A Faster Algorithm for the Fréchet Distance in 1D for the Imbalanced Case

Lotte Blank
University of Bonn, Germany

Anne Driemel
University of Bonn, Germany

Abstract

The fine-grained complexity of computing the Fréchet distance has been a topic of much recent work, starting with the quadratic SETH-based conditional lower bound by Bringmann from 2014. Subsequent work established largely the same complexity lower bounds for the Fréchet distance in 1D. However, the imbalanced case, which was shown by Bringmann to be tight in dimensions $d \geq 2$, was still left open. Filling in this gap, we show that a faster algorithm for the Fréchet distance in the imbalanced case is possible: Given two 1-dimensional curves of complexity n and n^{α} for some $\alpha \in (0, 1)$, we can compute their Fréchet distance in $O(n^{2\alpha} \log^2 n + n \log n)$ time. This rules out a conditional lower bound of the form $O((nm)^{1-\epsilon})$ that Bringmann showed for $d \geq 2$ and any $\varepsilon > 0$ in turn showing a strict separation with the setting $d = 1$. At the heart of our approach lies a data structure that stores a 1-dimensional curve P of complexity n, and supports queries with a curve Q of complexity m for the continuous Fréchet distance between P and Q. The data structure has size in $\mathcal{O}(n \log n)$ and uses query time in $\mathcal{O}(m^2 \log^2 n)$. Our proof uses a key lemma that is based on the concept of visiting orders and may be of independent interest. We demonstrate this by substantially simplifying the correctness proof of a clustering algorithm by Driemel, Krivošija and Sohler from 2015.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, distance oracle, data structures, time series

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.28

Related Version *Full Version*: https://arxiv.org/abs/2404.18738

Funding This work was funded by 390685813 (Germany's Excellence Strategy – EXC-2047/1: Hausdorff Center for Mathematics); 416767905; and the Deutsche Forschungsgemeinschaft (DFG, German ResearchFoundation) – 459420781 (FOR AlgoForGe).

1 Introduction

Since its introduction to Computational Geometry by Alt and Godau [4], the complexity of computing or approximating the *Fréchet distance* has been the topic of much research and debate [2, 10]. Alt conjectured in 2009 [3] that the associated decision problem may be 3-SUM hard, which would likely rule out subquadratic algorithms [26]. With the advent of fine-grained complexity theory came the celebrated result by Bringmann [7] in 2014, and we now know that strictly subquadratic algorithms with running time in $O(n^{2-\epsilon})$ are indeed not possible, unless the Strong Exponential Time Hypothesis (SETH) fails. While Bringmann's construction uses curves in the plane, subsequent work by Bringmann and Mulzer [9] and later Buchin, Ophelders and Speckmann [12] extended the result showing the same hardness also for curves in one dimension. Bringmann in his initial paper showed the tightness of the algorithm by Alt and Godau up to logarithmic factors also in the imbalanced case. Namely, given two polygonal curves in the plane of complexity n and $m = n^{\alpha}$ for some $\alpha \in (0, 1]$, there is no algorithm that computes the Fréchet distance in time $O((nm)^{1-\epsilon})$ for any $\epsilon > 0$ unless SETH fails. However, the imbalanced case was left open for curves in 1D in [9, 12].

32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 28; pp. 28:1–28:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we fill this gap by showing a faster algorithm for computing the Fréchet distance in one dimension ruling out lower bounds of the form $O((nm)^{1-\epsilon})$ and thus showing a strict separation between the complexity in 1D and higher dimensions.

Signatures and visiting orders. In order to obtain our result, we give necessary and sufficient conditions for when the decision algorithm should accept a 1D instance. Our characterization uses the concept of *signatures* which were earlier introduced by Driemel, Krivošija and Sohler [21] in 2016. Intuitively, a δ-signature of P is a curve that consists of important minima and maxima of P preserving its overall shape. We combine this with the concept of *visiting orders* formalized by Bringmann, Driemel, Nusser and Psarros [8]. We show that the Fréchet distance between two 1-dimensional curves is at most δ if and only if there exist a so-called *coupled δ-visiting order*, subject to additional technical conditions. A coupled δ-visiting order is a sequence of ordered tuples of important indices of the vertices that define the polygonal curves P and Q including the indices of the *δ-signature vertices.* Our lemma leads to a shorter alternative proof of Theorem 3.7 of Driemel, Krivošija and Sohler [21], thereby substantially simplifying their proof of 17 pages in [20]. Our proof makes use of a greedy algorithm devised by Bringmann et. al. [8], which we slightly adapt to suit our needs.

Fréchet distance oracles. We develop our results in the form of a Fréchet distance oracle and obtain a more refined result. A Fréchet distance oracle is a data structure that preprocesses an input curve to answer queries with a query curve for the Fréchet distance between input and query curve. We give a brief history of results focussing on the continuous Fréchet distance. Fréchet distance oracles were introduced by Driemel and Har-Peled [18] in 2012 in the context of their work on the shortcut Fréchet distance in fixed dimension d. They constructed a $(1+\varepsilon)$-approximate Fréchet distance oracle for the case where the query curves are line segments. Their data structure uses space in $\mathcal{O}(n(1/\varepsilon)^{2d} \cdot \log^2(1/\varepsilon))$ and query time in $\mathcal{O}((1/\varepsilon)^2 \log n \log\log n)$. It can also be used to answer queries to a sub-curve of P. They extended this data structure to support queries of complexity m. In the extended version it has size $\mathcal{O}(n \log n)$, uses query time in $\mathcal{O}(m^2 \log n \log(m \log n))$ and approximates the continuous Fréchet distance up to a (large) constant factor. For exact Fréchet distance queries, the picture looks much different. In a very recent result, Cheng and Huang [16] presented a data structure that builds upon point location in arrangements of polynomials of bounded degree. Their data structure has size in $\mathcal{O}(nm)^{\mathrm{poly}(d,m)}$, expected query time in $\mathcal{O}((md)^{\mathcal{O}(1)} \log(nm))$ solving Fréchet distance queries in d dimensions, where the query complexity $m \geq 2$ is given at preprocessing time. Given the apparent difficulty of the problem, some works have focused on the restricted setting where the queries are line segments in the plane (i.e., $d = 2$ and $m = 2$) [17, 13]. In this vein, Buchin, van der Hoog, Ophelders, Schlipf, Silveira, and Staals [13] describe a data structure that uses storage in $\mathcal{O}(nk^{3+\varepsilon} + n^2)$ and query time in $\mathcal{O}((n/k) \log^2 n + \log^4 n)$, where $k \in [1, \ldots, n]$ is a tradeoff parameter and $\varepsilon > 0$ is an arbitrarily small constant. Another line of work studies approximate distance oracles in the discrete setting where the discrete Fréchet distance is derived either from the Euclidean metric or from a general graph metric [5, 22, 23, 24, 28].

Distance approximation. There has been an ongoing search for faster approximation algorithms [9, 14, 29, 30], however the gap between upper and lower bounds remains large. While the lower bounds do not rule out a subquadratic-time 3-approximation, the best-known approximation factors that can be obtained in this regime remain polynomial in n, even in 1D. Van der Horst and Ophelders show, given two 1-dimensional curves and a parameter $\alpha \in [1, n]$, one can α-approximate the continuous Fréchet distance in time in $O(n \log^3 n + (n^2/\alpha^3) \log^2 n \log \log n)$ [29] and this is currently the best possible in 1D. For specialized classes of well-behaved curves better approximations are known [6, 11, 19, 27].

Our Results. Our results are for the exact computation of the continuous Fréchet distance of 1-dimensional curves. Our data structure supports query curves of arbitrary complexity m, which need not be known during preprocessing. We can preprocess a curve of complexity n using storage and preprocessing time in $\mathcal{O}(n \log n)$ to support queries with time in $\mathcal{O}(m^2 \log^2 n)$. Our data structure leads to several improvements for the exact computation of the Fréchet distance in 1D: First, it leads to an improved running time for the decision problem of the Fréchet distance in the case that the complexity of the δ-signatures is low. Specifically, given two curves of complexity n and m, with $n \geq m$, we obtain an exact algorithm with running time in $\mathcal{O}(s_P s_Q \log n + n \log n)$, where s_P and s_Q denote the complexities of the δ-signatures of the two curves. Second, if $m = n^\alpha$ for some $\alpha \in (0, 1)$, we can compute the continuous Fréchet distance in $\mathcal{O}(n^{2\alpha} \log^2 n + n \log n)$ time. This improves upon the running time of $\mathcal{O}(n^{1+\alpha} \log n)$ of the classical algorithm [4] and rules out lower bounds of type shown in [7] for the imbalanced case.

2 Preliminaries

For any two points $p_1, p_2 \in \mathbb{R}$, $\overline{p_1 p_2}$ denotes the directed line segment connecting p_1 with p_2. A time series 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 $P(i) \in \mathbb{R}$. We obtain a polygonal curve that can be viewed as a function $P : [1, n] \to \mathbb{R}$, where $P(i + \alpha) = (1 - \alpha)P(i) + \alpha P(i + 1)$ for $i \in \{1, \ldots, n\}$ and $\alpha \in [0, 1]$. This curve is also denoted with $\langle P(1), \ldots, P(n)\rangle$. We call the points $P(i)$ vertices and the line segments $\overline{P(i)P(i+1)}$ edges of P. Further, $P[s_1, s_2]$ is the subcurve of P starting in $P(s_1)$ and ending in $P(s_2)$ for $s_1 \leq s_2$. We define $\min(P[s_1, s_2]) = \min\{P(s) |\ s \in [s_1, s_2]\}$ and $\arg\min(P[s_1, s_2])$ to be an $s_1 \leq s \leq s_2$ such that $P(s) = \min(P[s_1, s_2])$. The δ-range of a time series P is the interval $B(P, \delta) = \{x |\ \exists s \in [1, n] \text{ s.t. } |x - P(s)| \leq \delta\}$.

Let $P : [1, n] \to \mathbb{R}$ and $Q : [1, m] \to \mathbb{R}$ be two time series. Then, the (continuous) *Fréchet distance* between them is defined as

$$d_{\mathrm{F}}(P, Q) = \min_{h_P \in \mathcal{F}_P, h_Q \in \mathcal{F}_Q} \max_{a \in [0,1]} |P(h_P(a)) - Q(h_Q(a))|,$$

where $\mathcal{F}_P$ is the set of all continuous, non-decreasing functions $h_P : [0, 1] \to [1, n]$ with $h_P(0) = 1$ and $h_P(1) = n$, respectively $\mathcal{F}_Q$ for Q. We say that a point $Q(t)$ is *matchable* to a point $P(s)$ for a value $\delta \geq 0$ if there exist functions $h_P \in \mathcal{F}_P$ and $h_Q \in \mathcal{F}_Q$ and a $b \in [0, 1]$ such that $\max_{a \in [0,1]} |P(h_P(a)) - Q(h_Q(a))| \leq \delta$ and $h_P(b) = s$ and $h_Q(b) = t$.

▶ **Observation 1.** *Let $s \leq n$ and $t \leq m$. If $d_F(P[1, s], Q[1, t]), d_F(P[s, n], Q[t, m]) \leq \delta$, then $d_F(P[1, n], Q[1, m]) \leq \delta$.*

The next lemma is a generalization of Lemma 37 of Bringmann, Driemel, Nusser and Psarros [8] and will be used to prove Lemma 9. We need the definition of a δ-monotone time series. Let $P : [1, n] \to \mathbb{R}$ be a time series, then it is δ-monotone increasing (resp. decreasing) if for all $s < s' \in [1, n]$, it holds that $P(s) \leq P(s') + \delta$ (resp. $P(s) \geq P(s') - \delta$).

▶ **Lemma 2** (Generalization of Lemma 37 in [8]). *For two time series $P = \langle P(1), \ldots, P(n)\rangle$ and $Q = \langle Q(1), \ldots Q(m)\rangle$, it holds that $d_F(P, Q) \leq \delta$ if*

(i) *P, Q are 2δ-monotone increasing (resp. decreasing),*
(ii) *$|P(1) - Q(1)| \leq \delta$, $|P(n) - Q(m)| \leq \delta$,*

(iii) $P \subseteq B(Q, \delta)$, $Q \subseteq B(P, \delta)$,
(iv) $P(1)$ *or* $Q(1)$ *is a global minimum (resp. maximum) of its time series, and*
(v) $P(n)$ *or* $Q(m)$ *is a global maximum (resp. minimum) of its time series.*

The lemma can be proven using the same arguments as the proof of Lemma 37 in [8]. For the sake of completeness, we provide a full proof in the appendix.

2.1 Signatures and visiting orders

Signatures and (coupled) visiting orders are key concepts used in this paper. Figure 1 gives an example of a δ-signature.

▶ **Definition 3** (δ-signature)**.** *Let* $P = \langle P(1), \dots, P(n) \rangle$ *be a time series and* $\delta \geq 0$*. Then, a* δ-signature $P' = \langle P(i_1), \dots, P(i_t) \rangle$ *with* $1 = i_1 < \dots < i_t = n$ *of* P *is a time series with the following properties:*

(a) (non-degenerate) *For* $k = 2, \dots, t-1$*, it holds* $P(i_k) \notin \overline{P(i_{k-1}), P(i_{k+1})}$.
(b) (2δ-monotone) *For* $k = 1, \dots, t-1$, $P[i_k, i_{k+1}]$ *is* 2δ*-monotone increasing or decreasing.*
(c) (minimum edge length) *For* $k = 2, \dots, t-2$, $|P(i_k) - P(i_{k+1})| > 2\delta$, $|P(i_1) - P(i_2)| > \delta$, *and* $|P(i_{t-1}) - P(i_t)| > \delta$.
(d) (range) *For* $k = 2, \dots, t-2$*, it holds that* $P(s) \in \overline{P(i_k), P(i_{k+1})}$ *for all* $s \in [i_k, i_{k+1}]$ *and for* $k = 1$ *or* $k = t-1$*, it holds* $P(s) \in B(\langle P(i_k), P(i_{k+1}) \rangle, \delta)$ *for all* $s \in [i_k, i_{k+1}]$.

We define an *extended* δ*-signature* P'' in the following way: If a δ-signature of P does not exist or $t = 2$, P'' consists of $P(1)$ and $P(n)$ and the global minimum and global maximum of P in order of their indices. Otherwise, we use P' as defined above and we add two vertices with indices $1 \leq i' < i_2$ and $i_{t-1} < i'' \leq n$ such that $P[1, i_2] \subset \overline{P(i'), P(i_2)}$ and $P[i_{t-1}, n] \subset \overline{P(i_{t-1}), P(i'')}$. The vertices of P'' are called δ*-signature vertices* of P. Further, we say a point $P(s)$ of P is *supported by the* δ*-signature edge* $\overline{P(i_k)P(i_{k+1})}$ if $i_k \leq s \leq i_{k+1}$. It holds that the subcurve of P from the first until the second (resp. from the second last until the last) δ-signature vertex of P is contained in a 2δ-range.

The signatures have a unique hierarchical structure as the following lemma shows.

▶ **Lemma 4** (Driemel, Krivošija and Sohler [21])**.** *Given a time series* $P : [1, n] \to \mathbb{R}$ *with vertices in general position, there exists a series of unique signatures* $P_1, \dots, P_k$ *and corresponding parameters* $0 = \delta_1 < \delta_2 < \dots < \delta_{k+1}$*, such that*

(a) P_i *is a* δ*-signature of* P *for any* $\delta \in [\delta_i, \delta_{i+1})$,
(b) *the vertex set of* P_{i+1} *is a subset of the vertex set of* P_i,
(c) P_k *is the linear interpolation of* $P(1)$ *and* $P(n)$.

With this hierarchical structure, it is possible to construct the following data structure.

▶ **Theorem 5** (Driemel, Krivošija and Sohler [21])**.** *Given a time series* $P : [1, n] \to \mathbb{R}$ *with vertices in general position, we can construct a data structure in time in* $\mathcal{O}(n \log n)$ *and space in* $\mathcal{O}(n)$*, such that given a parameter* l *we can extract in time in* $\mathcal{O}(l \log l)$ *a signature of maximal size* l' *with* $l' \leq l$.

We adopt the concept of δ-visiting orders as defined by Bringmann et. al. [8]. A δ-visiting order of Q on P for vertices $Q(j_1), \dots, Q(j_t)$ consists of indices $v_1 \leq \dots \leq v_t$ of vertices of P such that $|P(v_k) - Q(j_k)| \leq \delta$ for all k. We extend this notion as follows. Our idea is to fix vertices of both time series and to interleave two δ-visiting orders – one of Q on P and one of P on Q. Figure 1 shows an example of a coupled δ-visiting order, which we define further below.

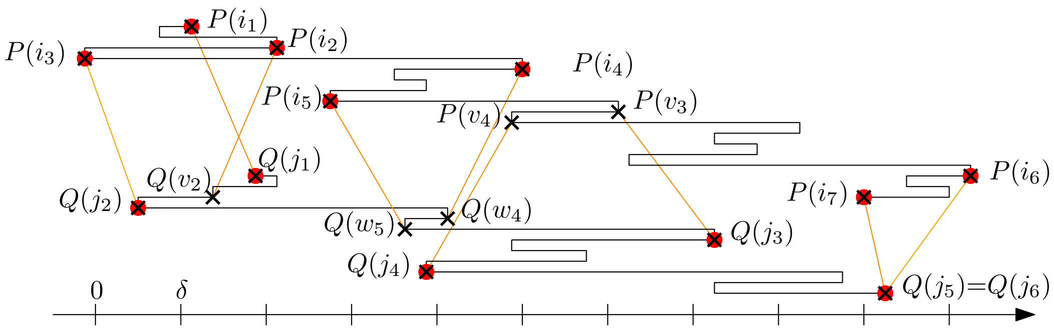

Figure 1 In this paper, the vertices of the time series are drawn as vertical segments for clarity. The δ-signature vertices are marked with a red disk and $((i_1, j_1), (i_2, v_2), (i_3, j_2), (i_4, w_4), (i_5, w_5), (v_3, j_3), (v_4, j_4), (i_6, j_5), (i_7, j_6))$ is a coupled δ-visiting order.

▶ **Definition 6** (coupled δ-visiting order). *Consider two time series $P = \langle P(1), \ldots, P(n)\rangle$ and $Q = \langle Q(1), \ldots, Q(m)\rangle$. Let $v_1 \le \cdots \le v_{s_Q}$ be a δ-visiting order of Q on P for the δ-signature vertices $Q(j_1), \ldots, Q(j_{s_Q})$ and $w_1 \le \cdots \le w_{s_P}$ be a δ-visiting order of P on Q for the δ-signature vertices $P(i_1), \ldots, P(i_{s_P})$. These two δ-visiting orders are said to be crossing-free if there exists no k, l such that $i_k < v_l$ and $j_l < w_k$, or $v_k < i_l$ and $w_l < j_k$. In this case, the ordered sequence containing all tuples (v_k, j_k) and (i_l, w_l), where $k = 1, \ldots, s_Q$ and $l = 1, \ldots, s_P$, is called* coupled δ-visiting order.

For the sake of a clean presentation, we allow to add a vertex that lies on one of the edges between the first and the second δ-signature vertex of P as well as one vertex between the second to last and the last δ-signature vertex of P, resp. for Q (see $Q(v_2)$ in Figure 1). Adding those at most four vertices does not change the Fréchet distance, it only helps us defining the δ-visiting orders via vertex indices. Our algorithm described in Section 4.1.1 chooses the parameter for these vertices (named $v_{1,2}$, $w_{2,1}$, $\tilde{v}$, and $\tilde{w}$) when handling the traversals at the beginning and the ending parts of the two curves. In this context, we call these vertices *pseudo-start*, resp., *pseudo-end vertex*. It is important to note that they are chosen with respect to the other curve.

2.2 Orthogonal Range Successor Problem

The *Orthogonal range successor problem* is a variant of orthogonal range reporting. Here, a set S of points must be stored such that the point in S with the smallest x-coordinate contained in a query range Q can be reported efficiently. The query range is an axis-aligned hyper-rectangle. For points in the plane, the *layered range tree* is a modified version of range trees which uses fractional cascading [15], and can be used to solve the orthogonal range successor problem. For a comprehensive overview and further details, see Agarwal [1].

▶ **Theorem 7** ([1, 15]). *In the pointer machine model, there exist a data structure solving the orthogonal range successor problem in the plane of size and preprocessing time in $\mathcal{O}(n \log n)$ that uses query time in $\mathcal{O}(\log n)$.*

3 Key Lemma

Driemel, Krivošija and Sohler [21] showed that if $d_F(P, Q) \le \delta$, then there exists a δ-visiting order for the δ-signature vertices of P on Q. We strengthen their statement and show an equivalence that characterizes the Fréchet distance in one dimension using the coupled δ-visiting order. In the proof, we use the following lemma:

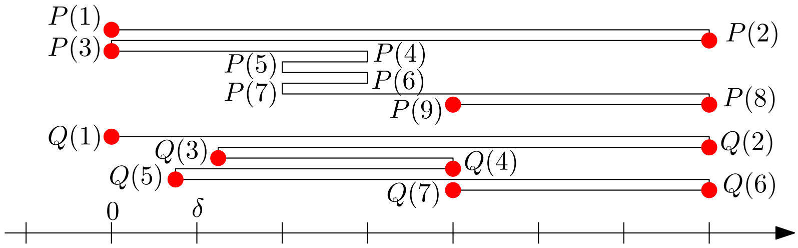

Figure 2 The red disks are the δ-signature vertices. The indices $1 \leq 2 \leq 5 \leq 6 \leq 7$ of Q are a δ-visiting order of the δ-signature vertices of P on Q, resp. $1 \leq 2 \leq 5 \leq 6 \leq 7 \leq 8 \leq 9$ of the δ-signature vertices of Q on P. Those are the only existing δ-visiting orders for the δ-signatures. Therefore, there does not exist a coupled δ-visiting order of P and Q, since $(3, 5)$ and $(5, 3)$ cross.

▶ **Lemma 8.** *If there exists a coupled δ-visiting order $((v_1, w_1), \ldots, (v_t, w_t))$ of P and Q with $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$ and $d_F(P[v_{t-1}, n], Q[w_{t-1}, m]) \leq \delta$, then there exists a coupled δ-visiting order $((v'_1, w'_1), \ldots, (v'_{t'}, w'_{t'}))$ with*

(a) $d_F(P[1, v'_2], Q[1, w'_2]) \leq \delta$ *and* $d_F(P[v'_{t'-1}, n], Q[w'_{t'-1}, m]) \leq \delta$,

(b) $v'_2 < v'_3 < \cdots < v'_{t'-1}$ *and* $w'_2 < w'_3 < \cdots < w'_{t'-1}$,

(c) $|P(v'_k) - Q(w'_{k+1})| > \delta$, $|P(v'_{k+1}) - Q(w'_k)| > \delta$ *for* $k = 2, \ldots, t' - 2$, *and*

(d) $P[v'_k, v'_{k+1}]$ *and* $Q[w'_k, w'_{k+1}]$ *are both 2δ-monotone increasing or both 2δ-monotone decreasing for* $k = 1, \ldots, t' - 1$.

Proof. If $w_2 = w_3$, there exist values $w' \leq w_2$ and $v' \leq v_2$ such that $P[1, v_3] \subset B(Q(w'), \delta)$, $Q[w', w_3] \subset B(P(v_3), \delta)$ and $d_F(P[1, v'], Q[1, w']) \leq \delta$. Hence, $d_F(P[1, v_3], Q[1, w_3]) \leq \delta$. The same holds in the case $v_2 = v_3$, resp. for $v_{t-2} = v_{t-1}$ and $w_{t-2} = w_{t-3}$. If $v_k = v_{k+1}$ for a $k = 2, \ldots, t - 2$, then $Q(w_k)$ or $Q(w_{k+1})$ is no δ-signature vertex. Hence, we can omit (v_k, w_k) or (v_{k+1}, w_{k+1}) and keep a coupled δ-visiting order with Property (b). If $|P(v_k) - Q(w_{k+1})| \leq \delta$ and $P(v_k)$ is a δ-signature vertex, then $P(v_{k+1})$ and $Q(w_k)$ are no δ-signature vertices. In this case, we set $w_k = w_{k+1}$ and remove the tuple (v_{k+1}, w_{k+1}). Similarly, if $|P(v_{k+1}) - Q(w_k)| \leq \delta$.

Consider the case that $Q(w_k)$ is a δ-signature vertex, $Q[w_k, w_{k+1}]$ is 2δ-monotone increasing and $P[v_k, v_{k+1}]$ is not for a $k = 1, \ldots, t$. Then, $P(v_k)$ cannot be δ-signature vertex. Set $v'_k = \arg\min(P[v_k, v_{k+1}])$. Then, $P[v'_k, v_{k+1}] \subseteq [P(v'_k), P(v'_k) + 2\delta]$ since $P[v_k, v_{k+1}]$ is 2δ-monotone decreasing. Hence, $P[v'_k, v_{k+1}]$ is also 2δ-monotone increasing. If $Q(w_{k+1})$ is a δ-signature vertex, then $|P(v'_k) - Q(w_k)| \leq \delta$ since $Q(w_k) \leq Q(w_{k+1}) - 2\delta$ and $|Q(w_{k+1}) - P(v_{k+1})| \leq \delta$. Otherwise, $P(v_{k+1})$ is a δ-signature vertex and $v_{k+1} = v'_k$. Since $|Q(w_{k+1}) - P(v_{k+1})| \leq \delta$, $|Q(w_k) - P(v_k)| \leq \delta$ and $Q(w_k)$, $P(v_{k+1})$ are local minima, it follows that $|P(v'_k) - Q(w_k)| \leq \delta$. Hence, we can reassign $v_k = v'_k$. If $k = 2$, we reassign v_3 instead of v_2 in a similar way. This does not interfere with the other reassignments by the monotonicity conditions of δ-signatures. After the reassignment $((v_1, w_1), \ldots, (v_t, w_t))$ is a coupled δ-visiting order with Properties (a)-(d). ◀

▶ **Lemma 9** (Key Lemma). *Let P, Q be two time series which are endowed with pseudo-start and pseudo-end vertices. Then, $d_F(P, Q) \leq \delta$ if and only if there exist a coupled δ-visiting order $((v_1, w_1), \ldots, (v_t, w_t))$ of P and Q such that $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$ and $d_F(P[v_{t-1}, n], Q[w_{t-1}, m]) \leq \delta$.*

Figure 2 shows that it is not sufficient to use the notion of a δ-visiting order alone. In this example, it holds that $d_F(P, Q) > \delta$.

Proof. We start with showing one direction of the equivalence statement. So, assume $d_F(P,Q) \leq \delta$. Then, there exist values $v'_1 \leq v'_2 \leq \cdots \leq v'_t \in \mathbb{R}$ and $w'_1 \leq w'_2 \leq \cdots \leq w'_t \in \mathbb{R}$ such that

- $d_F(P[v'_1, v'_2], Q[w'_1, w'_2]) \leq \delta$ and $d_F(P[v'_{t-1}, v'_t], Q[w'_{t-1}, w'_t]) \leq \delta$,
- $|P(v'_k) - Q(w'_k)| \leq \delta$,
- $P(v'_k)$ or $Q(w'_k)$ is a δ-signature vertex for all $k = 1, \ldots, t$, and
- all δ-signature vertices of P and Q are contained in that sequence.

Set $v_k = v'_k$ and $w_k = w'_k$ for $k = 1, 2, t-1, t$. For $k = 3, \ldots, t-2$, consider the case that $Q(w'_k)$ is a δ-signature vertex. Refer to the left of Figure 3 for a visualization. Then, set $w_k = w'_k$. Let $Q(w'_a)$ be the δ-signature vertex before $Q(w'_k)$ and $Q(w'_b)$ the one afterwards. If w'_k is a local maximum (resp. minimum), then by the range property of a δ-signature for all $v'_a \leq v \leq v'_b$ holds that $P(v) \leq Q(w'_k) + \delta$ (resp. $P(v) \geq Q(w'_k) - \delta$). Therefore, one of the endpoints of the edge containing $P(v'_k)$ must lie in $[Q(w'_k) - \delta, Q(w'_k) + \delta]$ and we set v_k to that point. A symmetric construction works for the case that $P(v'_k)$ is a δ-signature vertex.

We need to show that $v_1 \leq v_2 \leq \cdots \leq v_t$ and $w_1 \leq w_2 \leq \cdots \leq w_t$. Since we round each point v'_k to an endpoint of the edge containing v'_k, the only time where it might get violated is if v'_k and v'_{k+1} lie on the same edge and $Q(w'_k)$ and $Q(w'_{k+1})$ are δ-signature vertices (or with the role of P and Q changed). However, if $Q(w'_k) < Q(w'_{k+1})$ (resp. $Q(w'_k) > Q(w'_{k+1})$), then $P(v'_k) < P(v'_{k+1})$ (resp. $P(v'_k) > P(v'_{k+1})$) by the edge-length property of a signature. Hence, $v_k = \lfloor v'_k \rfloor$ and $v_{k+1} = \lceil v'_{k+1} \rceil$. So, $(v_1, w_1), \ldots, (v_t, w_t)$ is a coupled δ-visiting order with the desired properties.

To show the other direction, let $((v_1, w_1), \ldots, (v_t, w_t))$ be a coupled δ-visiting order of P and Q with $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$, $d_F(P[v_{t-1}, n], Q[w_{t-1}, m]) \leq \delta$ and the properties of Lemma 8. We show that Lemma 2 can be applied to $P[v_k, v_{k+1}]$ and $Q[w_k, w_{k+1}]$ for all $k = 2, \ldots, t-2$. Then, it follows that $d_F(P[v_k, v_{k+1}], Q[w_k, w_{k+1}]) \leq \delta$ and by Observation 1, it holds that $d_F(P, Q) \leq \delta$.

Condition (i), (ii) and (iv) of Lemma 2 are true by construction. So, it remains to prove that Conditions (iii) and (v) are true. We show those properties for the case that $Q(w_k)$ is a δ-signature vertex which is a local minimum of Q. The other cases follow analogously.

1. If $P(v_k), P(v_{k+1})$ are supported by a 2δ-monotone increasing δ-signature edge, then $P(v_{k+1})$ or $Q(w_{k+1})$ is a δ-signature vertex that is a local maximum, i.e., Condition (v) holds. Let $P(v^*)$ denote the δ-signature vertex before $P(v_{k+1})$ on P.
   a. If $|P(v^*) - Q(w_k)| \leq \delta$, then $P[v_k, v_{k+1}] \geq Q(w_k) - \delta$.
   b. Otherwise, $P(v^*) + \delta < Q(w_k) \leq Q[w_{k-1}, w_k]$. Refer to the right of Figure 3. Therefore, $Q(w_{k-1})$ is a δ-signature and $P(v_{k-1})$ not. By the minimum edge length property and the 2δ-monotonicity, it follows that $P(s) \geq P(v_{k-1}) - 2\delta \geq (Q(w_{k-1}) - \delta) - 2\delta > Q(w_k) - \delta$ for all $v_{k-1} \leq s \leq v_{k+1}$.
2. Otherwise, $P(v_k), P(v_{k+1})$ are supported by a 2δ-monotone decreasing δ-signature edge. Then, $Q(w_{k+1})$ must be a δ-signature vertex of Q that is a global maximum on $Q[w_k, w_{k+1}]$ by Property (c) of Lemma 8. Hence, Condition (v) holds. Further, for all $v_k \leq v' \leq v_{k+1}$, it holds that $P(v') \geq P(v_{k+1}) - 2\delta \geq Q(w_{k+1}) - 3\delta \geq Q(w_k) - \delta$.

If $Q(w_{k+1})$ is a local maximum δ-signature vertex, then $P[v_k, v_{k+1}] \leq Q(w_{k+1}) + \delta$, resp. for P and Q switched. Therefore, Condition (iii) is true. We have shown that all conditions of Lemma 2 are true for $P[v_k, v_{k+1}]$ and $Q[w_k, w_{k+1}]$, which completes the proof. ◀

Lemma 8 and Lemma 9 yield the following:

▶ **Observation 10.** *If $d_F(P,Q) \leq \delta$, then the number of δ-signature vertices of P is at most two more than the complexity of Q.*

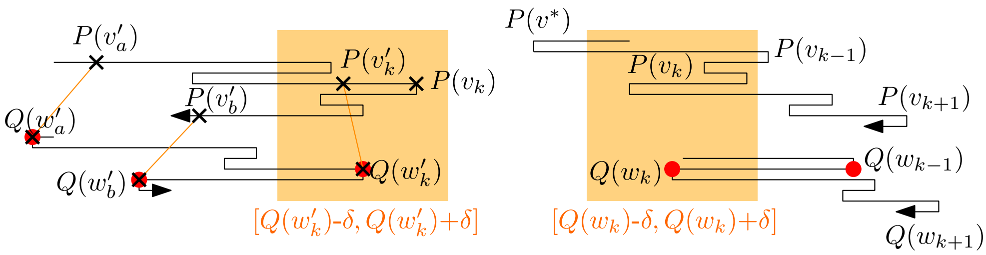

Figure 3 Visualization of parts of the proof of Lemma 9.

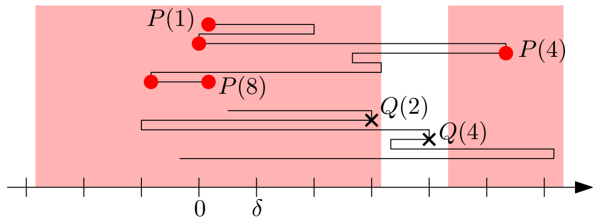

Figure 4 $P(1), P(3), P(4), P(7)$ and $P(8)$ are the δ-signature vertices of P. If $\widehat{Q}$ obtained by removing $Q(2)$ from Q, then $d_F(P, \widehat{Q}) > \delta$. So, we have to define $r_1 = [P(1) - 3\delta, P(1) + 3\delta]$. If $\widehat{Q}$ is Q after omitting $Q(4)$, then it still holds that $d_F(P, \widehat{Q}) \leq \delta$.

Lemma 9 implies the following corollary, which is Theorem 3.7 by Driemel, Krivošija and Sohler [21]. We can even strengthen their statement to say $r_a = [P'(a) - 3\delta, P'(a) + 3\delta]$ instead of $[P'(a) - 4\delta, P'(a) + 4\delta]$ for $a = 1$ or $a = s_P$. Figure 4 shows why we define $r_1 = [P'(1) - 3\delta, P'(1) + 3\delta]$.

▶ **Corollary 11** (Theorem 3.7 in [21]). *Let $P' = \langle P(i_1), \ldots, P(i_{s_P})\rangle$ be a δ-signature of P. Let $r_1 = B(P(1), 3\delta)$, $r_{s_P} = B(P(i_{s_P}), 3\delta)$ and $r_k = B(P(i_k), \delta)$ for $k = 2, \ldots, s_P - 1$. Let Q and $\widehat{Q}$ be time series such that $d_F(P, Q) \leq \delta$ and $\widehat{Q}$ is Q after removing some vertex $Q(j')$ with $Q(j') \notin \bigcup_{1 \leq k \leq l} r_k$. Then, it holds that $d_F(P, \widehat{Q}) \leq \delta$.*

Proof Sketch. We double the vertex $Q(j' - 1)$ in $\widehat{Q}$ to get the same complexity of Q and $\widehat{Q}$. By Lemma 9, there exists a coupled δ-visiting order $V = ((v_1, w_1), \ldots, (v_t, w_t))$ of P and Q with $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$ and $d_F(P[v_{t-1}, n], Q[w_{t-1}, m]) \leq \delta$. Therefore, $j' > w_2$ since $Q[1, w_2] \subset B(P[1, v_2], \delta) \subset B(P(1), 3\delta)$. It similarly follows that $j' < w_{t-1}$. The δ-signature $\widehat{Q}'$ of $\widehat{Q}$ can differ from the δ-signature $Q'$of Q only if $j' = w_l$ is a δ-signature vertex. In this case, let $Q(w_a)$ be the δ-signature vertex before $Q(w_l)$ and $Q(w_b)$ the one after $Q(w_l)$. If $Q[w_a, j' - 1]$ (resp. $Q[j' + 1, w_b]$) is contained in a 2δ-range, then $\widehat{Q}'$ is Q' after omitting $Q(w_a)$ (resp. $Q(w_b)$) and $Q(j')$. Otherwise, $\widehat{Q}'$ is Q' after adding a vertex $Q(j'')$ instead of $Q(j')$, where $\widehat{Q}[w_a, w_b] \subset \overline{Q(w_a)Q(j'')} \cup \overline{Q(j'')Q(w_b)}$. Then, it holds that $|P(v_l) - Q(j'')| \leq \delta$. After deleting the corresponding tuples from V in the first case and adding (v_l, j'') instead of (v_l, j') in the second, we get a coupled δ-visiting order V' of P and $\widehat{Q}$. Since $w_2 < j' < w_{t-1}$, it fulfills the properties of Lemma 9 for P and $\widehat{Q}$ and $d_F(P, \widehat{Q}) \leq \delta$. ◀

4 Fréchet Distance Oracle

In this section, we present a data structure to store a time series P and to compute the Fréchet distance between P and a query time series Q. Note that the data structure can be built without knowing the complexity of the query.

The Data Structure. Let P be a time series of complexity n. Then, we store the data structure of Theorem 5 and a layered range tree storing the points $(1, P(1)), (2, P(2)), \dots, (n, P(n))$.

4.1 Decision Algorithm

We begin with the decision variant of the problem. Our query algorithm is based on Lemma 9. Usually, the so-called free space diagram is used to solve the decision problem of the Fréchet distance. In contrast to this approach, we focus solely on *super cells*, which arise from the grid superimposed by the δ-signature edges onto the parametric space of the two time series. For example, the orange cell $[i_k, i_{k+1}] \times [j_l, j_{l+1}]$ in Figure 5 is a super cell. The computations within the super cell corresponding to the first δ-signature edge of each time series, as well as for the last one (yellow cells in Figure 5), are handled differently by Step (B) and (D). For all other super cells, Step (C) computes the right exit point and top exit point. These exit points are defined as follows:

▶ **Definition 12.** *Let P and Q be time series and $\langle P(i_1), \dots, P(i_{s_P})\rangle$ and $\langle Q(j_1), \dots, Q(i_{s_Q})\rangle$ be their extended δ-signatures. Then for $k = 1, \dots, s_{P-1}$, $l = 1, \dots, s_{Q-1}$, we define the* right (resp. top) exit point *$v_{k,l} \le i_{k+1}$ (resp. $w_{k,l} \le j_{l+1}$) to be the smallest index such that there exist tuples $(v_1, w_1), \dots, (v_r, w_r)$ with $v_r, w_r \in \mathbb{N}$ for $r = 3, \dots, t-2$, where*
(a) *$(v_r, w_r) = (v_{k,l}, j_l)$ (resp. $(v_r, w_r) = (i_k, w_{k,l})$),*
(b) *$v_1 \le \dots \le v_r$ and $w_1 \le \dots \le w_r$,*
(c) *$i_1, \dots, i_k \in \{v_1, \dots, v_r\}$ and $j_1, \dots, j_l \in \{w_1, \dots, w_r\}$,*
(d) *$|P(v_s) - Q(w_s)| \le \delta$ for $s = 1, \dots, r$, and*
(e) *$d_F(P[1, v_2], Q[1, w_2]) \le \delta$.*
If no such index exists, we set $v_{k,l} = \infty$ (resp. $w_{k,l} = \infty$).

The Query Algorithm. Let Q denote the query time series of complexity m and δ the distance parameter. Then, we check whether the δ-signature of P has at most m vertices. If it has more, we stop and return $d_F(P, Q) > \delta$. Otherwise, the algorithm proceeds with the following four steps:

Step (A) Compute the δ-signature $\langle P(i_1), \dots, P(i_{s_P})\rangle$ of P and $\langle Q(j_1), \dots, Q(j_{s_Q})\rangle$ of Q and construct a layered range tree storing the points $(1, Q(1)), \dots, (m, Q(m))$.

Step (B) Compute the first value $v_{1,2} \le i_2$ such that $d_F(P[1, v_{1,2}], Q[1, j_2]) \le \delta$ and the first value $w_{2,1} \le j_2$ such that $d_F(P[1, i_2], Q[1, w_{2,1}]) \le \delta$. If no such value exists, we set it to be infinity.

Step (C) Compute the exit points $v_{k,l}$ and $w_{k,l}$ by iterating over the super cells in a row-by-row order and within each row from left to right.

Step (D) Compute the latest value $\widetilde{v} \ge i_{s_P-1}$ such that $d_F(P[\widetilde{v}, n], Q[j_{s_Q-1}, m]) \le \delta$ and the latest value $\widetilde{w} \ge j_{s_q-1}$ such that $d_F(P[i_{s_P-1}, n], Q[\widetilde{w}, m]) \le \delta$. If no such value exists, we set it to be zero.

If $v_{s_P-1, s_Q-1} \le \widetilde{v}$ or $w_{s_P-1, s_Q-1} \le \widetilde{w}$, return $d_F(P, Q) \le \delta$. Otherwise, return $d_F(P, Q) > \delta$.

In Section 4.1.1 and 4.1.2, we describe how Step (B)-(D) can be computed.

▶ **Theorem 13.** *Given a time series P of complexity n, we can construct a data structure that uses size and preprocessing time in $\mathcal{O}(n \log n)$ and supports the following type of queries. For a distance parameter $\delta > 0$ and a time series Q of complexity $m < n$, it can decide whether $d_F(P, Q) \le \delta$ in time in $\mathcal{O}(m^2 \log n)$. If s_P denotes the complexity of the δ-signature of P, respectively s_Q for Q, then the running time is also in $\mathcal{O}(s_P s_Q \log n + m \log n)$.*

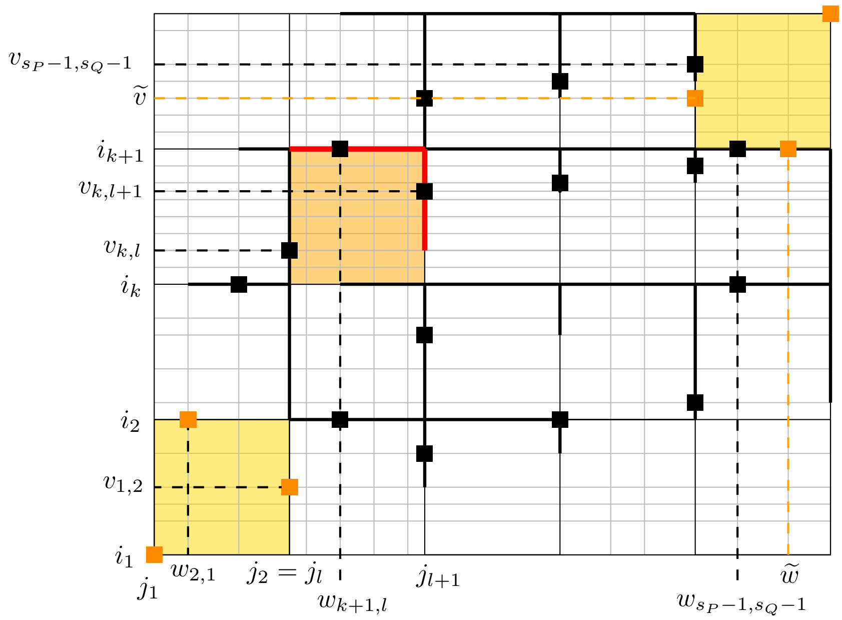

Figure 5 For the orange cell $[i_k, i_{k+1}] \times [j_l, j_{l+1}]$, we find the smallest index $v_{k,l-1}$ such that $(v_{k,l-1}, P(v_{k,l-1})) \in [v_{k,l}, i_{k+1}] \times B(Q(j_{l+1}), \delta)$ and the smallest index $w_{k+1,l}$ such that $(w_{k+1,l}, Q(w_{k+1,l})) \in [j_l, j_{l+1}] \times B(P(i_{k+1}, \delta)$. The thick lines mark the query intervals for the indices of the exit points. To compute $i_2, j_2, \widetilde{v}$ and $\widetilde{w}$, Lemma 14 is used.

Proof. It takes time in $\mathcal{O}(n \log n)$ to compute the layered range tree of same size by Theorem 7 and time in $\mathcal{O}(n \log n)$ to construct the data structure of Theorem 5 of size in $\mathcal{O}(n)$. With slight modifications, it can answer also queries as in the beginning and Step (A) of the query algorithm. Using the definition of Lemma 4, this data structure stores the vertices of P exactly once together with some token separators such that the vertices after the i-th separator belong to P_i. By Lemma 4, we can store a pointer from the i-th separator to the interval $[\delta_i, \delta_{i+1})$ such that the δ-signature, for any $\delta \in [\delta_i, \delta_{i+1})$, contains all vertices after this separator. To check whether the δ-signature of P has at most m vertices takes $\mathcal{O}(m)$ time. If it has more the total running time is in $\mathcal{O}(m)$. Otherwise, it holds that $s_P \leq m$. Step (A) takes $\mathcal{O}(m \log m)$ time by Theorem 5 and 7. To compute $v_{1,2}$, $w_{2,1}$, $\widetilde{v}$ and $\widetilde{w}$ takes $\mathcal{O}(m \log n)$ time by Lemma 14. Each recursive iteration of Step (C) takes $\mathcal{O}(\log n)$ time by Lemma 15. There are $\mathcal{O}(s_P s_Q)$ recursion steps. Since $s_Q, s_P \leq m$, the total running time is in $\mathcal{O}(m^2 \log n)$.

It remains to prove the correctness. If the δ-signature of P has more than m vertices, $d_F(P, Q) > \delta$ by Observation 10. Assume that $v_{s_P-1,s_Q-1} \leq \widetilde{v}$. We add vertices at $P(v_{1,2})$, $Q(w_{2,1})$, $P(\widetilde{v})$ and $Q(\widetilde{w})$ if there were none before. Then, by the definition of left exit points, there exist tuples such that $(v_1, w_1), \ldots, (v_{t-2}, w_{t-2}), (\widetilde{v}, j_{s_Q-1}), (n, m)$ is a coupled δ-visiting order with $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$. Further, by Step (D), it holds $d_F(P[\widetilde{v}, n], Q[j_{s_Q-1}, m]) \leq \delta$. Hence, $d_F(P, Q) \leq \delta$ by Lemma 9. In the case that $d_F(P, Q) \leq \delta$, there exist a coupled δ-visiting order $(v_1, w_1), \ldots, (v_t, w_t)$ of P and Q such that $d_F(P[1, v_2], Q[1, w_2]) \leq \delta$ and $d_F(P[v_{t-1}, n], Q[w_{t-1}, m]) \leq \delta$ by Lemma 9. Hence, $v_{s_P-1,s_Q-1} \leq v_{t-1} \leq \widetilde{v}$ or $w_{s_P-1,s_Q-1} \leq w_{t-1} \leq \widetilde{w}$. So, the algorithm returns $d_F(P, Q) \leq \delta$ and is correct. ◀

4.1.1 Step (B) and Step (D)

Step (D) can be done in the same way as Step (B), after reversing the order of the vertices in both time series. Therefore, we only describe Step (B) here. Algorithm 1 computes the value $w_{2,1}$. With exchanged roles of P and Q, it computes $v_{1,2}$. An example of this algorithm is pictured in Figure 6.

▶ **Lemma 14.** *Given the layered range trees for P and Q, it is possible to compute $w_{2,1}$ in $\mathcal{O}(\min\{m,n\}(\log n + \log m))$ time.*

Algorithm 1 Computation of $w_{2,1}$.

1 Find minimum $w_{2,1} \leq j_2$ such that $P[1,i_2] \subset B(Q(w_{2,1}),\delta)$;
2 **if** *there does not exist one* **then**
3   **return** $w_{2,1} = \infty$;
4 Set $s = i_2$ and $t = w_{2,1}$;
5 **while** $s \neq 1$ *or* $t \neq 1$ **do**
6   Find minimum $s' \leq s$ such that $Q[1,t] \subset B(P(s'),\delta)$;
7   **if** $s' = s$ **then**
8   **return** $w_{2,1} = \infty$;
9   Set $s = s'$;
10   Find minimum $t' \leq t$ such that $P[1,s] \subset B(Q(t'),\delta)$;
11   **if** $t' = t$ **then**
12   **return** $w_{2,1} = \infty$;
13   Set $t = t'$;
14 **return** $w_{2,1} = w_{2,1}$;

Proof. We show that if Algorithm 1 returns $w_{2,1} < \infty$, then it is the minimum $j \leq j_2$ such that $d_F(P[1,i_2], Q[1,j]) \leq \delta$ and otherwise there does not exist one. If $w_{2,1} < \infty$, we found a traversal: Every time we reassign s (resp. t), the part $P[s',s]$ (resp. $Q[t',t]$) is matched to $Q(t)$ (resp. $P(s)$). By construction, it holds that their distance is at most δ. Hence, $d_F(P[1,i_2], Q[1,w_{2,1}]) \leq \delta$. See Figure 6 for an example.

Let $s^* \leq i_2$ be such that $P[1,i_2] \subset \overline{P(s^*)P(i_2)}$. Further, assume that there exist $t_1 \leq j_2$ and $t_2 \leq j_2$ such that $|P(i_2) - Q(t_1)| \leq \delta$ and $|P(s^*) - Q(t_2)| \leq \delta$. Then, there must exist a point t' on Q before j_2 such that $P[1,i_2] \subset B(Q(t'),\delta)$, because $P[1,i_2]$ is contained in a 2δ-range by the definition of δ-signature. So, if the algorithm returns $w_{2,1} = \infty$ in Line 3, then there exist an $s \leq i_2$ such that $|P(s) - Q(t')| > \delta$ for all $t' \leq j_2$. Hence, for all $j \leq j_2$ it holds $d_F(P[1,i_2], Q[1,j]) > \delta$. This also shows that if $w_{2,1} < \infty$, it is the minimum t such that $d_F(P[1,i_2], Q[1,t]) \leq \delta$.

In the case that the algorithm returns $w_{2,1} = \infty$ in Line 8 or 12, it holds that $P(s') \neq P(s)$ for all $s' < s$, where s is the value at the time it stops. Therefore, $P(s)$ is the maximum or minimum of $P[1,s]$, respectively $Q(t)$ of $Q[1,t]$. Let $s^* \leq s$ be such that $P[1,s] \subset \overline{P(s^*)P(s)}$. For all $t' < t$, it holds that $|P(s) - Q(t)| \leq \delta$. Therefore, $|P(s^*) - Q(t')| > \delta$. In the same way, it follows that there exists a point $t^* < t$ such that $|Q(t^*) - P(s')| > \delta$ for all $s' < s$. Hence, $d_F(P,Q) > \delta$, because $P(s^*)$ is not matchable to anything before $Q(t)$ and $Q(t^*)$ not to anything before $P(s)$.

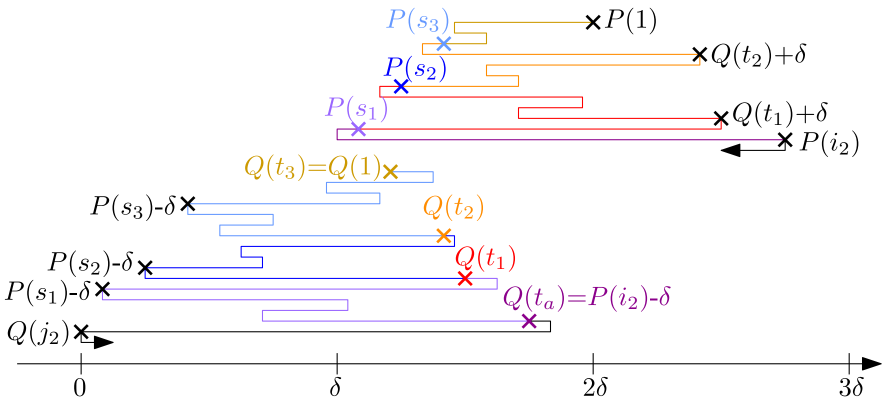

Figure 6 Example of a matching computed by Algorithm 1. The parts matched together are of the same color.

It remains to prove the running time. Assume there exist two values $s_1 < s_2$ on the same edge of P that were both s' at some point during the algorithm and are no vertices. Without loss of generality, let $P(s_1) < P(s_2)$. Then, $P(s_1) = \min(P[1, s_1])$ and $P(s_2) = \min(P[1, s_2])$. Let t_1 be the value t' attained after $s' = s_1$. Then, $Q(t_1) = P(s_1) + \delta$ as otherwise the algorithm would have stopped the iteration after. Further, since $P(s_2) = \min(P[1, s_2])$, it must hold that $P(s_2) = Q(t_1) + \delta = P(s_1) + 2\delta$. This is a contradiction to $P[1, i_2]$ is contained in a 2δ-range and $P(s_1)$, $P(s_2)$ are no vertices. The same arguments work for $Q(t_1)$ and $Q(t_2)$. Hence, there are at most $\mathcal{O}(\min\{m, n\})$ steps until $s = 1$ and $t = 1$ or the algorithm stops. Each iteration takes $\mathcal{O}(\log n + \log m)$ time by Theorem 7. So, the total running time is in $\mathcal{O}(\min\{m, n\}(\log n + \log m))$. ◀

4.1.2 Step (C): Computation of the Exit Points

In Step (C), we compute iterative the right and top exit points. For $k = 1, \dots, s_Q - 1$, we set $v_{k,1} = \infty$ and $w_{1,k} = \infty$ since there does not exist exit points for those values by Property (c) and (e) of the definition.

Given the values $v_{k,l}$ and $w_{k,l}$, we compute the values $v_{k,l+1}$ and $w_{k+1,l}$ in the following way: If $v_{k,l} \neq \infty$, set $w_{\min} = j_l$. Otherwise, set $w_{\min} = w_{k,l}$. Then, using the layered range tree, we can compute the minimum index $w_{k+1,l} \leq j_{l+1}$ such that $(w_{k+1,l}, Q(w_{k+1,l})) \in [w_{\min}, j_{l+1}] \times B(P(i_{k+1}), \delta)$. If there does not exist such a point, we set $w_{k+1,l} = \infty$. Symmetrically, if $w_{k,l} \neq \infty$, set $v_{\min} = i_k$. Otherwise, set $v_{\min} = v_{k,l}$. Then, we compute the minimum value $v_{k,l+1}$ such that $(v_{k,l+1}, Q(v_{k,l+1})) \in [v_{\min}, i_{k+1}] \times B(Q(j_{l+1}), \delta)$. If there does not exist such a point, we set $v_{k,l+1} = \infty$. The orange super cell in Figure 5 visualizes this construction. Here, $w_{k,l} = \infty$ and the red bars show the query intervals for the indices.

▶ **Lemma 15.** *Given the indices $v_{k,l}$ and $w_{k,l}$, the indices $v_{k,l+1}$ and $w_{k+1,l}$ can be computed in $\mathcal{O}(\log n)$ time.*

Proof. We use the algorithm above to compute $v_{k,l+1}$ and $w_{k+1,l}$. Its running time is in $\mathcal{O}(\log n)$ by Theorem 7. Let v^* be the value computed by the algorithm for $v_{k,l+1}$. If $v^* \neq \infty$, then $v_{k,l} \neq \infty$ or $w_{k,l} \neq \infty$ and one of them is an exit point. Therefore, there exist tuples $(v_1, w_1), \dots, (v_r, w_r), (v^*, j_{l+1})$ fulfilling the properties (a)-(e) of the definition of exit points. It remains to prove that it is the minimum. Let $v_{k,l+1}$ be the left exit point. Then, by definition of the exit points it holds that $v_{k,l} \neq \infty$ or $w_{k,l} \neq \infty$. Further, it must hold that $|P(v^*) - Q(j_{l+1})| \leq \delta$ and $v_{k,l+1} \leq i_{k+1}$ and $w_{\min} \leq v^*$. Therefore, $v^* = v_{k,l+1}$. Symmetrically, it follows that the construction of $w_{k+1,l}$ is correct. ◀

4.2 Critical values for computation

To compute the exact Fréchet distance, we search over a set of critical values (candidate values for the Fréchet distance) for δ using the decision algorithm in each step. For curves in 1D, this set consists of (i) the distances between a vertex of P and one of Q, i.e., $|P(i) - Q(j)|$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$, and (ii) for each pair of vertices of the same time series, half the value of their distance, i.e., $|P(i) - P(j)|/2$ for $i, j = 1, \ldots, n$ and $|Q(i) - Q(j)|/2$ for $i, j = 1, \ldots, m$. In the following, we assume without loss of generality that $m \leq n$.

Our search over these values is carried out in several (but a constant number) of stages. In the end, we combine the results of the different stages by taking the minimum over all valid answers.

We first discuss how to search over the values of type (i). We can extract an (increasing-order) sorted list A of the n vertices of P from the data structure in linear time. In addition, we compute a sorted list B of the m vertices of Q. Consider the implicit $n \times m$ matrices M_1 and M_2, where the entry $M_1(i,j)$ is the value $\max(0, a_i - b_j)$, and the entry $M_2(i,j)$ is the value $\max(0, b_j - a_i)$. Observe that both matrices consist of entries that are sorted in each row and column. Now, Fredrickson and Johnson [25] showed that we can find the k-th smallest item in M_1 (resp. M_2) in $\mathcal{O}(m \log(2n/m))$ time. Hence, we can perform an implicit binary search over all values in M_1 and M_2 with total running time in $\mathcal{O}(\log n(m \log(2n/m) + m^2 \log n))$.

For the critical values of type (ii) we have to be careful not to search over the entire set of at most $\binom{n}{2}$ values from curve P (resp., Q). Even with the procedure used above, this would incur a running time of at least $O(n \log n)$ in the worst case which we want to avoid. We observe that it is sufficient to search over the $O(n)$ values of δ where the δ-signature changes. These values can be extracted in sorted order from the data structures of P and Q. Once we have these values stored in sorted order, we can perform a binary search over them using $O(\log n)$ calls to the decision algorithm.

Overall, using Theorem 13 we obtain the following result.

▶ **Theorem 16.** *Given a time series P of complexity n, we can construct a data structure that returns $d_F(P,Q)$ for a time series Q of complexity $m \leq n$. It has size and preprocessing time in $\mathcal{O}(n \log n)$ and its query time lies in $\mathcal{O}(m^2 \log^2 n)$.*

This theorem leads directly to a new algorithm to compute the Fréchet distance in 1D.

▶ **Corollary 17.** *Let P be a time series of complexity n and Q of complexity n^α with $\alpha \in [0,1]$. Then, it is possible to compute the Fréchet distance between them in $\mathcal{O}(n^{2\alpha} \log^2 n + n \log n)$ time.*

References

1 Pankaj K. Agarwal. Range searching. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*, pages 809–837. Chapman and Hall/CRC, 2004. doi:10.1201/9781420035315.CH36.

2 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.

3 Helmut Alt. The computational geometry of comparing shapes. *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*, pages 235–248, 2009.

4 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.

5 Boris Aronov, Tsuri Farhana, Matthew J. Katz, and Indu Ramesh. Discrete Fréchet distance oracles, 2024. doi:10.48550/arXiv.2404.04065.

6 Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In *Algorithms–ESA 2006: 14th Annual European Symposium, Zurich, Switzerland, September 11-13, 2006. Proceedings 14*, pages 52–63. Springer, 2006.

7 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.

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 Wolfgang Mulzer. Approximability of the discrete Fréchet distance. *Journal of Computational Geometry*, 7(2):46–76, 2016.

10 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.

11 Kevin Buchin, Jinhee Chun, A Markovic, W Meulemans, M Löffler, Yoshio Okamoto, and Taichi Shiitada. Folding free-space diagrams: computing the Fréchet distance between 1-dimensional curves. In *33rd International Symposium on Computational Geometry (SoCG 2017)*, pages 641–645. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

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 Maike Buchin, Ivor van der Hoog, Tim Ophelders, Lena Schlipf, Rodrigo I. Silveira, and Frank Staals. Efficient Fréchet distance queries for segments. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 29:1–29:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.29.

14 Timothy M. Chan and Zahed Rahmati. An improved approximation algorithm for the discrete Fréchet distance. *Inf. Process. Lett.*, 138:72–74, 2018. doi:10.1016/J.IPL.2018.06.011.

15 Bernard Chazelle and Leonidas J. Guibas. Fractional cascading: I. A data structuring technique. *Algorithmica*, 1(2):133–162, 1986. doi:10.1007/BF01840440.

16 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.

17 Mark de Berg, Ali D. Mehrabi, and Tim Ophelders. Data structures for Fréchet queries in trajectory data. In Joachim Gudmundsson and Michiel H. M. Smid, editors, *Proceedings of the 29th Canadian Conference on Computational Geometry, CCCG 2017, July 26-28, 2017, Carleton University, Ottawa, Ontario, Canada*, pages 214–219, 2017.

18 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.

19 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.

20 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. *CoRR*, abs/1512.04349, 2015. arXiv:1512.04349.

21 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 766–785. SIAM, 2016. doi:10.1137/1.9781611974331.CH55.

22 Anne Driemel, Ioannis Psarros, and Melanie Schmidt. Sublinear data structures for short Fréchet queries. *arXiv preprint arXiv:1907.04420*, 2019.

23 Anne Driemel, Ivor van der Hoog, and Eva Rotenberg. On the discrete Fréchet distance in a graph. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 36:1–36:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.SOCG.2022.36.

24 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.

25 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.

26 Anka Gajentaan and Mark H Overmars. On a class of o (n2) problems in computational geometry. *Computational geometry*, 5(3):165–185, 1995.

27 Joachim Gudmundsson, Majid Mirzanezhad, Ali Mohades, and Carola Wenk. Fast Fréchet distance between curves with long edges. *Int. J. Comput. Geom. Appl.*, 29(2):161–187, 2019. doi:10.1142/S0218195919500043.

28 Ivor van der Hoog, Eva Rotenberg, and Sampson Wong. Approximate discrete Fréchet distance: simplified, extended and structured. *CoRR*, abs/2212.07124, 2022. doi:10.48550/arXiv.2212.07124.

29 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.

30 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.