Fréchet Distance in the Imbalanced Case

Lotte Blank
University of Bonn, Germany

Abstract

Given two polygonal curves P and Q defined by n and m vertices with $m \leq n$, we show that the discrete Fréchet distance in 1D cannot be approximated within a factor of $2-\varepsilon$ in $\mathcal{O}((nm)^{1-\delta})$ time for any $\varepsilon, \delta > 0$ unless OVH fails. Using a similar construction, we extend this bound for curves in 2D under the continuous or discrete Fréchet distance and increase the approximation factor to $1+\sqrt{2}-\varepsilon$ (resp. $3-\varepsilon$) if the curves lie in the Euclidean space (resp. in the L_∞-space). This strengthens the lower bound by Buchin, Ophelders, and Speckmann to the case where $m = n^\alpha$ for $\alpha \in (0,1)$ and increases the approximation factor of 1.001 by Bringmann. For the discrete Fréchet distance in 1D, we provide an approximation algorithm with optimal approximation factor and almost optimal running time. Further, for curves in any dimension embedded in any L_p space, we present a $(3+\varepsilon)$-approximation algorithm for the continuous and discrete Fréchet distance using $\mathcal{O}((n+m^2)\log n)$ time, which almost matches the approximation factor of the lower bound for the L_∞ metric.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, SETH, Orthogonal Vectors, Lower Bounds, distance oracle, data structures

Digital Object Identifier 10.4230/LIPIcs.SoCG.2026.17

Related Version *Full Version*: `https://arxiv.org/abs/2602.09551`

Funding *Lotte Blank*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 459420781 (FOR AlgoForGe).

1 Introduction

The Fréchet distance is a distance metric for polygonal curves, where each curve is represented as a piecewise-linear interpolation between its vertices. There are different variants of this metric and the two most common variants are the discrete and the continuous Fréchet distance. Alt and Godau [3] were the first to study the computational complexity of the Fréchet distance. Let n and m denote the number of vertices of the two curves, which we refer to as their complexities, with $n \geq m$. They showed that the continuous Fréchet distance can be computed in $\mathcal{O}(nm \log n)$ time. Eiter and Mannilla [21] defined the discrete Fréchet distance and gave an $\mathcal{O}(nm)$ time algorithm. Although both results were slightly improved later [15, 11, 1], Bringmann [6] showed that for d-dimensional curves with $d \geq 2$ it is very unlikely that significantly faster algorithms exists: unless the Strong Exponential Time Hypothesis (SETH) fails, no (1.001)-approximation algorithm for the continuous or discrete Fréchet distance can run in $\mathcal{O}((nm)^{1-\delta})$ time for any $\delta > 0$. Later, Bringmann and Mulzer [10] extended this result for the discrete Fréchet distance in 1D and increased the approximation factor up to $1.4-\varepsilon$ for any $\varepsilon > 0$[1]. For the continuous and the discrete Fréchet distance in 1D, Buchin, Ophelders, and Speckmann [12] showed a hardness result for approximation factors up to $3-\varepsilon$, but only in the case where $n = m$. A recent result by Blank and Driemel [5] shows that the continuous Fréchet distance in 1D can indeed be computed faster when $m = n^\alpha$ with $\alpha \in (0,1)$, namely in $\mathcal{O}(n \log n + m^2 \log^2 n)$ time.

[1] In [10], this result is stated only for $n = m$, but the same construction works for the case $n \neq m$ as well.

Related Work. In recent years, many works have studied approximation algorithms for the Fréchet distance [18, 26, 27]. This line of research culminated in the first constant-factor approximation algorithm for the Fréchet distance that runs in subquadratic time, due to Cheng, Huang, and Zhang [17]. Their randomized $(7+\varepsilon)$-approximation algorithm for the continuous Fréchet distance runs in $\mathcal{O}(nm^{0.99}\log(n/\varepsilon))$ time for any $\varepsilon > 0$ and succeeds with probability at least $1-1/n^6$.

A simple idea to 3-approximate the Fréchet distance between curves where $m = n^\alpha$ with $\alpha \in (0,1)$ in subquadratic time is the following. Compute a curve P' of complexity $n' \in o(n)$ and its distance δ to P such that every curve Q of complexity m has Fréchet distance at least δ to P. Then compute the Fréchet distance δ' between P' and Q. Using triangle inequality, one can show that the Fréchet distance between P and Q is at least $\max\{\delta'/2, \delta\}$ and at most $3 \cdot \max\{\delta'/2, \delta\}$, which yields a 3-approximation. The difficulty of this approach is to compute such a curve P'. Two related variants of this simplification problem have been studied. The first is computing an optimal *m-simplification* of P, that is, a curve P^* of complexity at most m minimizing the Fréchet distance to P among all curves of complexity at most m. The second is computing an optimal *δ-simplification*, that is, a curve P^* of minimum complexity among all curves that have Fréchet distance at most δ to P. However, no previous result achieved a subquadratic running time for curves in arbitrary dimension when the approximation factor on the complexity of the simplified curve is constant [2, 4, 7, 13, 16, 22, 28].

A different approach to reduce the running time of computing the Fréchet distance is via Fréchet distance oracles. Such an oracle is a data structure that preprocesses an input curve and subsequently answers Fréchet distance queries with respect to query curves. Driemel and Har-Peled [19] described a data structure of size $\mathcal{O}(n\log n)$ using $\mathcal{O}(m^2\log n\log(m\log n))$ query time, which approximates the continuous Fréchet distance up to a (large) constant factor. For exact queries, [14] presented a data structure of size $\mathcal{O}(nm)^{\mathrm{poly}(d,m)}$ and query time in $\mathcal{O}((md)^{\mathcal{O}(1)}\log(nm))$, where the query complexity m is given at preprocessing time. Blank and Driemel [5] showed that for curves in 1D, there is a data structure of size $\mathcal{O}(n\log n)$ that computes the continuous Fréchet distance exactly in $\mathcal{O}(m^2\log^2 n)$ time. For the discrete Fréchet distance, Filtser and Filtser [22] improved [20] and presented a data structure with size in $\mathcal{O}(1/\varepsilon)^{kd}\log(\varepsilon)^{-1}$ that approximates the discrete Fréchet distance within a factor of $(1+\varepsilon)$ using expected $\widetilde{\mathcal{O}}(kd)$ query time, where $\widetilde{\mathcal{O}}(\cdot)$ hides polylogarithmic factors in n. Gudmundsson, Seybold, and Wong [24] used the lower bound construction by Bringmann [6] to show in a similar fashion to [8, 20, 25] the following. In 2D, there exists no Fréchet distance oracle with $\mathrm{poly}(n)$ preprocessing time and $\mathcal{O}((nm)^{1-\delta})$ query time and approximation factor 1.001 unless SETH fails.

Our Results. Our first set of results shows lower bounds for algorithms that approximate the Fréchet distance conditioned on the Orthogonal Vectors Hypothesis (OVH). We show that the discrete Fréchet distance between 1-dimensional curves cannot be $(2-\varepsilon)$-approximated in $\mathcal{O}((nm)^{1-\delta})$ time for any $\varepsilon, \delta > 0$ unless OVH fails. This increases the approximation factor of $1.4-\varepsilon$ by Bringmann and Mulzer [10]. Using a similar construction in 2D, we show that the continuous and discrete Fréchet distance cannot be approximated within a factor less than $1+\sqrt{2}$ (resp. 3) when the curves lie in the Euclidean space (resp. in the L_∞-space) in $\mathcal{O}((nm)^{1-\delta})$. This leads to lower bounds for Fréchet distance oracles as well and answers an open question by [24].

To obtain this first set of results, we reduce from the *Orthogonal Vector Problem:* Given two sets of vectors $U, V \subset \{0,1\}^d$ of size m and n, do there exist two vectors $u \in U$ and $v \in V$ such that $\langle u, v\rangle = 0$? Then, the results follow by OVH, which is implied by SETH [29, 9].

Table 1 Upper and Lower Bounds for the Fréchet distance Computation.

	Upper Bounds					
	Discrete			Continuous		
1D	$\mathcal{O}(mn)$	exact	[21, 1]	$\widetilde{\mathcal{O}}(mn)$	exact	[3, 11, 15]
	$\widetilde{\mathcal{O}}(n+m^2)$	2-approx.	Cor.20	$\widetilde{\mathcal{O}}(n+m^2)$	exact	[5]
dD	$\mathcal{O}(mn)$	exact	[21, 1]	$\widetilde{\mathcal{O}}(mn)$	exact	[3, 11, 15]
	$\widetilde{\mathcal{O}}(n+m^2)$	$(3+\varepsilon)$-approx.	Thm.23	$\widetilde{\mathcal{O}}(n+m^2)$	$(3+\varepsilon)$-approx.	Thm.23
	$\widetilde{\mathcal{O}}(nm^{0.99})$	$(7+\varepsilon)$-approx.	[17]	$\widetilde{\mathcal{O}}(nm^{0.99})$	$(7+\varepsilon)$-approx.	[17]
	Lower Bounds					
	Discrete			Continuous		
1D	$\nexists\ \mathcal{O}(m^{2-\delta})$	$(3-\varepsilon)$-approx.	[12]	$\nexists\ \mathcal{O}(m^{2-\delta})$	$(3-\varepsilon)$-approx.	[12]
	$\nexists\ \widetilde{\mathcal{O}}((mn)^{1-\delta})$	$(1.4-\varepsilon)$-approx.	[10]			
	$\nexists\ \widetilde{\mathcal{O}}((mn)^{1-\delta})$	$(2-\varepsilon)$-approx.	Thm.9			
2D	$\nexists\ \mathcal{O}((mn)^{1-\delta})$	(1.001)-approx.	[6]			
	$\nexists\ \mathcal{O}((mn)^{1-\delta})$	$(1+\sqrt{2}-\varepsilon)$-approx. in L_2	Thm.10			
	$\nexists\ \mathcal{O}((mn)^{1-\delta})$	$(3-\varepsilon)$-approx. in L_∞	Thm.10			

▶ **Definition 1** (Orthogonal Vectors Hypothesis (OVH)). *For all $\delta > 0$ there exists a $c > 0$ such that there is no algorithm solving OV instances $U, V \subset \{0,1\}^d$ of size m and n with $m \leq n$ and $d = c \log n$ in time $\mathcal{O}((nm)^{1-\delta})$.*

We construct a curve P of complexity $\mathcal{O}(n+m)$ depending on V and the size of U and a curve Q of complexity $\mathcal{O}(m)$ depending only on U. Then, we show that the Fréchet distance between P and Q is at most 1 if and only if the OV instance has a solution. In order to get a lower bound also for the case where $n \gg m$, we construct P such that the main part of P lies in a ball of radius 1. Hence, this main part can be matched to a single point of Q. Further, to obtain an inapproximability result of $2-\varepsilon$ in 1D, the points of P and Q have integer coordinates. Hence, there are only three different values that the vertices of the main part of P can have. This significantly restricts our possibilities to define P and is the main difference to the construction by Bringmann and Mulzer [10]. They also note in their paper that obtaining a lower bound for an approximation factor of 1.4 or higher requires a significantly different construction compared to theirs.

On the positive side, we present a data structure that preprocesses a 1-dimensional curve of complexity n in $\mathcal{O}(n \log n)$ time using $\mathcal{O}(m)$ space that 2-approximates the discrete Fréchet distance to query curves of complexity m in $\mathcal{O}(m^2 \log m)$ time. Hence, there exists a 2-approximation algorithm for the discrete Fréchet distance in $\mathcal{O}(n \log n + m^2 \log m)$ time matching our approximation factor lower bound and essentially the time lower bound by [12]. Secondly, we match our approximation factor lower bound in the L_∞-space by giving a simple $(3+\varepsilon)$-approximation algorithm for the discrete and continuous Fréchet distance using $\mathcal{O}((n+m^2)\log n)$ time for curves in any dimension under any L_p-norm. Table 1 summarizes the results of this paper and related work.

2 Preliminaries

For any two points $p_1, p_2 \in \mathbb{R}^d$, $\overline{p_1p_2}$ denotes the directed line segment connecting p_1 with p_2. A polygonal curve P of complexity n is formed by ordered line segments $\overline{P(i)P(i+1)}$ of points $P(1), P(2), \ldots, P(n)$, where $P(i) \in \mathbb{R}^d$. It can be viewed as a function $P : [1,n] \to \mathbb{R}^d$, where $P(i+\alpha) = (1-\alpha)P(i) + \alpha P(i+1)$ for $i \in \{1, \dots, n\}$ and $\alpha \in [0,1]$. This curve is also denoted with $\langle P(1), \dots, P(n)\rangle$. We call the points $P(i)$ vertices and the line segments $\overline{P(i)P(i+1)}$ edges of P. Further, $P[s,t]$ is the subcurve of P obtained from restricting the domain to the interval $[s,t]$. Two curves $P = \langle P(1), \dots, P(n)\rangle$ and $Q = \langle Q(1), \dots, Q(m)\rangle$ can be composed into the curve $P \circ Q = \langle P(1), \dots, P(n), Q(1), \dots, Q(m)\rangle$. For a number $k \in \mathbb{N}_0$, define P^k be the composition of k copies of P.

Let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves. A *continuous matching* M between P and Q is defined by two functions $h_P \in \mathcal{F}_n$ and $h_Q \in \mathcal{F}_m$ and consists of the tuples $(h_P(a), h_Q(a))$ for $a \in [0,1]$, where $\mathcal{F}_n$ is the set of all continuous, non-decreasing functions $h : [0,1] \to [1,n]$ with $h(0) = 1$ and $h(1) = n$, respectively $\mathcal{F}_m$ for m. Then, the *(continuous) Fréchet distance* between P and Q is defined as

$$d_{\mathrm{F}}(P,Q) = \min_{\text{continuous matching } M} \max_{(x,y)\in M} \|P(x) - Q(y)\|.$$

For a fixed δ, the *free space* $F_\delta(P,Q)$ is a subset of $[1,n] \times [1,m]$ such that $(x,y) \in F_\delta(P,Q)$ if and only if $\|P(x) - Q(y)\| \le \delta$. Points in $([1,n] \times [1,m]) \setminus F_\delta(P,Q)$ are contained in the non-free space. Hence, the continuous Fréchet distance between P and Q is at most δ if a continuous matching $M \subset F_\delta(P,Q)$ between P and Q exists.

A *discrete matching* M between P and Q is an ordered sequence that starts in $(1,1)$, ends in (n,m) and for any consecutive tuples (i,j), (i',j') in M it holds that $(i',j') \in \{(i+1,j), (i,j+1), (i+1,j+1)\}$. Then, the *discrete Fréchet distance* between P and Q is

$$d_{\mathrm{dF}}(P,Q) = \min_{\text{discrete matching } M} \max_{(i,j)\in M} \|P(i) - Q(j)\|.$$

The *free space matrix* M_δ is an $n \times m$ matrix, where $M_\delta[i,j] = 1$ if $|P(i) - Q(j)| \le \delta$ and otherwise $M_\delta[i,j] = 0$. We say that (i,j) is *reachable* if $d_{\mathrm{dF}}(P[1,i], Q[1,j]) \le \delta$.

▶ **Observation 2.** *Let P_1, P_2, Q_1, Q_2 be curves. If $d_F(P_1,Q_1) \le \delta$ and $d_F(P_2,Q_2) \le \delta$, then $d_F(P_1 \circ P_2, Q_1 \circ Q_2) \le \delta$. For the discrete Fréchet distance, the analogue statement holds.*

▶ **Observation 3.** *Let P and Q be polygonal curves, then $d_F(P,Q) \le d_{dF}(P,Q)$.*

3 Approximation lower bounds for the imbalanced case

In this section, we show a lower bound for the discrete Fréchet distance in 1D and for the discrete and continuous Fréchet distance in 2D depending on OVH. The construction for both bounds is very similar and we start with giving a schematic view of the reduction.

Let $U, V \subset \{0,1\}^d$ be two sets of size m and n, where $m \le n$ and denote with v_{ik} (resp. u_{ik}) the k-th bit of the i-th vector of V (resp. U). We construct two 1-dimensional (resp. 2-dimensional) curves P and Q of complexity $\mathcal{O}(nd)$ and $\mathcal{O}(md)$, where P depends on V and Q on U. We show that their discrete (and continuous) Fréchet distance is at most 1 if and only if two vectors that are orthogonal exist in V and U. Define

$$P = P_s \circ (P^*)^{m-1} \circ P_{null} \circ \bigcirc_{i=1}^{n} \left((P_0)^{d+1} \circ P_1 \circ \bigcirc_{\ell=1}^{d} P_{v_{i\ell}} \circ P_1\right) \circ P_{null} \circ (P^*)^{m-1} \circ P_s,$$

$$Q = (Q^*)^{m-1} \circ \bigcirc_{j=1}^{m} \left(Q_{null} \circ Q_c \circ (Q_1)^{d+1} \circ Q_0 \circ \bigcirc_{\ell=1}^{d} Q_{u_{j\ell}} \circ Q_0 \circ Q_c\right) \circ Q_{null} \circ (Q^*)^{m-1},$$

where the subcurves are defined later. Since the complexity of P can be arbitrarily larger than the complexity of Q, we construct the curve P such that the main part of P, i.e., in between the two subcurves P_{null}, is contained in a ball of radius 1. Then, we can match this whole part to a single point on Q. This point is Q_c in our construction. At a high

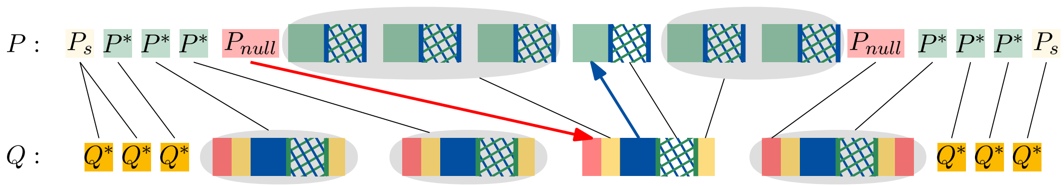

Figure 1 A schematic view of the construction of the curves P and Q. Here, Q_{null} is drawn in red, P_0 and Q_0 in green, P_1 and Q_1 in blue, and Q_c in orange.

level idea, we want to enforce that in a matching that realizes the Fréchet distance, the two subcurves P_{null} get roughly matched to different subcurves Q_{null}. Then, the subcurve in between the two points matched to P_{null} contains $(Q_1)^{d+1}$ as a subcurve. Now, we enforce that $(Q_1)^{d+1}$ gets roughly matched to $(P_0)^{d+1}$. Hence, afterwards $\bigcirc_{k=1}^{d} P_{v_{ik}}$ and $\bigcirc_{\ell=1}^{d} Q_{u_{j\ell}}$ have to be matched together for some i and j. We will show that this is only possible if $\langle v_i, u_j \rangle = 0$. This matching is schematically visualized in Figure 1. We define

$$P_s = \langle(-1,0)\rangle, \qquad P_{null} = \langle(-5,0)\rangle, \qquad P^* = \langle(-3,0)\rangle \circ \langle P_0\rangle^{2d+3} \circ \langle(-3,0)\rangle,$$
$$Q_c = \langle(0,0)\rangle, \qquad Q_{null} = \langle(-4,0)\rangle, \qquad Q^* = \langle(-2,0),(0,0),(-2,0)\rangle,$$

for 2D and the definition carries over directly to 1D. The curves P_0, P_1, Q_0, and Q_1 are visualized in Figure 2. For the 1D setting, let

$$P_0 = \langle -1, 1\rangle, \qquad P_1 = \langle -1, 0\rangle, \qquad Q_0 = \langle -2, 1\rangle, \qquad Q_1 = \langle -2, 2\rangle.$$

For the 2D constructions, define eight points with distance 1 to $(0,0)$ where $a_1 = (-1,0)$ and the direction of a_{i+1} is the direction of a_i rotated by $\pi/4$, i.e., in L_2

$$a_1 = (-1,0), \qquad a_2 = (-1/\sqrt{2}, -1/\sqrt{2}), \qquad a_3 = (0,-1), \qquad a_4 = (1/\sqrt{2}, -1/\sqrt{2}),$$
$$a_5 = (1,0), \qquad a_6 = (1/\sqrt{2}, 1/\sqrt{2}), \qquad a_7 = (0,1), \qquad a_8 = (-1/\sqrt{2}, 1/\sqrt{2}),$$

and in L_∞

$$a_1 = (-1,0), \qquad a_2 = (-1,-1), \qquad a_3 = (0,-1), \qquad a_4 = (1,-1),$$
$$a_5 = (1,0), \qquad a_6 = (1,1), \qquad a_7 = (0,1), \qquad a_8 = (-1,1).$$

Define $c = \sqrt{2}$ and $\tilde{a} = a_8 + a_7$ in L_2 and $c = 2$ and $\tilde{a} = 2a_8$ in L_∞. Note that c is defined such that ca_1 is the point with distance 1 to a_8 and a_2 that maximizes the distance to $(0,0)$ and $\tilde{a}$ such that $\|\tilde{a} - a_2\| = 1 + c$ and $\|\tilde{a} - a_8\| = 1$. Then we set

$$P_0 = \langle a_1, a_2, \ldots a_7, a_8, a_7 \ldots, a_1\rangle, \qquad P_1 = \langle a_1, a_2, \ldots, a_8, a_1, a_2, a_1 a_8, \ldots, a_1\rangle,$$
$$Q_0 = \langle 2a_1, 2a_2, \ldots, 2a_8, ca_1, 2a_8, \ldots, 2a_1\rangle, \qquad Q_1 = \langle 2a_1, 2a_2, \ldots, 2a_7, \tilde{a}, 2a_7, \ldots, 2a_1\rangle.$$

We show that this results in an inapproximability result of $(1+c-\varepsilon)$ for time in $\mathcal{O}((nm)^{1-\delta})$.

▶ **Lemma 4.** *If there exist indices i and j such that $\langle v_i, u_j\rangle = 0$, then in every construction for P and Q above it holds that $d_{dF}(P,Q) \le 1$.*

Proof. For every construction of P and Q, it holds that $d_{\mathrm{dF}}(P^*, Q^*) \le 1$, $d_{\mathrm{dF}}(P_s, Q^*) \le 1$. Further, for $x, y \in \{0,1\}$ with $\langle x, y\rangle = 0$, it holds that $d_{\mathrm{dF}}(P_x, Q_y) \le 1$. Then, we can match the following with Fréchet distance at most 1 and the lemma follows by Observation 2

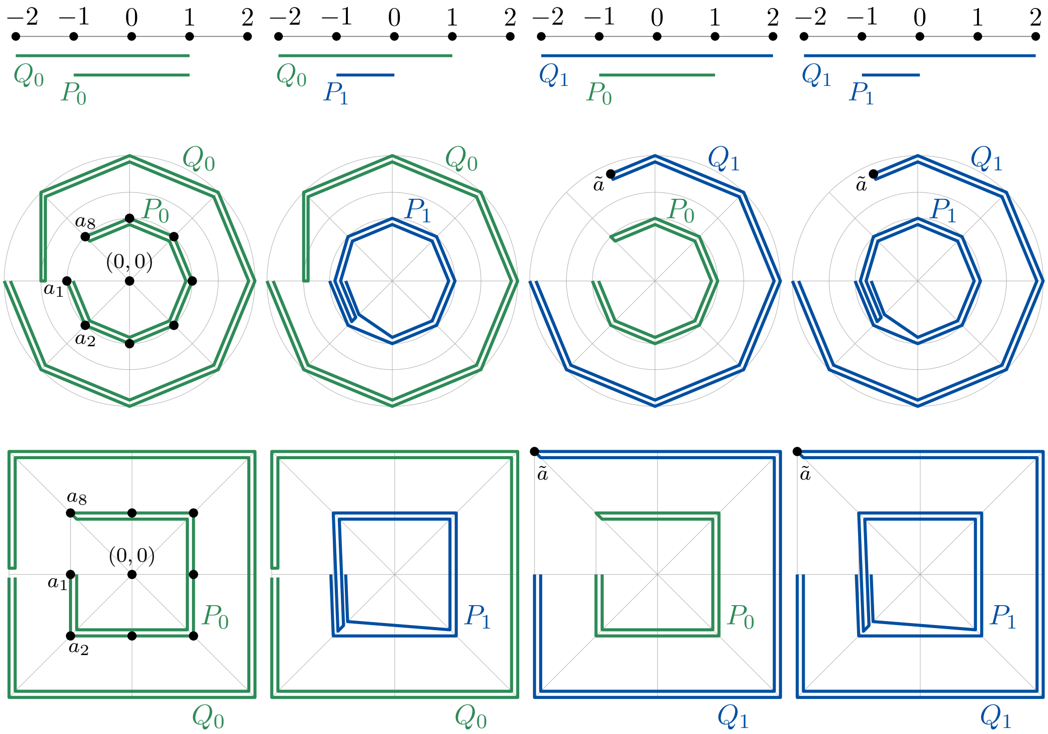

Figure 2 The curves P_0, P_1, Q_0, and Q_1. The upper curves are defined in 1D, the middle curves for the L_2-metric in 2D and the lower curves for the L_∞-metric in 2D.

- $P_s \circ (P^*)^{m-j}$ and $(Q^*)^{m-1}$,
- $(P^*)^{j-1} \circ P_{null}$ and $\bigcirc_{k=1}^{j-1} (Q_{null} \circ Q_c \circ (Q_1)^{d+1} \circ Q_0 \circ \bigcirc_{\ell=1}^{d} Q_{u_{k\ell}} \circ Q_0 \circ Q_c) \circ Q_{null}$,
- $\bigcirc_{k=1}^{i-1} \left((P_0)^{d+1} \circ P_1 \circ \bigcirc_{\ell=1}^{d} P_{v_{k\ell}} \circ P_1\right)$ and Q_c,
- $(P_0)^{d+1} \circ P_1 \circ \bigcirc_{\ell=1}^{d} P_{v_{i\ell}} \circ P_1$ and $(Q_1)^{d+1} \circ Q_0 \circ \bigcirc_{\ell=1}^{d} Q_{u_{j\ell}} \circ Q_0$,
- $\bigcirc_{k=i+1}^{n} \left((P_0)^{d+1} \circ P_1 \circ \bigcirc_{\ell=1}^{d} P_{v_{k\ell}} \circ P_1\right)$ and Q_c,
- $P_{null} \circ (P^*)^{m-j}$ and $\bigcirc_{k=j+1}^{m} (Q_{null} \circ Q_c \circ (Q_1)^{d+1} \circ Q_0 \circ \bigcirc_{\ell=1}^{d} Q_{u_{k\ell}} \circ Q_0 \circ Q_c) \circ Q_{null}$,
- $(P^*)^{j-1} \circ P_s$ and $(Q^*)^{m-1}$. ◀

To show the other direction, we first prove some lemmas.

▶ **Lemma 5.** *Consider the 1D setting and let $Q' = \bigcirc_{\ell=1}^{d'} Q_{\sigma_\ell}$, where $\sigma_\ell \in \{0,1\}$ for every ℓ. Further, let $s \leq t$ be such that $P[s,t]$ is a subcurve in between the two subcurves P_{null} and $d_{dF}(Q', P[s,t]) < 2$. Then $P[s,t] = \bigcirc_{\ell=1}^{d'} P_{\pi_\ell}$ where $\pi_\ell = 0$ if $\sigma_\ell = 1$ and $\pi_\ell \in \{0,1\}$ if $\sigma_\ell = 0$ for every ℓ.*

Proof. It holds that every second vertex of Q' is -2 and the vertices of value -1 on $P[s,t]$ are the only vertices on $P[s,t]$ within distance less than 2 to those vertices of Q'. Further, it holds that the vertices with value -1 of $P[s,t]$ have distance less than two only to the vertices of value -2 of Q'. Hence, since the vertices of value -1 on $P[s,t]$ are exactly every second vertex on $P[s,t]$, we must traverse $P[s,t]$ and Q' synchronously. Therefore, if $\sigma_\ell = 0$ then $\pi_\ell \in \{0,1\}$ and if $\sigma_\ell = 1$ then $\pi_\ell = 0$ by construction. ◀

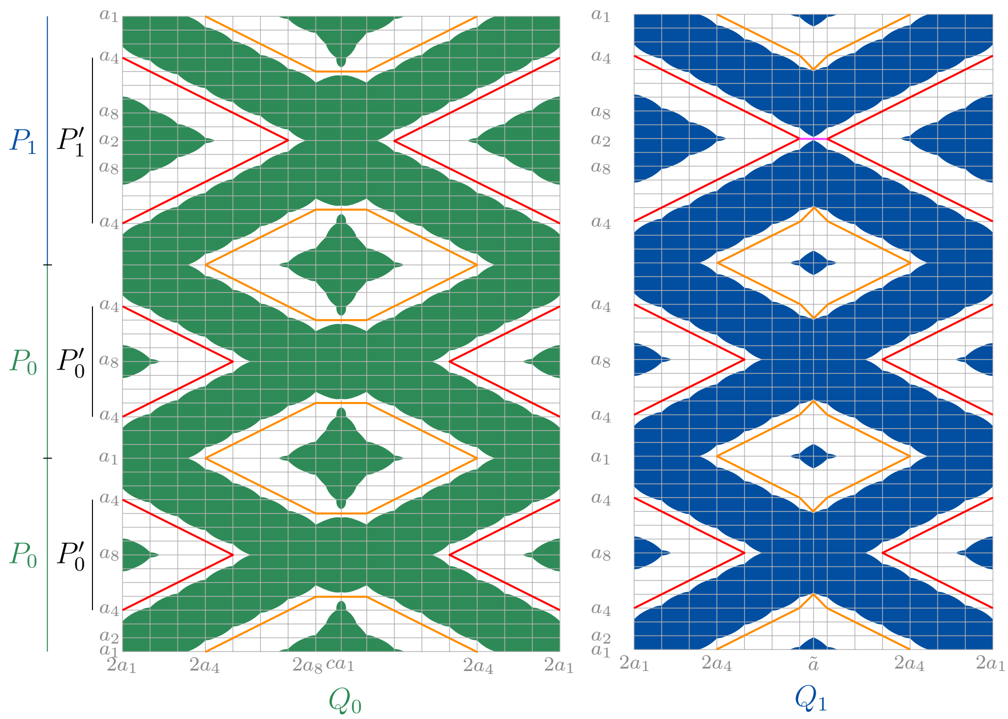

Figure 3 In white: the non-free space of the curve Q_0 and $P_0 \circ P_0 \circ P_1$ on the left and of the curve Q_1 and $P_0 \circ P_0 \circ P_1$ on the right. The red, orange and pink paths visualize the proof of Lemma 6.

In 2D, we also want to show a bound for the continuous Fréchet distance. Then the vertices of Q can also be matched to points on edges of P. We use the notation of the following subcurve of P_0 (resp. P_1): $P'_0 = \langle a_4, a_5, \ldots, a_8, \ldots, a_4 \rangle$ (resp. $P'_1 = \langle a_4, \ldots, a_8, a_1, a_2, a_1, a_8, \ldots, a_4 \rangle$).

▶ **Lemma 6.** *Consider the 2D setting. For any subcurve P' of P in between the two subcurves P_{null} with $d_F(P', Q_y) < 1 + c$ for $y \in \{0, 1\}$, it holds that P'_x is a subcurve of P' and P' is a subcurve of $\langle a_4, a_3, a_2 \rangle \circ P_x \circ \langle a_2, a_3, a_4 \rangle$, where $x = 0$ if $y = 1$ and $x \in \{0, 1\}$ if $y = 0$.*

Proof. We show this lemma by constructing paths in the non-free space for $\delta = 1 + c$ (see Figure 3). First observe that $\|2a_i - a_j\| \geq 1 + c$ for any $i, j \in \{1, \ldots, 8\}$ with $i - j \geq 3 \mod 8$.

Assume that the first (resp. last) vertex of Q_y is matched to a point on a subcurve P'_x with $x \in \{0, 1\}$. Then, there must exist a path completely contained in the free space ending at the last (resp. first) vertex of Q_y. This cannot be the case since there exists a continuous matching between P'_x and $\langle 2a_1, 2a_2 \ldots, 2a_5, \ldots, 2a_1 \rangle$ that is completely contained in the non-free space (see the red paths in Figure 3). Therefore, the first as well as the last vertex of Q_y has to be matched to a point on a subcurve P'' of P such that $P'' = \langle a_4, a_3, a_2, a_1, a_2, a_3, a_4 \rangle$.

Next, we show that P' contains exactly one P'_x as a subcurve. Since $\|ca_1 - a_5\| \geq 1 + c$ (resp. $\|\tilde{a} - a_5\| \geq 1 + c$), there exists a path that is completely contained in the non-free space between $\langle a_5, a_4, \ldots a_2, a_1, a_2, \ldots, a_4, a_5 \rangle$ and $\langle 2a_8, ca_1, 2a_8, 2a_7, \ldots, 2a_4, 2a_5, \ldots, 2a_8, ca_1, 2a_8 \rangle$(resp. $\langle \tilde{a}, 2a_7, \ldots, 2a_4, 2a_5, \ldots, \tilde{a} \rangle$) (see the orange paths in Figure 3). Hence by construction of P, it follows that P'_x is a subcurve of P' and P' is a subcurve of $\langle a_4, a_3, a_2 \rangle \circ P_x \circ \langle a_2, a_3, a_4 \rangle$ with $x \in \{0, 1\}$. It remains to show that $x \neq 1$, if $y = 1$. This follows since the distance between a_2 and $2a_7$, $\tilde{a}$, $2a_7$ is at least $1 + c$ (see the pink path in Figure 3). ◀

The next lemma follows directly by the construction of the curves and Lemma 6.

▶ **Lemma 7.** *Consider the 2D setting and let* $Q' = \bigcirc_{\ell=1}^{d'} Q_{\sigma_\ell}$*, where* $\sigma_\ell \in \{0,1\}$ *for every* ℓ*. Further, let* $s \leq t$ *be such that* $P[s,t]$ *is a subcurve in between the two subcurves* P_{null} *and* $d_F(Q', P[s,t]) < 1 + c$*. Then,* $P'_{\pi_1} \circ \bigcirc_{\ell=2}^{d'-1} P_{\pi_\ell} \circ P'_{\pi_{d'}}$ *is a subcurve of* $P[s,t]$ *and* $P[s,t]$ *is a subcurve of* $\langle a_4, a_3, a_2 \rangle \circ \bigcirc_{\ell=1}^{d'} P_{\pi_\ell} \circ \langle a_2, a_3, a_4 \rangle$ *where* $\pi_\ell = 0$ *if* $\sigma_\ell = 1$ *and* $\pi_\ell \in \{0,1\}$ *if* $\sigma_\ell = 0$ *for every* ℓ*.*

▶ **Lemma 8.** *If* $d_{dF}(P,Q) < 2$ *(resp.* $d_F(P,Q) < 1 + c$*) in the 1D (resp. 2D) construction, then there exist two indices* i *and* j *such that* $\langle v_i, u_j \rangle = 0$*.*

Proof. Let M be a discrete (resp. continuous) matching that realizes the discrete (resp. continuous) Fréchet distance between P and Q. Assume that $d_{\mathrm{F}}(P,Q) < 2$ (resp. $d_{\mathrm{dF}}(P,Q) < 1 + c$), then in the matching M, the two points of value P_{null} of P must be matched to two distinct points q_1, q_2 on Q both on an edge that contains the value Q_{null}, since they are the only points within distance at most $1 + c \leq 3$ to P_{null}. Let $P'_0 = P_0$ and $P'_1 = P_1$ in the 1D setting. Define j such that q_1 is contained on an incident edge to the j-th Q_{null} on Q. Then, the first $(Q_1)^{d+1}$ after the j-th Q_{null} on Q must be matched to a subcurve P' of P such that $P'_1 \circ (P_0)^{d+1} \circ P'_1$ is a subcurve of P' by Lemma 5 and Lemma 7. The only time we have $(P_0)^{d+1}$ in P is before $P_1 \circ \bigcirc_{\ell=1}^{d} P_{v_{i\ell}}$ for an i. Hence by Lemma 5 and Lemma 7, there exists an index i such that in the matching M the subcurve $P'_{v_{i,1}} \circ \bigcirc_{\ell=2}^{d-1} P_{v_{i,\ell}} \circ P'_{v_{i,d}}$ is matched to a subcurve of $\bigcirc_{\ell=1}^{d} Q_{u_{j,\ell}}$ and $v_{i,\ell} = 0$ if $u_{j,\ell} = 1$. Hence, it holds that $\langle v_i, u_j \rangle = 0$. ◀

By Lemma 4 and Lemma 8 together with Observation 3, the next two theorems follow.

▶ **Theorem 9.** *For any* $\delta, \varepsilon > 0$*, there does not exist a* $(2 - \varepsilon)$*-approximation algorithm for the discrete Fréchet distance between 1-dimensional curves of complexity* n *and* m *using* $\mathcal{O}((nm)^{1-\delta})$ *time unless OVH fails.*

▶ **Theorem 10.** *For any* $\delta, \varepsilon > 0$*, there does not exist a* $(k - \varepsilon)$*-approximation algorithm for the continuous or discrete Fréchet distance between 2-dimensional curves of complexity* n *and* m *using* $\mathcal{O}((nm)^{1-\delta})$ *time unless OVH fails, where* $k = 3$ *(resp.* $k = 1 + \sqrt{2} \approx 2.41$*) if the curves lie in the* L_∞*-space (resp. in the Euclidean space).*

Our construction gives also a bound on Fréchet distance oracles (proof in the full version).

▶ **Theorem 11.** *Assume OVH holds true. For any* $\delta > 0$*, there exists a* $c > 0$ *such that for any* n *and* $m > c \log n$ *there is no data structure storing a curve of complexity* n *with* $\mathrm{poly}(n)$ *preprocessing time, that when given a query curve of at most complexity* m *can answer continuous or discrete Fréchet distance queries in* $\mathcal{O}((nm)^{1-\delta})$ *query time within the same approximation factor as in Theorem 9 and Theorem 10.*

4 2-Approximation for the discrete Fréchet distance in 1D

Given a value m and a 1-dimensional curve P of complexity $n \geq m$. We want to preprocess P in near-linear time and store information about P using $\mathcal{O}(m)$ space such that for any 1-dimensional query curve Q of complexity at most m we can approximate the discrete Fréchet distance between P and Q in $\mathcal{O}(m^2 \log m)$ time.

Preprocessing. Define $\delta_m = \min\{d_{\mathrm{dF}}(P,Q) \mid Q \text{ has complexity } m\}$. We want to compute a curve P' such that $d_{\mathrm{dF}}(P,P') \leq \delta_m/2$. The complexity of P' might be n, but it should be possible to store a compact description of P' in $\mathcal{O}(m)$ space. To compute P', we first compute the optimal m-simplification P^* of P for the discrete Fréchet distance. The curve P^* is

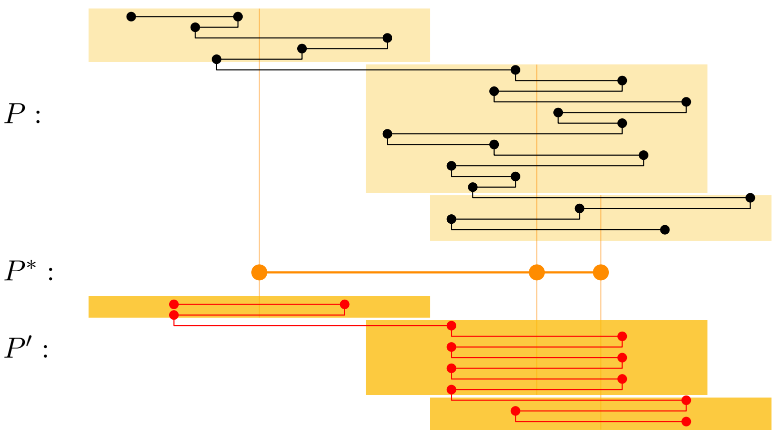

Figure 4 The black curve is P and the middle curve is an optimal 3-simplification for P. The red curve below is the compressed curve P'. Here, $k_1 = 3$, $k_2 = 7$, $k_3 = 3$ and $z_1 = -1$, $z_2 = -1$, $z_3 = 1$.

computed via searching over the possible values δ for δ_m and then computing the optimal δ-simplification. We compute the optimal δ-simplification of P using Algorithm 1. It begins by finding the longest prefix $P[1, i_1]$ that is contained in a 2δ-interval, whose center becomes the first vertex of the δ-simplification. Then, it continues by finding the largest index i_2 such that $P[i_1 + 1, i_2]$ is contained in a 2δ-interval and again the center of this interval is added to the δ-simplification. This process is repeated until the final vertex of P is reached.

Algorithm 1 Optimal δ-simplification of P.

$\ell \leftarrow 1$, $a \leftarrow P(1)$, $b \leftarrow P(1)$
for $i = 1$ *to* n **do**
  if $P(i) \leq a + 2\delta$ *and* $b - 2\delta \leq P(i)$ **then**
    $a \leftarrow \min\{a, P(i)\}$, $b \leftarrow \max\{b, P(i)\}$
  else
    $p_\ell \leftarrow \frac{a+b}{2}$, $a \leftarrow P(i)$, $b \leftarrow P(i)$, $i_\ell \leftarrow i - 1$, $\ell \leftarrow \ell + 1$
Return $P_\delta = \langle p_1, \ldots, p_\ell \rangle$, $i_0 = 0, i_1, \ldots, i_\ell = n$

The proof of the following lemma is in the full version.

▶ **Lemma 12.** *Algorithm 1 computes an optimal δ-simplification P_δ of P using $\mathcal{O}(n)$ time.*

If the complexity of the δ-simplification is greater than m, it holds that $\delta_m > \delta$. Otherwise, $\delta_m \leq \delta$. In contrast to higher dimensions, the optimal m-simplification for any m of a 1-dimensional curve P has distance $\delta_m \in \{|P(i) - P(j)|/2 \mid i, j \in \{1, \ldots, n\}\}$ to P, where the distance is only defined by two vertices of P. By Frederickson and Johnson [23], we can search over all possible values for δ_m using $\mathcal{O}(\log n)$ calls of Algorithm 1 and get the following lemma.

▶ **Lemma 13.** *We can compute an optimal m-simplification P^* of P together with a discrete matching that realizes the discrete Fréchet distance between P^* and P in $\mathcal{O}(n \log n)$ time.*

Let $P^* = \langle p_1, \ldots, p_\ell \rangle$ be an optimal m-simplification of P. Let M be the discrete matching between P and P^* computed by Algorithm 1, and $P(i_j)$ be the last vertex of P that is matched to p_j; i.e. $M = ((1,1), (2,1) \ldots, (i_1, 1), (i_1+1, 2), (i_1+2, 2), \ldots, (i_2, 2), (i_2+1, 3), \ldots, (i_\ell, \ell))$.

We construct a curve P' that has discrete Fréchet distance at most $\delta_m/2$ to P and can be stored in a compressed way using $\mathcal{O}(m)$ space (see Figure 4). We call P' a *compressed simplification*. For every $(x, y) \in M$, define $r_x = p_y - \delta_m/2$ if $P(x) < p_y$ and $r_x = p_y + \delta_m/2$ otherwise. Then, it holds that $|r_x - P(x)| \leq \delta_m/2$. The compressed simplification P' is $\langle r_1, \dots, r_n \rangle$ after deleting the vertices with identical predecessor, so that P' contains r_i only if $r_i \neq r_{i-1}$. For every j, we count the total number k_j that P' contains $p_j \pm \delta_m/2$ and define $z_j \in \{-1, 1\}$ such that the first time P' contains $p_j + z_j\delta_m/2$ is before P' contains $p_j - z_j\delta_m/2$. Therefore, the values $p_1, \dots, p_\ell$, $k_1, \dots, k_\ell$, $z_1, \dots, z_\ell$, and δ_m fully describe P'. The next lemma follows directly by construction and $\ell \leq m$.

▶ **Lemma 14.** *Given P and its optimal m-simplification P^*, we can compute in $\mathcal{O}(n)$ time a curve P' such that $d_{dF}(P, P') \leq \delta_m/2$ and P' can be stored using $\mathcal{O}(m)$ space.*

The Query Algorithm. In the following, we give a 2-approximation query algorithm for the discrete Fréchet distance between P and any query curve Q of complexity m. We use the precomputed compressed simplification P' and compute $d_{\mathrm{dF}}(P', Q)$ if it is at least $\frac{3}{2}\delta_m$.

▶ **Lemma 15.** *If $d_{dF}(P', Q) \leq \frac{3}{2}\delta_m$, then $d_{dF}(P, Q) \in [\delta_m, 2\delta_m]$. Otherwise, define δ such that $\frac{3}{2}\delta = d_{dF}(P', Q)$. Then, $d_{dF}(P, Q) \in [\delta, 2\delta)$.*

Proof. By definition of δ_m, it holds that $d_{\mathrm{dF}}(P, Q) \geq \delta_m$ for every curve Q of complexity m. Further, by Lemma 14 it holds that $d_{\mathrm{dF}}(P, P') \leq \delta_m/2$. Hence, if $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta_m$, then

$$\delta_m \leq d_{\mathrm{dF}}(P, Q) \leq d_{\mathrm{dF}}(P', Q) + d_{\mathrm{dF}}(P, P') \leq \frac{3}{2}\delta_m + \frac{1}{2}\delta_m \leq 2\delta_m.$$

Otherwise, it holds that $\frac{3}{2}\delta = d_{\mathrm{dF}}(P', Q) > \frac{3}{2}\delta_m$. Hence, $\delta > \delta_m$. Therefore,

$$\delta < \frac{3}{2}\delta - \frac{1}{2}\delta_m \leq d_{\mathrm{dF}}(P', Q) - d_{\mathrm{dF}}(P', P) \leq d_{\mathrm{dF}}(P, Q) \leq d_{\mathrm{dF}}(P', Q) + d_{\mathrm{dF}}(P', P) < 2\delta. \blacktriangleleft$$

Using Lemma 15, we could simply compute $d_{\mathrm{dF}}(P', Q)$ to get a 2-approximation algorithm for $d_{\mathrm{dF}}(P, Q)$. However, computing $d_{\mathrm{dF}}(P', Q)$ with known discrete Fréchet distance algorithms takes roughly $\mathcal{O}(m \cdot \sum_{i=1}^{\ell} k_i)$ time and $\sum_{i=1}^{\ell} k_i$ could be n. In the following, we make use of the special structure of P' to get a faster algorithm that decides whether $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta$ for any value $\delta \geq \delta_m$. We use the concept of the free space matrix $M_{(3/2)\delta}$, but do not compute all reachable entries explicitly. Define $s_i = \sum_{j=1}^{i-1} k_j + 1$ to be the index of the first vertex with value $p_i \pm \delta_m/2$ in P' and $s_{\ell+1} = |P'| + 1$. Algorithm 2 iterates over the sub-curves $P'[s_i, s_{i+1} - 1]$ of P' from $i = 1$ to ℓ and stores in the set F_{pre} the indices j of the vertices of Q such that $(s_i - 1, j)$ is reachable (see the hatched cells in Figure 5). Using F_{pre}, in iteration i we successively compute the set F_{new} of indices such that $(s_{i+1} - 1, j)$ is reachable and make use of the following property.

▶ **Observation 16.** *For any point $q \in \mathbb{R}$, it holds that exactly one of the following is true:*

- *$|q - p_i| \leq \delta$ and $|q - P'(s_i + x)| \leq \frac{3}{2}\delta$ for all $x \in \{0, \dots, k_i - 1\}$, or*
- *$|q - p_i| \in (\delta, 2\delta]$ and $|q - P'(s_i + x)| \leq \frac{3}{2}\delta$ for all even (resp. odd) $x \in \{0, \dots, k_i - 1\}$ and $|q - P'(s_i + x)| > \frac{3}{2}\delta$ for all odd (resp. even) $x \in \{0, \dots, k_i - 1\}$, or*
- *$|q - p_i| > 2\delta$.*

By Observation 16, it holds that whenever $|Q(j) - p_i| \leq \delta$, there exists an number a such that $(s_i + x, j)$ with $x \in \{0, \dots, k_i - 1\}$ is reachable if and only if $x \geq a$ and whenever $|Q(j) - p_i| > 2\delta$, then $(s_i + x, j)$ is not reachable for any integer $x \in \{0, \dots, k_i - 1\}$ and we set $a = \infty$. When $|Q(j) - p_i| \in (\delta, 2\delta]$, things get slightly more complicated. However, we can still store a number a such that $(s_i + x, j)$ is reachable for all $x \in \{a, a+2, \dots\}$ with $x \leq k_i - 1$. Since there might be more reachable pairs $(s_i + x, j)$, we store a set A of size at most m and a number u which can be used to reconstruct all other reachable pairs $(s_i + x, j)$ with $x \in \{0, \dots, a\}$. Then, let $B \subset \{0, \dots, \min\{k_i - 1, a\}\}$ be the set such that for all $x \leq \min\{k_i - 1, a\}$ it holds that $(s_i + x, j-1)$ is reachable if and only if $x \in B$. It holds that $B \neq \emptyset$ only if $|Q(j-1) - p_i| \in (\delta, 2\delta]$. In that case, define $z = 0$ (resp. $z = 1$) if $Q(j-1)$ and $Q(j)$ lie on the same side (resp. different sides) of p_i. Then for any $x + z > 0$, it holds that that $(s_i + x + z, j-1)$ is reachable if and only if $x \in B$. Using this property, we store a counter u that counts the sum of the values z until we reach again a vertex with $|Q(j') - p_i| \notin (\delta, 2\delta]$. If $x + z = 0$, we additionally check whether $(s_i - 1, j-1)$ or $(s_i - 1, j)$ is reachable by looking at the set F_{pre}. If one of them is reachable, then $(s_i, j-1) = (s_i + (u-u), j)$ is reachable and we add the current number u to the stored set A. In short, we intend to maintain the following invariant. During iteration i of the outer for-loop and at the end of iteration j of the inner for-loop, it holds that $(s_i + x, j)$ is reachable with $x \in \{0, \dots, k_i - 1\}$ if and only if

a) $x \in \{u - z \mid z \in A\} \cup \{a\}$ or **b)** $x > a$ and $|Q(j) - P'(s_i + x)| \leq (2/3)\delta$.

See Figure 5 for an example. Here, during iteration $i = 3$ of the outer for-loop and at the end of iteration $j = 7$ (resp. $j = 8$) of the inner for-loop, it holds that $A = [0, 2]$, $u = 2$, and $a = \infty$ (resp. $A = [\,]$, $u = 0$, and $a = 0$).

Algorithm 2 Decision Algorithm for $d_{\text{dF}}(P', Q) \leq \frac{3}{2}\delta$.

$F_{\text{new}} \leftarrow \{0\}$
for $i = 1$ *to* ℓ **do**
  $A \leftarrow [\,]$, $a \leftarrow \infty$, $u \leftarrow 0$, $F_{\text{pre}} \leftarrow F_{\text{new}}$, $F_{\text{new}} \leftarrow [\,]$
  for $j = 1$ *to* m **do**
    if $|Q(j) - p_i| > 2\delta$ **then**
      $A \leftarrow [\,]$, $a \leftarrow \infty$, $u \leftarrow 0$
    if $|Q(j-1) - Q(j)| > 2\delta$ *and* $|Q(j) - p_i|, |Q(j-1) - p_i| \in (\delta, 2\delta]$ **then**
      $a \leftarrow a + 1$, $u \leftarrow u + 1$
    if *($j \in F_{pre}$ or $j - 1 \in F_{pre}$) and* $|P'(s_i) - Q(j)| \leq \frac{3}{2}\delta$ **then**
      Append u to A
    if $|Q(j) - p_i| \leq \delta$ **then**
      $a \leftarrow \min\{a, u - \min\{\{x \mid x \in A\}\} \cup \{\infty\}\}$
      $A \leftarrow [\,]$, $u \leftarrow 0$
    Delete all $x \in A$ with $x \leq u - k_i$
    if $|P'(s_{i+1} - 1) - Q(j)| \leq \frac{3}{2}\delta$ *and ($a < k_i$ or $u - k_i + 1 \in A$)* **then**
      $F_{\text{new}} \leftarrow j$

▶ **Lemma 17.** *Assume that at the beginning of iteration i of the outer for-loop after setting $F_{pre} \leftarrow F_{new}$, it holds $(s_i - 1, j)$ is reachable if and only if $j \in F_{pre}$. Then, at the end of iteration i it holds that $(s_{i+1} - 1, j)$ is reachable if and only if $j \in F_{new}$.*

Proof. Denote with A_j, a_j, and u_j the value of A, a, and u at the end of iteration j in the inner for-loop of Algorithm 2. Then, we show that the following invariant is true; for every $x \in \{0, \dots, k_i - 1\}$, it holds that $(s_i + x, j)$ is reachable if and only if

a) $x \in \{u_j - z \mid z \in A_j\} \cup \{a_j\}$ or **b)** $x > a_j$ and $|Q(j) - P'(s_i + x)| \leq (2/3)\delta$.

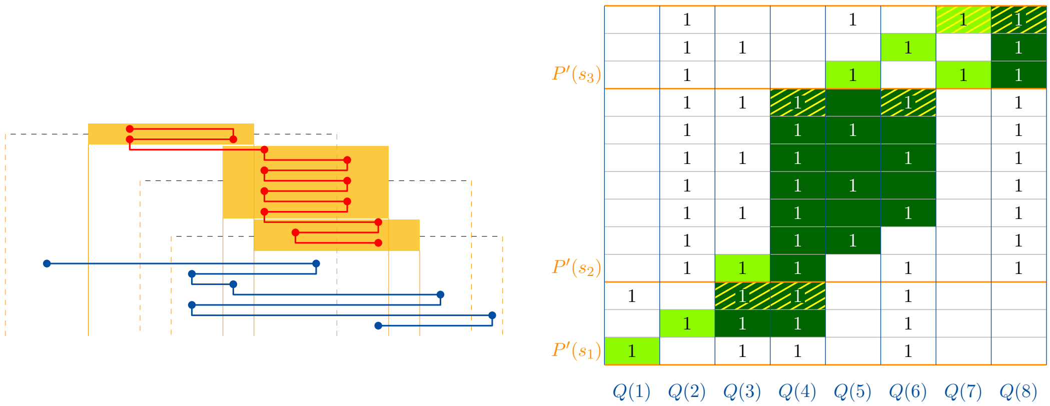

Figure 5 The compressed curve P' and the query curve Q are on the left. On the right is the free space matrix of P' and Q, where only 1-entries are added. The dark green cells visualize the parameter a and the light green cells the set $\{u - z \mid z \in A\}$. The hatched cells are the cells that have been in F_{pre} and F_{new}.

We prove this invariant via induction on j. For $j = 0$, the invariant clearly holds as $\{u_j - z \mid z \in A_j\} \cup \{a_j\} = \{\infty\}$. Now assume the invariant holds for $j - 1$. Let $x \in \{0, \dots, k_i - 1\}$. Then, by definition of the discrete Fréchet distance $(s_i + x, j)$ is reachable if and only if $|Q(j) - P(s_i + x)| \leq \frac{2}{3}\delta$ and $(s_i + x - 1, j)$, $(s_i + x, j - 1)$, or $(s_i + x - 1, j - 1)$ is reachable. We consider three different cases.

1. If $|Q(j) - p_i| > 2\delta$, then $(s_i + x, j)$ is not reachable, since $|Q(j) - P(s_i + x)| > \frac{2}{3}\delta$. By line 5 of Algorithm 2, the invariant holds.
2. If $|Q(j) - p_i| \leq \delta$, then $|Q(j) - P(s_i + x)| \leq \frac{2}{3}\delta$ for any $x \in \{0, \dots, k_i - 1\}$. Further, observe that if $(s_i + y, j)$ is reachable for $y \in \{0, \dots, k_i - 1\}$, then $(s_i + x, j)$ is also reachable for every $y \leq x \leq k_i - 1$. Define $y \in \{0, \dots, k_i - 1\}$ to be the minimum such that $(s_i + y, j)$ is reachable. Hence, by line 13 it holds that the invariant holds if $a_j = y$. Unless $y = 0$ and $(s_i - 1, j)$ or $(s_i - 1, j - 1)$ is reachable, it must hold that $(s_i + y, j - 1)$ is reachable as otherwise y would not be the minimum. Further again since y is the minimum and by the invariant for $j - 1$, it holds that $y = \min\{u_{j-1} - z \mid z \in A_{j-1}\} \cup \{a_{j-1}\}$. Now, consider the special case that $y = 0$ and $(s_i - 1, j)$, or $(s_i - 1, j - 1)$ is reachable. By assumption, it holds that $(s_i - 1, j)$ (resp. $(s_i - 1, j - 1)$) is reachable if and only if $j \in F_{\text{pre}}$ (resp. $j - 1 \in F_{\text{pre}}$). Hence, a_j is set to y by line 9–12 and the invariant is true.
3. It remains the case that $\delta < |Q(j) - p_i| \leq 2\delta$.
   a. First, consider the case that $x \geq a_{j-1}$. Then since $(s_i + a_{j-1}, j - 1)$ is reachable by the induction hypothesis, it holds that $|Q(j-1) - p_i| \leq 2\delta$. Hence, $|Q(j-1) - P(s_i + x)| \leq \frac{2}{3}\delta$ or $|Q(j - 1) - P(s_i + x - 1)| \leq \frac{2}{3}\delta$. Further again by the induction hypothesis it holds that $(s_i + x, j - 1)$ or $(s_i + x - 1, j - 1)$ is reachable. Hence, by line 7–8 the invariant holds for $x \geq a_{j-1}$.
   b. Otherwise, it holds that $x < a_{j-1}$. Consider the special case that $x = 0$ and $(s_i + x, j)$ is reachable because $|Q(j) - P(s_i + x)| \leq \frac{2}{3}\delta$ and $(s_i - 1, j)$ or $(s_i - 1, j - 1)$ are reachable. Then by 9 and the assumption on F_{pre}, it holds that $u_j \in A_j$. Otherwise it holds that $(s_i + x, j)$ is reachable if and only if $(s_i + x, j - 1)$ or $(s_i + x - 1, j - 1)$ is reachable and $|Q(j) - p_i| \leq \frac{3}{2}\delta$. This is true because it holds that either $|Q(j) - P(s_i + x)| \leq \frac{2}{3}\delta$ or $|Q(j) - P(s_i + x - 1)| \leq \frac{2}{3}\delta$ since $\delta < |Q(j) - p_i| \leq 2\delta$. Hence, $x - 1 \in A_{j-1}$ or $x \in A_{j-1}$. Therefore, $A_{j-1} \neq [\,]$ and by the algorithm it holds that $\delta < |Q(j - 1) - p_i| \leq 2\delta$. If

$\min\{Q(j-1), Q(j)\} > p_i$ or $\max\{Q(j-1), Q(j)\} < p_i$, then $(s_i + x, j)$ is reachable if and only if $(s_i + x, j-1)$ is reachable by Observation 16. Otherwise $(s_i + x, j)$ is reachable if and only if $(s_i + x - 1, j - 1)$ is reachable. This is ensured by line 7–8.

This proves that the invariant is true for every j. Note that $(s_{i+1} - 1, j) = (s_i + k_i - 1, j)$. By the invariant it holds that $(s_i + k_i - 1, j)$ is reachable if and only if $|P'(s_{i+1} - 1) - Q(j)| \leq \frac{3}{2}\delta$ and $a_j < k_i$ or $u_j - k_i + 1 \in A_j$. Hence the lemma follows by line 15–16. ◀

▶ **Lemma 18.** *For any value $\delta \geq \delta_m$, we can decide whether $d_{dF}(P', Q) \leq \frac{3}{2}\delta$ in $\mathcal{O}(m\ell)$ time.*

Proof. We run Algorithm 2 and output $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta$ if and only if $m \in F_{\mathrm{new}}$ at the end of Algorithm 2. The running time of Algorithm 2 is in $\mathcal{O}(m\ell)$, since the array A is sorted and in every iteration of the inner for loop at most one point gets appended to A. By induction on the index i of s_i and Lemma 17, it holds that $(|P'|, m) = (s_{\ell+1} - 1, m)$ is reachable if and only if $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta$. ◀

It remains to optimize the δ value. First, we decide whether $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta_m$ or not using Algorithm 2. If $d_{\mathrm{dF}}(P', Q) \leq \frac{3}{2}\delta_m$, then $d_{\mathrm{dF}}(P, Q) \in [\delta_m, 2\delta_m]$ by Lemma 15 and δ_m is a 2-approximation for $d_{\mathrm{dF}}(P, Q)$. Otherwise $d_{\mathrm{dF}}(P', Q) > \frac{3}{2}\delta_m$. In this case, we use the observation that the discrete Fréchet distance between P' and Q gets realizes by a point to point distance between a vertex of P' and a vertex of Q. Hence,

$$d_{\mathrm{dF}}(P', Q) \in \{|(p_i \pm \delta/2) - Q(j)| \mid i \in \{1, \dots, \ell\}, j \in \{1, \dots, m\}\}.$$

We sort these values in $\mathcal{O}(m^2 \log m)$ time and search over the ones that are greater $\frac{3}{2}\delta_m$. Using Algorithm 2, we compute in $\mathcal{O}(m^2 \log m)$ time $\frac{3}{2}\delta = d_{\mathrm{dF}}(P', Q)$. Then, it holds that $d_{\mathrm{dF}}(P, Q) \in [\delta, 2\delta]$ by Lemma 15. Therefore, we get the following theorem:

▶ **Theorem 19.** *For a given value m, there exists a data structure that preprocesses a 1-dimensional curve P of complexity n in $\mathcal{O}(n \log n)$ time using $\mathcal{O}(m)$ space such that for any 1-dimensional query curve Q of complexity at most m, we can approximate the discrete Fréchet distance between P and Q within a factor of 2 in $\mathcal{O}(m^2 \log m)$ time.*

The next corollary is a direct consequence of the theorem above.

▶ **Corollary 20.** *Given two 1-dimensional curves P and Q of complexity n and m with $n \geq m$. There exists a 2-approximation algorithm to compute the discrete Fréchet distance between P and Q in $\mathcal{O}(n \log n + m^2 \log m)$ time.*

By Theorem 9, it follows that there does not exist an algorithm using $\mathcal{O}(n \log n + m^2 \log m)$ time with a better approximation factor. Further, the lower bound in [12] shows the almost optimality of the running time.

5 $(3 + \varepsilon)$-approximation Algorithm for the Fréchet distance

In this section, we give a $(3+\varepsilon)$-approximation algorithm to compute the continuous/discrete Fréchet distance between two curves P and Q of complexity n resp. m with $m \leq n$. We describe this section for the continuous Fréchet distance, but everything works analogously for the discrete Fréchet distance as well. We begin with the decision variant for δ. In the first step, using the curve Q, we compute a curve P' of complexity at most $2m$ such that $d_{\mathrm{F}}(P, P') \leq \delta$ or decide that $d_{\mathrm{F}}(P, Q) > \delta$ in linear time.

▶ **Lemma 21.** *Algorithm 3 computes a curve P' of complexity at most $2m$ with $d_F(P, P') \leq \delta$ or returns $d_F(P, Q) > \delta$ in $\mathcal{O}(n)$ time.*

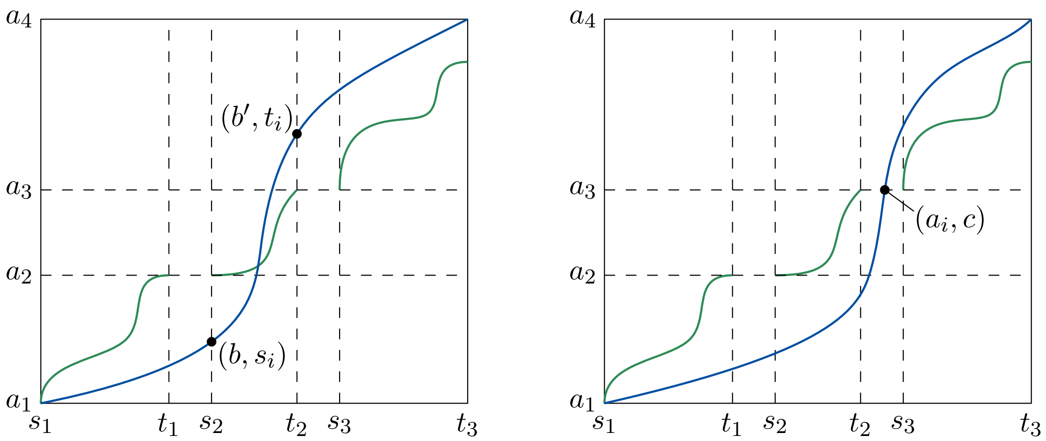

Figure 6 The green curve is a matching between the curves $P[a_i, a_{i+1}]$ and $Q[s_i, t_i]$ computed in Algorithm 3. The blue curves visualize the matching used in the proof of Lemma 21.

Proof. In the i-th iteration of the while-loop, it holds that $s_i \geq i$ by line 3 of Algorithm 3. Further, it holds that $a_i \leq a_{i+1}$ for every i and line 6 has running time $\mathcal{O}(a_{i+1} - a_i)$ as $\overline{Q(s_i)Q(t_i)} \subset \overline{Q(s_i)Q(\lfloor s_i + 1 \rfloor)}$. Hence, the running time follows. If a curve P' is returned, it holds that $d_F(P, P') \leq \delta$ by line 6 and Observation 2. The complexity of P' is at most $2m$ since there are at most m iterations of the while loop. It remains to show correctness for the case when $d_F(P, Q) > \delta$ is returned. Assume that $d_F(P, Q) > \delta$ is returned and in fact it holds $d_F(P, Q) \leq \delta$. Let M be the matching between P and Q that realizes the Fréchet distance. We consider two cases (see Figure 6). In the first case, there exist b, b' and an index i such that $(b, s_i), (b', t_i) \in M$ with $b \leq a_i$ and $b' > a_{i+1}$. This contradicts line 6 of the algorithm. Otherwise, there must exist an index i and a parameter c such that $(a_i, c) \in M$ with $t_{i-1} \leq c < s_i$. This contradicts line 3. Hence, it follows that $d_F(P, Q) > \delta$. ◀

Algorithm 3 Simplification of P depending on Q.

```
1 s_1 ← 1, a_1 ← 1, i ← 1
2 while True do
3     s_i ← min{s ≥ ⌊s_{i-1} + 1⌋ | ‖P(a_i) − Q(s)‖ ≤ δ} ∪ {∞}
4     if s_i = ∞ then
5         Return d_F(P, Q) > δ
6     Find largest a_{i+1} such that ∃ t_i ≤ ⌊s_i + 1⌋ with d_F(P[a_i, a_{i+1}], \overline{Q(s_i)Q(t_i)}) ≤ δ
7     if a_{i+1} = n then
8         Return P' = ⟨Q(s_1), Q(t_1), Q(s_2), Q(t_2), ..., Q(s_i), Q(t_i)⟩
9     i ← i + 1
```

Lemma 21 can be easily adapted such that it holds for the discrete Fréchet distance. Given P', we decide whether $d_F(P', Q) \leq 2\delta$ in $\mathcal{O}(m^2)$ time [3] or slightly faster [15]. If $d_F(P', Q) \leq 2\delta$, then $d_F(P, Q) \leq d_F(P', Q) + d_F(P, P') \leq 3\delta$. Otherwise, it holds that $d_F(P, Q) \geq d_F(P', Q) - d_F(P, P') > \delta$. Hence, we obtain the following lemma.

▶ **Lemma 22.** *Given two curves P and Q of complexity n and m in any metric space M. Assume that we can compute distances in M in constant time. We can 3-approximate the decision variant of the continuous/discrete Fréchet distance in $\mathcal{O}(n + m^2)$ time.*

For curves in the Euclidean space, Colombe and Fox (Theorem 11 of [18]) show how to transform a decision algorithm into an approximate optimization algorithm increasing the running time by a $\mathcal{O}(\log(n/\varepsilon))$ factor. Since all L_p-norms on $\mathbb{R}^d$ are equivalent and Lemma 22 holds for all L_p-norms, we can adopt their approach such that we can use it for all L_p-norms.

▶ **Theorem 23.** *There exist a $(3+\varepsilon)$-approximation algorithm for the continuous/discrete Fréchet distance using $\mathcal{O}((n+m^2)\log(n/\varepsilon))$ 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. Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42(3-4):203–219, 2005. `doi:10.1007/S00453-005-1165-Y`.
3. Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.*, 5:75–91, 1995. `doi:10.1142/S0218195995000064`.
4. Sergey Bereg, Minghui Jiang, Wencheng Wang, Boting Yang, and Binhai Zhu. Simplifying 3d polygonal chains under the discrete Fréchet distance. In Eduardo Sany Laber, Claudson F. Bornstein, Loana Tito Nogueira, and Luérbio Faria, editors, *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium, Búzios, Brazil, April 7-11, 2008, Proceedings*, volume 4957 of *Lecture Notes in Computer Science*, pages 630–641. Springer, 2008. `doi:10.1007/978-3-540-78773-0_54`.
5. Lotte Blank and Anne Driemel. A faster algorithm for the Fréchet distance in 1d for the imbalanced case. In Timothy M. Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom*, volume 308 of *LIPIcs*, pages 28:1–28:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. `doi:10.4230/LIPIcs.ESA.2024.28`.
6. 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`.
7. Karl Bringmann and Bhaskar Ray Chaudhury. Polyline simplification has cubic complexity. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.SOCG.2019.18`.
8. 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`.
9. Karl Bringmann and Marvin Künnemann. Multivariate fine-grained complexity of longest common subsequence. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1216–1235. SIAM, 2018. `doi:10.1137/1.9781611975031.79`.
10. Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *Journal of Computational Geometry*, 7(2):46–76, 2016. `doi:10.20382/JOCG.V7I2A4`.
11. Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discrete Comput. Geom.*, 58(1):180–216, July 2017. `doi:10.1007/s00454-017-9878-7`.

12 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 Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2887–2901. SIAM, 2019. doi:10.1137/1.9781611975482.179.

13 Siu-Wing Cheng and Haoqiang Huang. Curve simplification and clustering under Fréchet distance. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1414–1432. SIAM, 2023. doi:10.1137/1.9781611977554.CH51.

14 Siu-Wing Cheng and Haoqiang Huang. Solving Fréchet distance problems by algebraic geometric methods. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4502–4513. SIAM, 2024. doi:10.1137/1.9781611977912.158.

15 Siu-Wing Cheng and Haoqiang Huang. Fréchet distance in subquadratic time. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 5100–5113. SIAM, 2025. doi:10.1137/1.9781611978322.173.

16 Siu-Wing Cheng, Haoqiang Huang, and Le Jiang. Simplification of trajectory streams. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry, SoCG 2025, June 23-27, 2025, Kanazawa, Japan*, volume 332 of *LIPIcs*, pages 34:1–34:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.SOCG.2025.34.

17 Siu-Wing Cheng, Haoqiang Huang, and Shuo Zhang. Constant approximation of Fréchet distance in strongly subquadratic time. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 2329–2340. ACM, 2025. doi:10.1145/3717823.3718157.

18 Connor Colombe and Kyle Fox. Approximating the (continuous) Fréchet distance. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPIcs*, pages 26:1–26:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.SOCG.2021.26.

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

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

21 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance, 1994.

22 Arnold Filtser and Omrit Filtser. Static and streaming data structures for Fréchet distance queries. *ACM Trans. Algorithms*, 19(4):39:1–39:36, 2023. doi:10.1145/3610227.

23 Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM J. Comput.*, 13(1):14–30, 1984. doi:10.1137/0213002.

24 Joachim Gudmundsson, Martin P. Seybold, and Sampson Wong. Map matching queries on realistic input graphs under the Fréchet distance. *ACM Trans. Algorithms*, 20(2):14, 2024. doi:10.1145/3643683.

25 Aviad Rubinstein. Hardness of approximate nearest neighbor search. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1260–1268. ACM, 2018. doi:10.1145/3188745.3188916.

26 Thijs van der Horst and Tim Ophelders. Faster Fréchet distance approximation through truncated smoothing. *CoRR*, abs/2401.14815, 2024. doi:10.48550/arXiv.2401.14815.

27 Thijs van der Horst, Marc J. van Kreveld, Tim Ophelders, and Bettina Speckmann. A subquadratic n^{ϵ}-approximation for the continuous Fréchet distance. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1759–1776. SIAM, 2023. doi:10.1137/1.9781611977554.CH67.

28 Marc J. van Kreveld, Maarten Löffler, and Lionov Wiratma. On optimal polyline simplification using the Hausdorff and Fréchet distance. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, volume 99 of *LIPIcs*, pages 56:1–56:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.SOCG.2018.56.

29 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2-3):357–365, 2005. doi:10.1016/J.TCS.2005.09.023.