Note that for each ρ, we may use Lemma 10 to obtain both the $\varepsilon\rho/4$-simplifications of P and Q in $O((n + m) \log(n + m))$ time − which are required as input for $\mathrm{A}(P, Q, \varepsilon, \rho)$. We need to do this procedure at most $O(\log n)$ times before we reach one of two cases:

- Case 1: There exists $[a, b] \in I$, such that $D_{\mathcal{F}}(P, Q) > a$ and $D_{\mathcal{F}}(P, Q) < (1+\varepsilon)b$. We note that by definition of I, the values a and b differ by a factor 2. Thus, we may discretize the interval $[a, b]$ into $O(\varepsilon^{-1})$ points that are each at most $\frac{\varepsilon \cdot a}{2}$ apart (note that we implicitly discretize this interval, as an explicit discretization takes ε^{-1} time). By performing binary search over this discretized set, we report a $(1+\varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$ by using $\mathrm{A}(P, Q, \varepsilon, \rho)$ at most $O(\log \varepsilon^{-1})$ times.
- Case 2: There exists no $[x, y] \in I$ such that $D_{\mathcal{F}}(P, Q) > x$ and $D_{\mathcal{F}}(P, Q) < (1+\varepsilon)y$.
  - Denote by $[a_{max}, b_{max}]$ the right-most interval in I. Consider the special case where $D_{\mathcal{F}}(P, Q) > b_{max}$. Since $b_{max} \geq \ell(P), \ell(Q)$ it follows that all $\rho > b_{max}$, P^μ and Q^μ for $\mu = \varepsilon\rho/4$ are an edge. Computing $D_{\mathcal{F}}(P^\mu, Q^\mu)$ can therefore be done in $O(d)$ time which gives a $(1+\varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$.

  If the special case does not apply then there exist two intervals $[a, b], [e, f] \in I$ such that $D_{\mathcal{F}}(P, Q) > b$ and $D_{\mathcal{F}}(P, Q) \leq (1+\varepsilon)e$, such that there exists no interval $[x, y] \in I$ that intersects $[b, e]$. We claim that all $\rho_1, \rho_2 \in [b, e]$: $P^{\varepsilon\rho_1/4} = P^{\varepsilon\rho_2/4}$ and $Q^{\varepsilon\rho_1/4} = Q^{\varepsilon\rho_2/4}$.
  - Indeed, suppose for the sake of contradiction that $P^{\varepsilon\rho_1/4} \neq P^{\varepsilon\rho_2/4}$. Let $\rho_1 < \rho_2$ and choose without loss of generality the smallest ρ_2 for which this is the case. Then there must exist a pair $p_i, p_j \in P$ where $\ell(P[i, j]) = \varepsilon\rho_2/4$. However, the distance $\ell(P[i, j])$ is the distance between λ_i and λ_j in the curve Λ and so there exist $a', b' \in T_P$ with $a' \leq \ell(P[i, j]) \leq b' \leq 2a'$. It follows that $\rho_2 = 4\varepsilon^{-1}\ell(P[i, j])$ lies in an interval in I which is a contradiction with the assumption that $\rho_1, \rho_2 \in [b, e]$.

  We choose $\rho = e$. Denote by X the set of reachable cells the Free Space Diagram of $(P^{\varepsilon\rho/4}, Q^{\varepsilon\rho/4}, \rho^*)$. The set X contains $O(\frac{c(n+m)}{\varepsilon})$ cells [22, Lemma 4.4]. It follows that there are $O(\frac{c(n+m)}{\varepsilon})$ values ρ' for which the reachability of X changes. We compute and sort these to get a sorted set R.

  Suppose for some $\rho' \in [b, \rho]$ that $D_{\mathcal{F}}(P, Q) \leq \rho'$. Denote by F an xy-monotone path in the Free Space Diagram of $(P^{\varepsilon\rho'/4}, Q^{\varepsilon\rho'/4}, \rho') = (P^{\varepsilon\rho/4}, Q^{\varepsilon\rho/4}, \rho')$. Per definition, F lies within X. Thus, we may binary search over the set $R \cap [b, \rho]$ (applying the ε-approximate decider at every step) to compute a $(1+\varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$. ◀

5 Approximate distance oracles under the discrete Fréchet distance

We want to approximate $D_{\mathbb{F}}(P, Q)$ for curves P and Q that live in any geodesic ambient space $\mathcal{X}$. In most ambient spaces we do not have access to efficient exact distance oracles. In many ambient spaces however, it is possible to compute for any $\alpha > 0$ some $(1 + \alpha)$-approximate distance oracle. This is a data structure $\mathcal{D}^{\alpha}_{\mathcal{X}}$ that for any two $a, b \in \mathcal{X}$ can report a value $d^{\circ}(a, b) \in [(1 - \alpha)d(a, b), (1 + \alpha)d(a, b)]$ in $O(T_\alpha)$ time. To distinguish between inaccuracy as a result of our algorithm and as a result of our oracle, we refer to $d^{\circ}(a, b)$ as the *perceived value* (as opposed to an approximate value).

▶ **Oracles 12.** *We present some examples of approximate distance oracles:*

- *For $\mathcal{X} \subseteq \mathbb{R}^d$ under the L_1, L_2, L_∞ metric in real-RAM we can compute the exact $d(a, b)$ in $O(d)$ time. Thus, for any α, we have an oracle $\mathcal{D}^{\alpha}_{\mathcal{X}}$ with $T_\alpha = O(d)$ query time.*
- *For $\mathcal{X} \subseteq \mathbb{R}^d$ under the L_2 metric executed in word-RAM, we can compute $d(a, b)$ in $O(d^2)$ expected time. Thus, we have an oracle $\mathcal{D}^{\alpha}_{\mathcal{X}}$ with $O(d^2)$ expected query time.*
- *For any $\mathcal{X} \subseteq \mathbb{R}^d$ under the L_2 metric in word-RAM, we can $(1 + \alpha)$-approximate the distance between two points in $T_\alpha = O(d \log \alpha^{-1})$ worst case time using Taylor expansions.*
- *For $\mathcal{X}$ a planar weighted graph, Long and Pettie [36] store $\mathcal{X}$ with N vertices using $O\left(N^{1+o(1)}\right)$ space, to answer exact distance queries in $O\left((\log(N))^{2+o(1)}\right)$ time.*
- *For $\mathcal{X}$ as an arbitrary weighted graph, Thorup [43] compute a $(1+\alpha)$-approximate distance oracle in $O(N/\alpha \log N)$ time and space, and with a query-time of $O(1/\alpha)$.*
- *For $\mathcal{X}$ a simple N-vertex polygon, Guibas and Hershberger [32] store $\mathcal{X}$ in $O(N \log N)$ time in linear space, and answer exact geodesic distance queries in $O((\log N)$ time.*

We prove that we may approximately decide the Fréchet distance between P and Q using a $(1 + \alpha)$-approximate distance oracle (for the discrete Fréchet distance).

▶ **Lemma 3.** *For any $\rho > 0$ and $0 < \varepsilon < 1$, choose $\rho^* = (1 + \frac{1}{2}\varepsilon)\rho$ and $\mu \leq \frac{1}{6}\varepsilon\rho$. Let $\mathcal{X}$ be any geodesic metric space and $\mathcal{D}^{\varepsilon/6}_{\mathcal{X}}$ be a $(1 + \frac{1}{6}\varepsilon)$-approximate distance oracle. For any curve $P = (p_1, \ldots, p_n)$ in $\mathcal{X}$ and any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$:*

- *If for the discrete Fréchet distance, ${D_{\mathbb{F}}}^{\circ}(P^\mu, Q) \leq \rho^*$ then $D_{\mathbb{F}}(P, Q) \leq (1 + \varepsilon)\rho$.*
- *If for the discrete Fréchet distance, ${D_{\mathbb{F}}}^{\circ}(P^\mu, Q) > \rho^*$ then $D_{\mathbb{F}}(P, Q) > \rho$.*

Proof. Per definition of $\mathcal{D}^{\varepsilon/6}_{\mathcal{X}}$: $\forall (p, q) \in P \times Q$, $d^{\circ}(p, q) \in [(1 - \frac{1}{6}\varepsilon)d(p, q), (1 + \frac{1}{6}\varepsilon)d(p, q)]$. It follows from $0 < \varepsilon < 1$ that:

$$\forall (p, q) \in P \times Q : \quad d(p, q) \leq \left(1 + \frac{1.1}{6}\varepsilon\right) d^{\circ}(p, q) \quad \wedge \quad d^{\circ}(p, q) \leq \left(1 + \frac{1}{6}\varepsilon\right) d(p, q).$$

Suppose that ${D_{\mathbb{F}}}^{\circ}(P^\mu, Q) \leq \rho^*$. There exists a (monotone) discrete walk F through $P^\mu \times Q$ such that for each $(i, j) \in F$: $d^{\circ}(P^\mu[i], q_j) \leq \rho^* = (1 + \frac{1}{2}\varepsilon)\rho$. It follows that:

$$d(P^\mu[i], q_j) \leq \left(1 + \frac{1.1}{6}\varepsilon\right) d^{\circ}(P^\mu[i], q_j) \leq \left(1 + \frac{1.1}{6}\varepsilon\right)\left(1 + \frac{1}{2}\varepsilon\right)\rho \leq \left(1 + \frac{5}{6}\varepsilon\right)\rho.$$

We will prove that this implies $D_{\mathbb{F}}(P, Q) \leq (1 + \varepsilon)\rho$. We use F to construct a discrete walk F' through $P \times Q$. For each consecutive pair $(a, b), (c, d) \in F$ note that since F is a discrete walk, $P^\mu[a]$ and $P^\mu[c]$ are either the same vertex or incident vertices on P^μ. Denote by P_{ac} the vertices of P in between $P^\mu[a]$ and $P^\mu[c]$. It follows that:

$$\forall p' \in P_{ac} : \quad d(p', q_b) \leq d\left(P^\mu[a], q_b\right) + \mu \leq (1 + \frac{5}{6}\varepsilon)\rho + \frac{1}{6}\varepsilon\rho = (1 + \varepsilon)\rho.$$

Now consider the following sequence of pairs of points:

$L_{ac} = (P^\mu[a], q_b) \cup \{(p', q_b) \mid p' \in P_{ac}\} \cup (P^\mu[c], q_d)$. We add the lattice points corresponding to L_{ac} to F'. It follows that we create a discrete walk F' in the lattice $|P| \times |Q|$ where for each $(i, j) \in F'$: $d(p_i, q_j) \leq (1+\varepsilon)\rho$. Thus, $D_{\mathbb{F}}(P, Q) \leq (1+\varepsilon)\rho$.

Suppose otherwise that $D_{\mathbb{F}}(P, Q) \leq \rho$. We will prove that $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q) \leq \rho^*$. Indeed, consider a discrete walk F' in the lattice $|P| \times |Q|$ where for each $(i, j) \in F'$: $d(p_i, q_j) \leq \rho$. We construct a discrete walk F in $|P^\mu| \times |Q|$. Consider each $(i, j) \in F'$, If $p_i = P^\mu[a]$ for some integer a, we add (a, j) to F. Otherwise, denote by $P^\mu[a]$ the last vertex on P^μ that precedes p_i: we add (a, j) to F. Note that per definition of μ-simplification, $d(P^\mu[a], q_j) \leq d(p_i, q_j) + \mu \leq (1 + \frac{1}{6}\varepsilon)\rho$. It follows from the definition of our approximate distance oracle that $d^\circ(P^\mu[a], q_j) \leq (1 + \frac{1}{6}\varepsilon)(1 + \frac{1}{6}\varepsilon)\rho < (1 + \frac{1}{2}\varepsilon)\rho = \rho^*$. Thus, we may conclude that $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q) \leq \rho^*$. ◀

6 Approximate Discrete Fréchet distance

We denote by $\mathcal{D}^\alpha_{\mathcal{X}}$ a $(1+\alpha)$-approximate distance oracle over the geodesic metric space $\mathcal{X}$. Our input is some curve $P = (p_1, \ldots, p_n)$ in $\mathcal{X}$ which is c-packed in $\mathcal{X}$. We preprocess P to:

- answer A-decision(Q, ε, ρ) for any curve $Q = (q_1, \ldots, q_m)$, $\rho > 0$ and $0 < \varepsilon < 1$,
- answer A-value(Q, ε) for any curve $Q = (q_1, \ldots, q_m)$ and $0 < \varepsilon < 1$.

We obtain this result in four steps. In Section 5, we showed that we can answer A-decision(Q, ε, ρ) through comparing if the perceived Fréchet distance $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q) \leq \rho^*$ for conveniently chosen μ and ρ^*. In Section 6.1 we define what we call the perceived free-space matrix. This is a $(0,1)$-matrix $M^{A\times Q}_{\rho^*}$ for any two curves A and Q and any $\rho^* > 0$. We show that if A is the μ-simplified curve P^μ for some convenient μ, then the number of zeroes in $M^{P^\mu\times Q}_{\rho^*}$ is bounded.

In Section 6.2, we show a data structure that stores P to answer A-decision(Q, ε, ρ). We show how to cleverly navigate $M^{P^\mu\times Q}_{\rho^*}$ for conveniently chosen μ and ρ^*. The key insight in this new technique, is that we may steadily increase ρ^* whilst navigating the matrix. Finally, we extend this solution to answer A-value(Q, ε).

6.1 Perceived free space matrix and free space complexity

We define the perceived free space matrix to help answer A-decision queries. Given two curves (A, Q) and some ρ, we construct an $|A| \times |Q|$ matrix which we call the perceived free space matrix $M^{A\times Q}_{\rho}$. The i'th column corresponds to the i'th element $A[i]$ in A. We assign to each matrix cell $M^{P\times Q}_{\rho}[i, j]$ the integer 0 if $d^\circ(A[i], q_j) \leq \rho$ and integer 1 otherwise.

▶ **Observation 13.** *For all $\rho^* \geq 0$, and curves A and Q, the perceived discrete Fréchet distance $D_{\mathbb{F}}{}^{\circ}(A, Q)$ between A and Q is at most ρ^* if and only if there exists an (xy-monotone) discrete walk F from $(1,1)$ to $(|A|, |Q|)$ where $\forall (i, j) \in F$, $M^{A\times Q}_{\rho^*}[i, j] = 0$.*

Computing $D_{\mathbb{F}}{}^{\circ}(P^\mu, Q)$. Previous results upper bound, for any choice of ρ, the number of zeroes in the FSM between $P^{\varepsilon\rho}$ and $Q^{\varepsilon\rho}$. We instead consider the perceived FSM, and introduce a new parameter $k \geq 1$ to enable approximate distance oracles. For any value ρ and some simplification value $\mu \geq \frac{\varepsilon\rho}{k}$, we upper bound the number of zeroes in the perceived FSM $M^{P^\mu\times Q}_{\rho^*}$ for the conveniently chosen $\rho^* = (1 + \frac{\varepsilon}{2})\rho$:

▶ **Lemma 14.** *Let $P = (p_1, \ldots, p_n)$ be a c-packed discrete curve in $\mathcal{X}$. For any $\rho > 0$ and $0 < \varepsilon < 1$, denote $\rho^* = (1 + \frac{\varepsilon}{2})\rho$. For any $k \geq 1$, denote by P^μ its μ-simplified curve for $\mu \geq \frac{\varepsilon\rho}{k}$. For any curve $Q = (q_1, \ldots, q_m) \subset \mathcal{X}$ the matrix $M^{P^\mu\times Q}_{\rho^*}$ contains at most $8 \cdot \frac{c\cdot k}{\varepsilon}$ zeroes per row.*

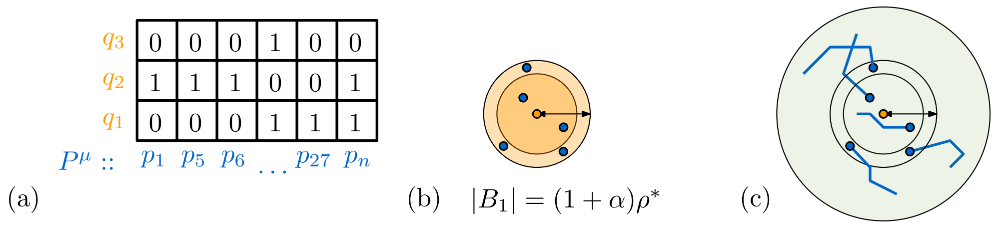

Figure 1 (a) For some value of μ, the μ-simplified curve is $(p_1, p_5, p_6, \ldots, p_{27}, p_n)$. We show the matrix $M_{\rho^*}^{P^\mu \times Q}$. (b) For the point q_3, we claim that there are more than $Z = 8\frac{ck}{\varepsilon}$ zeroes in its corresponding row. Thus, the ball B_1 with radius $(1+\alpha)\rho^*$ contains more than Z points. (c) For each of these points, there is a unique segment along P contained in the ball B_2.

Proof. The proof is by contradiction. Suppose that the j'th row of $M_{\rho^*}^{P^\mu \times Q}$ contains strictly more than $8 \cdot \frac{c \cdot k}{\varepsilon}$ zeroes. Let $P_0 \subset P^\mu$ be the vertices corresponding to these zeroes. Consider the ball B_1 centered at q_j with radius $|B_1| = (1+\alpha)\rho^*$ and the ball B_2 with radius $2|B_1|$ (Figure 1). Each $p_i \in P_0$ must be contained in B_1 and thus $d(p_i, q_j) \leq (1+\alpha)\rho^*$. For each $p_i \in P_0$ denote by S_i the contiguous sequence of vertices of P^μ starting at p_i of length μ. Observe that since $\varepsilon < 1$: $S_i \subset B_2$. Per definition of simplification, each S_i are non-coinciding subcurves. This lower bounds $\ell(P \cap B_2)$:

$$\ell(P \cap B_2) \geq \sum_{p \in P_0} \ell(S_i) = \sum_{p \in P_0} \mu > 2(1+\alpha)(1+\frac{\varepsilon}{2}) \cdot \frac{c \cdot k}{\varepsilon} \cdot \mu \geq 2(1+\alpha) \cdot c \cdot \rho^* \geq c \cdot |B_2|,$$

where $2(1+\alpha)(1+\frac{\varepsilon}{2}) \cdot \frac{c \cdot k}{\varepsilon} < 8 \cdot \frac{c \cdot k}{\varepsilon}$ (since $\alpha < 1$ and $\varepsilon < 1$) – contradicting c-packedness. ◀

6.2 A data structure for answering A-decision(Q, ε, ρ)

We showed in Sections 5 and 6.1 that for any c-packed curve P, $\rho > 0$ and $0 < \varepsilon < 1$ we can choose suitable values $\frac{\varepsilon\rho}{k} \leq \mu \leq \frac{\varepsilon\rho}{6}$ to upper bound the number of zeroes in $M_{\rho^*}^{P^\mu \times Q}$. Moreover, for $\rho^* = (1+\frac{1}{\varepsilon})\rho$ we know that comparing $D_{\mathbb{F}}^\circ(P^\mu, Q) \leq \rho^*$ implies an answer to A-decision(Q, ε, ρ).

We now define a data structure, so that for any μ and any (i, j) we can report the μ-simplification of $P[i, j]$ in $O(|P[i, j]|)$ time. We use this to answer the decision variant.

▶ **Theorem 4.** *Let $\mathcal{X}$ be any geodesic space and $\mathcal{D}_{\mathcal{X}}^\alpha$ be a $(1+\alpha)$-approximate distance oracle with $O(T_\alpha)$ query time. Let $P = (p_1, \ldots, p_n)$ be any c-packed curve in $\mathcal{X}$. We can store P using $O(n)$ space and preprocessing, such that for any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$ and any $\rho > 0$ and $0 < \varepsilon < 1$, we can answer A-decision(Q, ε, ρ) for the discrete Fréchet distance in:*

$$O\left(\frac{c \cdot m}{\varepsilon} \cdot (T_{\varepsilon/6} + \log n)\right) \text{ time.}$$

Proof. We store P in the data structure of Definition 9 using $O(n)$ space and preprocessing time. Given a query A-decision(Q, ε, ρ) we choose $\alpha = \frac{1}{6}\varepsilon$, $\rho^* = (1+\frac{1}{2}\varepsilon)$ and $\mu = \frac{\varepsilon\rho}{6}$. We test if $D_{\mathcal{F}}^\circ(P^\mu, Q) \leq \rho^*$. By Lemma 3, if $D_{\mathbb{F}}^\circ(P^\mu, Q) \leq \rho^*$ then $D_{\mathbb{F}}(P, Q) \leq (1+\varepsilon)\rho$ and otherwise $D_{\mathbb{F}}(P, Q) > \rho$. We consider the matrix $M_{\rho^*}^{P^\mu \times Q}$.

By Observation 13, $D_{\mathbb{F}}^\circ(P^\mu, Q) \leq \rho^*$ if and only if there exists a discrete walk F from $(1, 1)$ to $(|P^\mu|, |Q|)$ where for each $(i, j) \in F$: $M_{\rho^*}^{P^\mu \times Q}[i, j] = 0$. We will traverse this matrix in a depth-first manner as follows: starting from the cell $(1, 1)$, we test if $M_{\rho^*}^{P^\mu \times Q}[1, 1] = 0$. If so, we push $(1, 1)$ onto a stack. Each time we pop a tuple (i, j) from the stack, we inspect their $O(1)$ neighbors $\{(i+1, j), (i, j+1), (i+1, j+1)\}$. If $M_{\rho^*}^{P^\mu \times Q}[i', j'] = 0$, we push (i', j')

onto our stack. It takes $O(\log n)$ time to obtain the $i+1$'th vertex of P^μ, and $O(T_{\varepsilon/6})$ to determine the value of e.g., $M_{\rho^*}^{P^\mu \times Q}[i+1, j]$. Thus each time we pop the stack, we spend $O((T_{\varepsilon/6}) + \log n)$ time.

By Lemma 14 (noting $\varepsilon < 1$ and setting $k = 6$), we push at most $O(\frac{cm}{\varepsilon})$ tuples onto our stack. Therefore, we spend $O(\frac{cm}{\varepsilon}(T_{\varepsilon/6} + \log n))$ total time. By Observation 13, $D_{\mathbb{F}}^\circ(P^\mu, Q) \leq \rho^*$ if and only if we push $(|P^\mu|, |Q|)$ onto our stack. We test this in $O(1)$ additional time per operation. Thus, the theorem follows. ◀

6.3 A data structure for answering A-value(Q, ε)

Finally, we show how to answer the A-value(Q, ε, ρ) query. At this point, we could immediately apply Theorem 2 to answer A-value(Q, ε) at the cost of a factor $O(\log n + \log \varepsilon^{-1})$. However, for the discrete Fréchet distance we show that the factor $O(\log \varepsilon^{-1})$ can be avoided. To this end, we leverage the variable $k \geq 1$ introduced in the definition of $\mu \geq \frac{\varepsilon\rho}{k}$:

▶ **Theorem 5.** *Let $\mathcal{X}$ be a geodesic metric space and $\mathcal{D}_{\mathcal{X}}^\alpha$ be a $(1+\alpha)$-approximate distance with $O(T_\alpha)$ query time. Let $P = (p_1, \ldots, p_n)$ be any c-packed curve in $\mathcal{X}$. We can store P using $O(n)$ space and $O(n \log n)$ preprocessing time, such that for any curve $Q = (q_1, \ldots, q_m)$ in $\mathcal{X}$ and any $0 < \varepsilon < 1$, we can answer A-Value(Q, ε) for the discrete Fréchet distance in:*

$$O\left(\frac{c \cdot m}{\varepsilon} \cdot \log n \cdot \left(T_{\varepsilon/6} + \log \frac{c \cdot m}{\varepsilon} + \log n\right)\right) \text{ time.}$$

Proof. We preprocess P using Lemma 10 in $O(n)$ space and time. We store P in the data structure of Definition 11. This way, we obtain $T = \text{TADD}(\Lambda, \Lambda)$ where Λ is the curve P mapped to $\mathbb{R}^1$. We denote for all $s \in T$ by $I_s = [c_s, 2 \cdot c_s]$ the corresponding interval and obtain a sorted set of intervals $\mathcal{I} = \{I_s\}$.

Given a query (Q, ε), we set $\alpha \leftarrow \varepsilon/6$ and obtain $\mathcal{D}_{\mathcal{X}}^\alpha$. We (implicitly) rescale each interval $I_i \in \mathcal{I}$ by a factor $\frac{6}{\varepsilon}$, creating for I_s the interval $I_s^\varepsilon = [\frac{6 \cdot c_s}{\varepsilon}, \frac{12 \cdot c_s}{\varepsilon}]$. This creates a sorted set $\mathcal{I}^\varepsilon$ of pairwise disjoint intervals. Intuitively, these are the intervals over $\mathbb{R}^1$ where for $\rho \in I_s^\varepsilon$, the μ-simplification P^μ for $\mu = \frac{\varepsilon\rho}{6}$ may change.

We binary search over $\mathcal{I}^\varepsilon$. For each boundary point λ of an interval I_s^ε we query A-decision$(Q, \varepsilon, \lambda)$: discarding half of the remaining intervals in $\mathcal{I}^\varepsilon$. It follows that in $O(\frac{c \cdot m}{\varepsilon} \cdot \log n \cdot (T_{\varepsilon/6} + \log n))$ time, we obtain one of two things:

a) an interval I_s^ε where $\exists \rho^* \in I_s^\varepsilon$ that is a $(1+\varepsilon)$-approximation of $D_{\mathbb{F}}(P, Q)$, or

b) a maximal interval I^* disjoint of the intervals in $\mathcal{I}^\varepsilon$ where $\exists \rho^* \in I^*$ that is a $(1+\varepsilon)$-approximation of $D_{\mathbb{F}}(P, Q)$.

Denote by λ the left boundary of I_s^ε or I^*: it lower bounds $D_{\mathbb{F}}(P, Q)$. Note that if I^* precedes all of $\mathcal{I}^\varepsilon$, $\lambda = 0$. We now compute a $(1+\varepsilon)$-approximation of $D_{\mathbb{F}}(P, Q)$ as follows:

FindApproximation(λ).

1. Compute $C = \frac{d^\circ(p_1, q_1)}{(1 + \frac{1}{2}\varepsilon)}$.
2. Initialize $\rho^* \leftarrow (1 + \frac{1}{2}\varepsilon) \cdot \max\{C, \lambda\}$ and set a constant $\mu \leftarrow \frac{\varepsilon}{6} \cdot \lambda$.
3. Push the lattice point $(1, 1)$ onto a stack.
4. Whilst the stack is not empty do:
   - Pop a point (i, j) and consider the $O(1)$ neighbors (p_a, q_b) of (p_i, q_j) in $M_{\rho^*}^{P^\mu \times Q}$:
     - If $d^\circ(p_a, q_b) \leq \rho^*$, push (a, b) onto the stack.
     - Else, store $d^\circ(p_a, q_b)$ in a min-heap.
   - If we push (p_n, q_m) onto the stack do:
     - Output $\nu = \frac{\rho^*}{(1 + \frac{1}{2}\varepsilon)}$.
5. If the stack is empty, we extract the minimal $d^\circ(p_a, p_b)$ from the min-heap.
   - Update $\rho^* \leftarrow (1 + \frac{1}{2}\varepsilon) \cdot d^\circ(p_a, q_b)$, push (a, b) onto the stack and go to line 4.

Correctness. Suppose that our algorithm pushes (p_n, q_m) onto the stack and let at this time of the algorithm, $\rho^* = (1 + \frac{1}{2}\varepsilon)\nu$. Per definition of the algorithm, $\nu \geq \lambda$ is the minimal value for which the matrix $M_{\rho^*}^{P^\mu \times Q}$ contains a walk F from $(1, 1)$ to (n, m) where for each $(i, j) \in F$: $M_{\rho^*}^{P^\mu \times Q}[i, j] = 0$. Indeed, each time we increment ρ^* by the minimal value required to extend any walk in $M_{\rho^*}^{P^\mu \times Q}$. Moreover, we fixed $\mu \leftarrow \frac{\varepsilon}{6}\lambda$ and thus $\mu \leq \frac{\varepsilon}{6}\nu$. Thus we may apply Lemma 3 to defer that ν is the minimal value for which $D_{\mathbb{F}}(P, Q) \leq (1 + \varepsilon)\nu$.

Running time. We established that the binary search over $\mathcal{I}^\varepsilon$ took $O(\frac{c \cdot m}{\varepsilon} \cdot \log n \cdot (T_{\varepsilon/6} + \log n))$ time. We upper bound the running time of our final routine. For each pair (p_i, q_j) that we push onto the stack we spend at most $O(T_{\varepsilon/6} + \log \frac{c \cdot m}{\varepsilon} + \log n)$ time as we:

- Obtain the $O(1)$ neighbors of (p_i, q_j) through our data structure in $O(\log n)$ time,
- Perform $O(1)$ distance oracle queries in $O(T_{\varepsilon/6})$ time, and
- Possibly insert $O(1)$ neighbors into a min-heap. The min-heap has size at most K: the number of elements we push onto the stack. Thus, this takes $O(\log K)$ insertion time.

What remains is to upper bound the number of items we push onto the stack. Note that we only push an element onto the stack, if for the current value ρ^* the matrix $M_{\rho^*}^{P^\mu \times Q}$ contains a zero in the corresponding cell. We now refer to our earlier case distinction.

Case (a): Since $\varepsilon < 1$ we know that $\rho^* \in [\lambda, 4 \cdot \lambda]$. We set $\mu = \frac{\varepsilon}{6}\lambda$. So $\mu \geq \frac{1}{k}\varepsilon\rho^*$ for $k = 24$. Thus, we may immediately apply Lemma 14 to conclude that we push at most $O(\frac{c \cdot m}{\varepsilon})$ elements onto the stack.

Case (b): Denote by $\gamma = \frac{\varepsilon}{6}\nu$. Per definition of our re-scaled intervals, the open interval (μ, γ) does not intersect with any interval in the non-scaled set $\mathcal{I}$. It follows that $P^\mu = P^\gamma$ and that for two consecutive vertices $p_i, p_l \in P^\mu$: $\ell(P[i, l]) > \gamma$. From here, we essentially redo Lemma 14 for this highly specialized setting. The proof is by contradiction, where we assume that for $\rho^* = (1 + \frac{\varepsilon}{2})\nu$ there are more than $8 \cdot 6 \cdot \frac{c}{\varepsilon}$ zeroes in the j'th row of $M_{\rho^*}^{P^\mu \times Q}$. Denote by $P_0 \subset P^\mu$ the vertices corresponding to these zeroes. We construct a ball B_1 centered at q_j with radius $2\rho^*$ and a ball B_2 with radius $2|B_1|$. We construct a subcurve S_i of P starting at $p_i \in P_0$ of length γ. The critical observation is, that our above analysis implies that all the subcurves S_i do not coincide (since each of them start with a vertex in P^μ). Since $\varepsilon < 1$, each segment S_i is contained in B_2. However, this implies that B_2 is not c-packed since: $\ell(P \cap B_2) \geq \sum_i \ell(S_i) = \sum_i \gamma > 8 \cdot 6 \frac{c}{\varepsilon}\gamma \geq 4 \cdot c \cdot \rho^* \geq 2 \cdot c \cdot |B_2|$. Thus, we always push at most $O(\frac{c \cdot m}{\varepsilon})$ elements onto our stack and this implies our running time. ◀

References

1. Pankaj K Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014. doi:10.1137/130920526.
2. Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995. doi:10.1142/S0218195995000064.
3. Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2004. doi:10.1007/S00453-003-1042-5.
4. Boris Aronov, Omrit Filtser, Michael Horton, Matthew J. Katz, and Khadijeh Sheikhan. Efficient nearest-neighbor query and clustering of planar curves. In Zachary Friggstad, Jörg-Rüdiger Sack, and Mohammad R Salavatipour, editors, *Algorithms and Data Structures*, pages 28–42, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-24766-9_3.
5. Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In Yossi Azar and Thomas Erlebach, editors, *Algorithms - ESA 2006, 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006, Proceedings*, volume 4168 of *Lecture Notes in Computer Science*, pages 52–63. Springer, 2006. doi:10.1007/11841036_8.

6 Alessandro Bombelli, Lluis Soler, Eric Trumbauer, and Kenneth D Mease. Strategic air traffic planning with Fréchet distance aggregation and rerouting. *Journal of Guidance, Control, and Dynamics*, 40(5):1117–1129, 2017.

7 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proceedings of the 31st international conference on Very large data bases*, pages 853–864, 2005. URL: http://www.vldb.org/archives/website/2005/program/paper/fri/p853-brakatsoulas.pdf.

8 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 661–670. IEEE, 2014. doi:10.1109/FOCS.2014.76.

9 Karl Bringmann, Anne Driemel, André Nusser, and Ioannis Psarros. Tight bounds for approximate near neighbor searching for time series under the fréchet distance. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 517–550. SIAM, 2022. doi:10.1137/1.9781611977073.25.

10 Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds. *International Journal of Computational Geometry & Applications*, 27(01n02):85–119, 2017. doi:10.1142/S0218195917600056.

11 Kevin Buchin, Maike Buchin, David Duran, Brittany Terese Fasy, Roel Jacobs, Vera Sacristan, Rodrigo I Silveira, Frank Staals, and Carola Wenk. Clustering trajectories for map construction. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017. doi:10.1145/3139958.3139964.

12 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four Soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discrete & Computational Geometry*, 58(1):180–216, 2017. doi:10.1007/S00454-017-9878-7.

13 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Seth says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2887–2901. SIAM, 2019. doi:10.1137/1.9781611975482.179.

14 Maike Buchin, Bernhard Kilgus, and Andrea Kölzsch. Group diagrams for representing trajectories. *International Journal of Geographical Information Science*, 34(12):2401–2433, 2020. doi:10.1080/13658816.2019.1684498.

15 Maike Buchin, Ivor van der Hoog, Tim Ophelders, Lena Schlipf, Rodrigo I Silveira, and Frank Staals. Efficient Fréchet distance queries for segments. *European Symposium on Algorithms*, 2022.

16 Daniel Chen, Anne Driemel, Leonidas J. Guibas, Andy Nguyen, and Carola Wenk. Approximate map matching with respect to the fréchet distance. In Matthias Müller-Hannemann and Renato Fonseca F. Werneck, editors, *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments, ALENEX 2011, Holiday Inn San Francisco Golden Gateway, San Francisco, California, USA, January 22, 2011*, pages 75–83. SIAM, 2011. doi:10.1137/1.9781611972917.8.

17 Jacobus Conradi, Anne Driemel, and Benedikt Kolbe. Revisiting the fr\'echet distance between piecewise smooth curves. *arXiv preprint arXiv:2401.03339*, 2024. doi:10.48550/arXiv.2401.03339.

18 Mark De Berg, Atlas F Cook IV, and Joachim Gudmundsson. Fast Fréchet queries. *Computational Geometry*, 46(6):747–755, 2013. doi:10.1016/J.COMGEO.2012.11.006.

19 Mark de Berg, Ali D Mehrabi, and Tim Ophelders. Data structures for Fréchet queries in trajectory data. In *29th Canadian Conference on Computational Geometry (CCCG'17)*, pages 214–219, 2017.

20 Thomas Devogele. A new merging process for data integration based on the discrete Fréchet distance. In *Advances in spatial data handling*, pages 167–181. Springer, 2002.

21 Anne Driemel and Sariel Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013. doi:10.1137/120865112.

22 Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discret. Comput. Geom.*, 48(1):94–127, 2012. doi:10.1007/s00454-012-9402-z.

23 Anne Driemel and Ioannis Psarros. $(2+\epsilon)$-ANN for time series under the Fréchet distance. *Workshop on Algorithms and Data structures (WADS)*, 2021.

24 Anne Driemel, Ioannis Psarros, and Melanie Schmidt. Sublinear data structures for short Fréchet queries. *CoRR*, abs/1907.04420, 2019. arXiv:1907.04420.

25 Anne Driemel, Ivor van der Hoog, and Eva Rotenberg. On the discrete Fréchet distance in a graph. In *International Symposium on Computational Geometry (SoCG 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

26 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

27 Arnold Filtser and Omrit Filtser. Static and streaming data structures for fréchet distance queries. In Dániel Marx, editor, *Symposium on Discrete Algorithms (SODA) 2021*, pages 1150–1170. SIAM, 2021. doi:10.1137/1.9781611976465.71.

28 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate nearest neighbor for curves – simple, efficient, and deterministic. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:19, 2020. doi:10.4230/LIPIcs.ICALP.2020.48.

29 Omrit Filtser. Universal approximate simplification under the discrete fréchet distance. *Inf. Process. Lett.*, 132:22–27, 2018. doi:10.1016/j.ipl.2017.10.002.

30 Joachim Gudmundsson, Martin P. Seybold, and Sampson Wong. Map matching queries on realistic input graphs under the fréchet distance. *Symposium on Discrete Algorithms (SODA)*, 2023.

31 Joachim Gudmundsson, André van Renssen, Zeinab Saeidi, and Sampson Wong. Fréchet distance queries in trajectory data. In *The Third Iranian Conference on Computational Geometry (ICCG 2020)*, pages 29–32, 2020.

32 Leonidas J Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. In *Symposium on Computational geometry (SoCG)*, 1987.

33 Sariel Har-Peled. *Geometric approximation algorithms*, volume 173 of *Mathematical Surveys and Monographs*. American Mathematical Soc., 2011.

34 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure–structure alignment with discrete Fréchet distance. *Journal of bioinformatics and computational biology*, 6(01):51–64, 2008. doi:10.1142/S0219720008003278.

35 Maximilian Konzack, Thomas McKetterick, Tim Ophelders, Maike Buchin, Luca Giuggioli, Jed Long, Trisalyn Nelson, Michel A Westenberg, and Kevin Buchin. Visual analytics of delays and interaction in movement data. *International Journal of Geographical Information Science*, 31(2):320–345, 2017. doi:10.1080/13658816.2016.1199806.

36 Yaowei Long and Seth Pettie. Planar distance oracles with better time-space tradeoffs. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2517–2537. SIAM, 2021. doi:10.1137/1.9781611976465.149.

37 Ariane Mascret, Thomas Devogele, Iwan Le Berre, and Alain Hénaff. Coastline matching process based on the discrete Fréchet distance. In *Progress in Spatial Data Handling*, pages 383–400. Springer, 2006.

38 Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983. doi:10.1145/2157.322410.

39 Otfried Schwarzkopf and Jules Vleugels. Range searching in low-density environments. *Inf. Process. Lett.*, 60(3):121–127, 1996. doi:10.1016/S0020-0190(96)00154-8.

40 Roniel S. De Sousa, Azzedine Boukerche, and Antonio A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5), September 2020. doi:10.1145/3406096.

41 E Sriraghavendra, K Karthik, and Chiranjib Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 461–465. IEEE, 2007. doi:10.1109/ICDAR.2007.4378752.

42 Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, 2020. doi:10.1007/S00778-019-00574-9.

43 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004. doi:10.1145/1039488.1039493.

44 Ivor van der Hoog, Eva Rotenberg, and Sampson Wong. Data structures for approximate discrete Fréchet distance. *CoRR*, abs/2212.07124, 2022. doi:10.48550/arXiv.2212.07124.

45 A. Frank van der Stappen. *Motion planning amidst fat obstacles*. University Utrecht, 1994.

46 Rene Van Oostrum and Remco Veltkamp. Parametric search made practical. In *Symposium on Computational Geometry (C)*, pages 1–9, 2002.

47 Dong Xie, Feifei Li, and Jeff M Phillips. Distributed trajectory similarity search. *Proceedings of the VLDB Endowment*, 10(11):1478–1489, 2017. doi:10.14778/3137628.3137655.

48 Daming Xu. *Well-separated pair decompositions for doubling metric spaces*. PhD thesis, Carleton University, 2005.