A Subquadratic $n^\varepsilon$-approximation for the Continuous Fréchet Distance

Thijs van der Horst∗  Marc van Kreveld†  Tim Ophelders‡  Bettina Speckmann§

Abstract

The Fréchet distance is a commonly used similarity measure between curves. It is known how to compute the continuous Fréchet distance between two polylines with $m$ and $n$ vertices in $\mathbb{R}^d$ in $O(mn(\log \log n)^2)$ time; doing so in strongly subquadratic time is a longstanding open problem. Recent conditional lower bounds suggest that it is unlikely that a strongly subquadratic algorithm exists. Moreover, it is unlikely that we can approximate the Fréchet distance to within a factor 3 in strongly subquadratic time, even if $d = 1$. The best current results establish a tradeoff between approximation quality and running time. Specifically, Colombe and Fox (SoCG, 2021) give an $O(\alpha)$-approximate algorithm that runs in $O((n^3/\alpha^2) \log n)$ time for any $\alpha \in [\sqrt{n}, n]$, assuming $m = n$. In this paper, we improve this result with an $O(\alpha)$-approximate algorithm that runs in $O((n + mn/\alpha) \log^3 n)$ time for any $\alpha \in [1, n]$, assuming $m \leq n$ and constant dimension $d$.

∗Department of Information and Computing Sciences, Utrecht University, Netherlands and Department of Mathematics and Computer Science, TU Eindhoven, Netherlands; t.w.j.vanderhorst@uu.nl
†Department of Information and Computing Sciences, Utrecht University, Netherlands; m.j.vankreveld@uu.nl
‡Department of Information and Computing Sciences, Utrecht University, Netherlands and Department of Mathematics and Computer Science, TU Eindhoven, Netherlands; t.a.e.ophelders@uu.nl
§Department of Mathematics and Computer Science, TU Eindhoven, Netherlands; b.speckmann@tue.nl
1 Introduction

Measuring similarity is a fundamental data analysis task which is used, for example, to cluster data, search in databases, or construct phylogenetic trees. The similarity of two geometric shapes is generally defined via distance measures. Common distance measures are the Hausdorff distance, the area of symmetric difference, and the Fréchet distance. While the Hausdorff distance is defined for any two compact subsets of a space, the area of symmetric difference applies only to simple polygons in the plane. The Fréchet distance is most easily defined for curves, although generalizations to surfaces exist. Since the Fréchet distance respects the order along the two input curves, it is generally the measure of choice to determine the similarity of two curves.

Godau [15] presented the first polynomial time algorithm for computing the continuous Fréchet distance. The algorithm runs in $O((mn^2 + m^2 n) \log mn)$ time, and was later improved by Alt and Godau [2] into an algorithm with running time $O(mn \log mn)$. Both algorithms rely on an efficient solution for the decision variant of the problem, and then using parametric search [17] to solve the optimization problem. Around the same time, the discrete version of the problem was studied by Eiter and Mannila [13], who give an $O(mn)$ time algorithm.

Under the word RAM model of computation, these results have since been improved. Agarwal et al. [4] gave an $O(mn \log n / \log n)$ time algorithm for the discrete problem, and Buchin et al. [7] improved the complexity bound for the continuous problem to $O(mn(\log \log n)^2)$. Unfortunately, there is little hope of improving these results further, at least significantly, as Bringmann [4] showed that a strongly subquadratic (i.e., $(mn)^{1-O(1)}$) time algorithm would refute the Strong Exponential Time Hypothesis (SETH).

The main focus has hence been on efficient approximation algorithms and on specific classes of curves. Aronov et al. [6] studied the discrete Fréchet distance between $\kappa$-bounded curves and backbone curves, which model some “realistic” families of curves. They give a $(1+\epsilon)$-approximation algorithm running in strongly subquadratic time. Later on, Driemel et al. [12] gave efficient $(1+\epsilon)$-approximation algorithms for computing the continuous Fréchet distance between various realistic curves. These families of curves again include $\kappa$-bounded curves, but also $c$-packed curves. An improved algorithm for $c$-packed curves, which matches conditional lower bounds, was later given by Bringmann and Künnemann [5]. Gudmundsson et al. [16] give a $\sqrt{d}$-approximate algorithm for the case where the curves have sufficiently long edges (relative to their Fréchet distance) that runs in linear time. They give a near-linear time exact algorithm for this case as well.

When dealing with arbitrary curves, SETH again gives conditional lower bounds. Bringmann [4] not only gave a conditional lower bound for the exact problems, but also showed that a 1.001-approximation algorithm running in strongly subquadratic time would refute the SETH. This lower bound was later improved by Buchin et al. [8], who prove that assuming SETH, an approximation factor of 3 is the best we can hope for, even for curves in one dimension.

For computing the discrete Fréchet distance between arbitrary curves, Bringmann and Mulzer [6] gave an $O(\alpha)$-approximate algorithm with a running time of $O(n^2/\alpha)$, for any $\alpha \in [1, n/\log n]$. Chan and Rahmati [9] later gave an improved algorithm that returns an $O(\alpha)$-approximation in $O(n^2/\alpha^2)$ time, for any $\alpha \in [1, \sqrt{n}/\log n]$.

Bringmann and Mulzer [6] also gave an approximation algorithm for the continuous problem. The running time of this algorithm is linear, but the approximation ratio is exponential (i.e., $2^{\Theta(n)}$). This bound was improved only recently, by Colombe and Fox [10]. They give the best result of prior work: an $O(\alpha)$-approximate algorithm that runs in $O((n^3/\alpha^2) \log n)$ time, for any $\alpha \in [\sqrt{n}, n]$.

**Results.** We improve upon the result of Colombe and Fox [10]. We give an $O(\alpha)$-approximate algorithm that runs in $O((n+mn/\alpha)\log^3 n)$ time, for any $\alpha \in [1, n]$, if the dimension $d$ of the input curves is constant.
After introducing the necessary preliminaries in Section 1.1, we present an outline of both our algorithm and the paper in Section 1.2. Section 2 then presents our algorithm in detail; the technical Sections 3 and 4 contain further details for the two major subroutines used by our algorithm.

## 1.1 Preliminaries

**Polygonal curves.** A (polygonal) curve is a piecewise-linear function \( P : [0, 1] \to \mathbb{R}^d \), connecting a sequence \( p_1, \ldots, p_m \) of \( d \)-dimensional points, which we refer to as vertices. In this work, we consider curves where the dimension \( d \) is constant only. We will parameterize the domain \([0, 1]\) such that the points \( p_i \) are uniformly spaced over the domain (i.e. \( P((i - 1)/(m - 1)) = p_i \)). Note that the above definition allows for consecutive vertices to share the same position. The linear interpolation between \( p_i \) and \( p_{i+1} \), whose image is equal to the directed line segment \( p_i p_{i+1} \), is called an edge. We denote by \( P[x_1, x_2] \) the subcurve of \( P \) over the domain \([x_1, x_2]\). Similarly, we denote by \( P(x_1, x_2) \), the subcurve of \( P \) over the open domain \((x_1, x_2)\). We proceed to define concepts like concatenation and matchings for curves over a closed domain; their adaptations to curves over open domains can be made in the natural way.

For a point \( p \in \mathbb{R}^d \), we write \( p^\ell \) to denote its \( \ell \)-th coordinate. We extend this notation to curves, denoting by \( P^\ell : [0, 1] \to \mathbb{R} \) the curve where \( P^\ell(x) = P(x)^\ell \). Let \( P, Q \) be two polygonal curves with vertices \( p_1, \ldots, p_m \) and \( q_1, \ldots, q_n \), respectively. If \( p_m = q_1 \), we denote by \( P \circ Q \) the curve connecting \( p_1, \ldots, p_m = q_1, \ldots, q_n \).

**Fréchet distance.** A reparameterization is a monotonically increasing, continuous surjection \( f : [0, 1] \to [0, 1] \). Two orientation-preserving reparameterizations \( f, g \) describe a matching \((f, g)\) between two curves \( P \) and \( Q \), where any point \( P(f(t)) \) is matched to \( Q(g(t)) \). A matching \((f, g)\) between \( P \) and \( Q \) is said to have cost

\[
\max_i \|P(f(t)) - Q(g(t))\|_\infty.
\]

It is common to use the Euclidean norm \( \|P(f(t)) - Q(g(t))\|_2 \) instead, but for our purposes it is more convenient to use the \( L_\infty \)-norm. Since we aim for approximation factors between \( O(1) \) and \( O(n) \), and the norms are at most a factor \( \sqrt{d} \) different, approximations using the one norm imply the same approximations for the other norm, as long as \( d \) is considered constant. We refer to a matching with cost at most \( \delta \) as a \( \delta \)-matching. The Fréchet distance between \( P \) and \( Q \) is the minimum cost over all matchings.

**Free space diagram and matchings.** Let the square \( \mathcal{D}(P,Q) = [0, 1]^2 \) define the parametric space of \( P \times Q \). A point \((x, y) \in \mathcal{D}(P,Q)\) corresponds to the points \( P(x) \) and \( Q(y) \) on the two curves. For \( \delta \geq 0 \), a point \((x, y) \in \mathcal{D}(P,Q)\) is \( \delta \)-close if \( \|P(x) - Q(y)\|_\infty \leq \delta \). The \( \delta \)-free space \( \mathcal{F}_{\leq \delta}(P,Q) \) of \( P \) and \( Q \) is the subset of \( \mathcal{D}(P,Q) \) containing all \( \delta \)-close points.

A point \( z_2 = (x_2, y_2) \in \mathcal{F}_{\leq \delta}(P,Q) \) is \( \delta \)-reachable from a point \( z_1 = (x_1, y_1) \) if there exists a bimonotone path in \( \mathcal{F}_{\leq \delta}(P,Q) \) from \( z_1 \) to \( z_2 \). Points that are \( \delta \)-reachable from \((0,0)\) are simply called \( \delta \)-reachable points. Alt and Godau \([2]\) observe that the Fréchet distance between \( P[x_1, x_2] \) and \( Q[y_1, y_2] \) is at most \( \delta \) if and only if there is a bimonotone path in \( \mathcal{F}_{\leq \delta}(P,Q) \) from \( z_1 \) to \( z_2 \). We can therefore abuse terminology slightly and refer to a bimonotone path from \( z_1 \) to \( z_2 \) as a \( \delta \)-matching between \( P[x_1, x_2] \) and \( Q[y_1, y_2] \).

The column of \( P(x) \) is the set \( \{x\} \times [0, 1] \). A maximal vertical line segment \( \{x\} \times [y_1, y_2] \) in \( \delta \)-free space is called a \( \delta \)-passage in the column of \( P(x) \). Given a value \( \alpha \geq 1 \), an \((\alpha, \delta)\)-exit set of a point \((0,y)\) with respect to \( P \) and \( Q \) is a set \( E_\alpha(y) = \{1\} \times [0, 1] \) that (1) contains all points in the column of \( P(1) \) that are \( \delta \)-reachable from \((0,y)\), and (2) for which all \( \delta \)-close points in \( E_\alpha(y) \) are \( \alpha \delta \)-reachable from \((0,y)\). An \((\alpha, \delta)\)-exit set for \((0,0)\) with respect to \( P \) and \( Q \) is simply called an \((\alpha, \delta)\)-exit set with respect to \( P \) and \( Q \).
1.2 Algorithmic outline

Let $P,Q: [0,1] \to \mathbb{R}^d$ be our two input curves with $m$ and $n$ vertices, respectively, and let $\alpha \in [1,n]$ be the chosen parameter for the approximation. We describe an algorithm that solves the $O(\alpha)$-approximate decision problem. Here we are given a further parameter $\delta \geq 0$. Our algorithm answers YES if $d_F(P,Q) \leq c\alpha\delta$ for some constant $c$, and NO if $d_F(P,Q) > \delta$. That is, we either confirm that there is a $c\alpha\delta$-matching between $P$ and $Q$ or we assert that no $\delta$-matching exists. We use the same procedure as Colome and Fox \[10\] to turn this approximate decision algorithm into an approximation algorithm for the Fréchet distance (with logarithmic overhead in running time and an arbitrarily small increase in approximation ratio).

Recall that a $\delta$-matching between $P$ and $Q$ represents a bimonotone path from $(0,0)$ to $(1,1)$ in the $\delta$-free space $F_{\leq \delta}(P,Q)$. Our decision algorithm searches for such a path. The complexity of the free space can be as high as $\Theta(mn)$, so exploring it completely cannot structurally result in a subquadratic algorithm. We therefore identify so-called sparse signature columns in the $\delta$-free space $F_{\leq \delta}(P,Q)$. These columns correspond to signature vertices that can in some sense be matched to only $O(n/\alpha)$ locations on the other curve. Signature vertices were originally introduced by Driemel et al. \[13\]; we show in Section 2 how to generalize their definition for our purposes. Sparse signature columns have a limited number $(n/\alpha)$ of $\delta$-reachable passages.

Given two sparse signature columns, corresponding to signature vertices $P(s_i)$ and $P(s_{i+1})$, and a set $S(s_i)$ of $\delta$-close points in the column of $P(s_i)$, we compute an $O(\alpha, \delta)$-exit set $E_\alpha(z)$ for each point $z \in S(s_i)$; Figure 1 shows this for one point $z$ on the left. This computation is rather technical: it requires using a simplification of a part of $Q$ and replacing $P[s_i, s_{i+1}]$ by a monotone curve. Once we have done this for all $z \in S(s_i)$, we compute the union $E_\alpha(S(s_i))$ of these exit sets, which contains all points in column $P(s_{i+1})$ that are $\delta$-reachable from points in $S(s_i)$, and only points that are $O(\alpha\delta)$-reachable from points in $S(s_i)$. We construct the set $S(s_{i+1})$ by taking the bottom-most point of each connected component of the intersection between $E_\alpha(S(s_i))$ and the $\delta$-passages of column $P(s_{i+1})$; Figure 1 shows this in the middle. Once we have $S(s_{i+1})$, we can continue with the next part of $P$ between the sparse signature vertices $s_{i+1}$ and $s_{i+2}$. At the end of this iterative construction, if $(1,1)$ is an element of the final exit set that we compute, then there

![Figure 1: A visual summary of our algorithm; blue segments lie in δ-free space, green regions lie in O(αδ)-free space. Vertical bars show sparse signature columns P(s_i) and P(s_{i+1}). The blue segments are δ-passages; any matching passes through a blue segment in each column. The point z is a point in S(s_i) from where the algorithm explores to find an (O(α), δ)-exit set E_α(z) for z; the right side of the green triangle. In the middle figure the set E_α(S(s_i)) is illustrated. The green ticks on the right column form the set S(s_{i+1}). The last figure shows the parts of free space that are explored by our algorithm, as well as an O(αδ)-matching in purple.](image-url)
is a matching between $P$ and $Q$ of cost $O(\alpha \delta)$, otherwise no matching of cost at most $\delta$ exists. Figure 1 on the right shows an existing matching in purple.

We show in Section 2 how to compute the exit sets from any sparse signature column in $O(|P[s_i, s_{i+1}]|(n/\alpha) \log^2 n)$ time. Together they contain $O(n/\alpha)$ $\delta$-passages, so $S(s_i)$ and $S(s_{i+1})$ have complexity $O(n/\alpha)$. Constructing $S(s_{i+1})$ therefore takes $O(|P[s_i, s_{i+1}]|(n/\alpha) \log^2 n)$ time. This result depends on the technical Sections 3 and 4. There we show that if $S(s_i)$ contains no sparse signature columns, then we can compute an $O(\delta)$.

Recall that a $\delta$-matching between $P$ and $Q$ represents a bimonotone path from $(0,0)$ to $(1,1)$ in $\delta$-free space $\mathcal{F}_{\leq \delta}(P,Q)$. Our decision algorithm looks for such a path in $\delta$-free space. However, since the complexity of this free space can be as high as $\Theta(mn)$, exploring the entire free space may take $\Omega(mn)$ time. We therefore identify sparse signature columns in free space. We first repeat the (slightly reworded) definition of the $\delta$-signature of a curve in $\mathbb{R}$ by Driemel et al. [13]. Then we give an extension to $\mathbb{R}^d$ that we will use.

**Definition 1 ($\delta$-Signature [13])**. Let $\delta > 0$ and $P: [0, 1] \rightarrow \mathbb{R}$ be a curve. A $\delta$-signature of $P$ is a curve $\Sigma: [0, 1] \rightarrow \mathbb{R}$, whose vertices $P(s_1), \ldots, P(s_k)$ are defined by a series of values $0 = s_1 < \cdots < s_k = 1$, with the following properties if $k > 2$:

1. (non-degeneracy) For all $2 \leq i \leq k - 1$: $P(s_i) \notin P(s_{i-1})P(s_{i+1})$.

2. (approximately direction-preserving) For all $1 \leq i \leq k - 1$:
   - If $P(s_i) < P(s_{i+1})$, then for all $s, s' \in [s_i, s_{i+1}]$ with $s < s'$, $P(s) - P(s') \leq 2\delta$.
   - If $P(s_i) > P(s_{i+1})$, then for all $s, s' \in [s_i, s_{i+1}]$ with $s < s'$, $P(s') - P(s) \leq 2\delta$.

3. (minimum edge length)
   - For all $2 \leq i \leq k - 2$: $|P(s_{i+1}) - P(s_i)| > 2\delta$.
   - $|P(s_2) - P(s_1)| > \delta$ and $|P(s_k) - P(s_{k-1})| > \delta$.
4. (range) For all \(s \in [s_i, s_{i+1}]\) with \(1 \leq i \leq k-1:\)
   - If \(2 \leq i \leq k - 2\), then \(P(s) \in P(s_i)P(s_{i+1})\).
   - If \(i = 1\), then \(P(s) \in P(s_1)P(s_2) \cup [P(s_1) - \delta, P(s_1) + \delta]\).
   - If \(i = k - 1\), then \(P(s) \in P(s_{k-2}P(s_{k-1}) \cup [P(s_{k-1}) - \delta, P(s_{k-1}) + \delta]\).

   If \(k = 2\), then property 2 holds and the following version of property 4 for all \(s \in [0, 1]\):
   \(P(s) \in P(0)P(1) \cup [P(0) - \delta, P(0) + \delta] \cup [P(1) - \delta, P(1) + \delta]\).

The vertices of \(\Sigma\) are called \(\delta\)-signature vertices.

**Definition 2 (Higher dimensional \(\delta\)-signature).** Let \(\delta > 0\) and \(1 \leq \ell \leq d\), and let \(P : [0, 1] \to \mathbb{R}^d\) be a curve. An \((\ell, \delta)\)-signature of \(P\) is a curve \(\Sigma : [0, 1] \to \mathbb{R}^d\), whose vertices \(P(s_1), \ldots, P(s_k)\) are defined by a series of values \(0 = s_1 < \cdots < s_k = 1\) that also define a \(\delta\)-signature of \(P^\ell\). The vertices of \(\Sigma\) are called \((\ell, \delta)\)-signature vertices.

Driemel et al. [13] remark that a \(\delta\)-signature of a curve in \(\mathbb{R}\) always exists. By extension, it follows that an \((\ell, \delta)\)-signature of a curve in \(\mathbb{R}^d\) always exists as well, for any \(1 \leq \ell \leq d\).

Let \(1 \leq \ell \leq d\), and let \(\Sigma : [0, 1] \to \mathbb{R}^d\) be an \((\ell, \delta)\)-signature of \(P\), defined by the values \(0 = s_1 < \cdots < s_k = 1\). Let \(Q[y_1, y_2]\) be a subcurve of \(Q\) that contains at least one vertex of \(Q\). If \(Q[y_1, y_2]\) is maximal such that \(Q[y_1, y_2] \subseteq \{P(s_1) - \delta, P(s_1) + \delta\}\), then we call the passage in column \(P(s_1)\) corresponding to \(Q[y_1, y_2]\) a candidate \(\delta\)-passage. As we show in the following lemma, any \(\delta\)-matching between \(P\) and \(Q\) intersects a candidate \(\delta\)-passage in each signature vertex column.

**Lemma 1.** Let \(1 \leq \ell \leq d\) and \(\delta > 0\). Let \(P,Q : [0, 1] \to \mathbb{R}^d\) be two curves with \(d_F(P,Q) \leq \delta\). Any \(\delta\)-matching between \(P\) and \(Q\) intersects the column of any signature vertex of \(P\) in a candidate \(\delta\)-passage.

**Proof.** Let \((f, g)\) be a \(\delta\)-matching between \(P\) and \(Q\). Let \(s_1, \ldots, s_k\) be the values defining an \((\ell, \delta)\)-signature of \(P\), and let \(\sigma_i = P(s_i)\). We prove that \((f, g)\) matches every vertex \(\sigma_i\) to a point on an edge \(e\) of \(Q\) that has a vertex with \(x^\ell\)-coordinate within \([\sigma^\ell_i - \delta, \sigma^\ell_i + \delta]\). From this it immediately follows that \((f, g)\) intersects column \(P(s_i)\) in a candidate \(\delta\)-passage, namely the one corresponding to (a curve containing) \(e\).

Observe that \((f, g)\) is a \(\delta\)-matching between \(P^\ell\) and \(Q^\ell\). By definition of \((\ell, \delta)\)-signature, \(P^\ell(s_i)\) is a signature vertex of \(P^\ell\). Recall that \(P^\ell(s_i) = P(s_i)^\ell = \sigma_i^\ell\) by construction. We show that \(\sigma_i^\ell\) must be matched to a point on some edge \(e^\ell\) of \(Q^\ell\) that has a vertex in \([\sigma_i^\ell - \delta, \sigma_i^\ell + \delta]\). This means that \(\sigma_i\) is matched to a point on an edge \(e\) of \(Q\) that is projected onto \(e^\ell\), and which must therefore have a vertex with \(x^\ell\)-coordinate in \([\sigma_i^\ell - \delta, \sigma_i^\ell + \delta]\).

Abusing notation slightly, we write \(\sigma_i = \sigma_i^\ell\), as well as \(P^\ell = P\) and \(Q^\ell = Q\). We follow the proof of Lemma 3.5 of Driemel et al. [13], who show that if \(d_F(P^\ell, Q^\ell) \leq \delta\), then every signature vertex \(\sigma_i\) must have a vertex of \(Q\) close to it, and these vertices appear on \(Q\) in the order of \(i\). We augment their proof to prove the statement in the theorem.

We say that \(Q\) visits a vertex \(\sigma_i\) of \(\Sigma\) when it comes within distance \(\delta\) of \(\sigma_i\). For all \(i\), let \(Q(y_i)\) be a point matched to \(\sigma_i\) by \((f, g)\). Note that \(Q\) visits \(\sigma_{i-1}\), \(\sigma_i\), and \(\sigma_{i+1}\) in order, as they are also in this order along \(P\). For \(3 \leq i \leq k - 2\), the definition of \(\delta\)-signature gives us that \(|\sigma_i - \sigma_{i-1}| > 2\delta\) and \(|\sigma_{i+1} - \sigma_i| > 2\delta\). Also, it gives us that \(\sigma_i \notin [\sigma_{i-1}, \sigma_{i+1}]\). This means that \(Q\) must change direction between visiting \(\sigma_{i-1}\) and \(\sigma_{i+1}\).

Observe that \(Q\) cannot move more than distance \(\delta\) away from the edge \([\sigma_{i-1}, \sigma_{i+1}]\) before it has visited both \(\sigma_i\) and \(\sigma_{i+1}\), in order. This means that \(Q[y_{i-1}, y_{i+1}]\) either lies left of \(\sigma_i + \delta\) or right of \(\sigma_i - \delta\), depending on the positions of \(\sigma_{i-1}\), \(\sigma_i\), and \(\sigma_{i+1}\) relative to each other. This shows that any edge of \(Q\) that comes close to \(\sigma_i\), must have a vertex that lies within distance \(\delta\) of \(\sigma_i\). In particular, this includes any edge that \(\sigma_i\) is matched to. This completes the proof for \(i \in \{3, \ldots, k - 2\}\).
For $i = 2$, note that by the definition of the Fréchet distance, $Q(0)$ must be matched to $\sigma_1 = P(0)$. Like before, $Q$ has to visit $\sigma_3$ and $\sigma_5$ in order. Either $|Q(0) - \sigma_2| \leq \delta$, in which case the statement is true immediately, or $|Q(0) - \sigma_2| > \delta$. In the latter case, because $|\sigma_2 - \sigma_3| > 2\delta$, the proof is analogous to the case where $3 \leq i \leq k - 2$. The case $i = k - 1$ is symmetric.

Finally, by the definition of the Fréchet distance, $Q(0)$ and $Q(1)$ must be matched to $\sigma_1 = P(0)$ and $\sigma_k = P(1)$, respectively. □

We say that an $(\ell, \delta)$-signature vertex $\sigma$ is $\beta$-sparse if there are $\beta$ candidate $\delta$-passages in its column. We call $\beta$ the sparsity of $\sigma$. As we show in the following lemma, we can efficiently compute the sparsity of a signature vertex using a range tree (using fractional cascading to improve query times). We also show how to efficiently compute the candidate $\delta$-passages in the column of $\sigma$.

**Lemma 2.** Let $1 \leq \ell \leq d$ and $\delta > 0$. Let $P, Q : [0, 1] \rightarrow \mathbb{R}^d$ be two curves, where $Q$ has $n$ vertices. With $O(n \log n)$ time preprocessing we can preprocess $Q$ into a data structure of $O(n \log n)$ size that, given an $(\ell, \delta)$-signature vertex $\sigma$ of $P$, computes the sparsity $\beta$ of $\sigma$ in $O(\log n)$ time. In $O(\log n + \beta \log \beta)$ additional time, the data structure returns all $\beta$ candidate $\delta$-passages in the column of $\sigma$.

**Proof.** Let $q_1, \ldots, q_n$ be the $n$ vertices of $Q$, in their order along $Q$. Recall that a candidate $\delta$-passage in the column of a signature vertex $\sigma$ corresponds to a maximal subcurve $Q[y_i, y_{i+1}]$ of $Q$ that contains at least one vertex of $Q$ and for which $Q[y_i, y_{i+1}] \subseteq R$, where $R = [\sigma^\ell - \delta, \sigma^\ell + \delta]$. Note that each such subcurve is uniquely identified by a vertex $q_i$ on $Q$ with $q_i \in R$ and $q_{i+1} \notin R$ or $i = 1$. We can thus count the number of candidate $\delta$-passages in the column of $\sigma$ by counting the number of vertices with the above property. Symmetrically, each subcurve corresponding to a candidate $\delta$-passage is uniquely identified by a vertex $q_i$ on $Q$ with $q_i \in R$ and $q_{i+1} \notin R$ or $i = n$. This means that we can report the first and last vertices in the subcurves defining the candidate $\delta$-passages, and construct the passages from them.

We store the vertices of $Q$ in $d$ range trees $T_\ell$ for two-dimensional orthogonal range searching, for $\ell = 1, \ldots, d$. The tree $T_\ell$ stores the points $(q^\ell_{i-1}, q^\ell_i)$ for $i = 2, \ldots, n$, as well as the point $(-\infty, q^\ell_1)$. Together with the point $(q^\ell_{i-1}, q^\ell_i)$ (or $(-\infty, q^\ell_1)$), we store the vertex $q_i$ itself, as well as its index $i$. This index is used to report the candidate passages. Constructing these trees takes $O(n \log n)$ time and they use $O(n \log n)$ space.

To compute the sparsity $\beta$ of an $(\ell, \delta)$-signature vertex $\sigma$, we query $T_\ell$ with the ranges $(-\infty, \sigma^\ell - \delta) \times R$ and $R \times (\sigma^\ell + \delta, \infty)$, count the number of points inside the ranges and add up the reported counts. This query returns $\beta$ in $O(\log n)$ time.

We can report all first vertices on the $\beta$ subcurves defining candidate $\delta$-passages in $O(\log n + \beta)$ time using the above range trees. With an analogous construction of the trees, we can construct $d$ range trees $T'_\ell$ that report all last vertices on these subcurves in $O(\log n + \beta)$ time. The vertices are reported together with their indices. Hence we can sort the reported vertices according to their order along $Q$ in $O(\beta \log \beta)$ time. We can compute a suitable representation of the candidate $\delta$-passages from these sorted vertices with a single scan of the vertices, taking $O(\beta)$ additional time. Thus we can report the $\beta$ candidate $\delta$-passages in the column of $\sigma$ in $O(\log n + \beta \log \beta)$ time. □

### 2.2 The algorithm

We now describe the decision algorithm. First we identify $(n/\alpha)$-sparse signature vertices. Let $\Sigma_\ell$ be an $(\ell, \delta)$-signature of $P$, for $\ell = 1, \ldots, d$. Let $0 = s_1 < \cdots < s_{k+1} = 1$ be the maximal set of unique values such that for all $2 \leq i \leq k$, $P(s_i)$ is an $(n/\alpha)$-sparse signature vertex. We split $P$
Theorem 3. Let \( \alpha \) between its total length \( P \). First we show that the subcurves \( P_i \) can be constructed in \( O((m + n) \log n) \) time.

Theorem 3 returns a set of \((\ell, \delta)\)-signatures \( \Sigma^\ell(P_i) \) of \( P_i \) as well, for all \( \ell \) and \( i \). These signatures have the following useful property. For an \((\ell, \delta)\)-signature \( \Sigma \) with vertices \( \sigma_i \), let \( R(\Sigma) = \bigcup_i [\sigma_i^\ell - \delta, \sigma_i^\ell + \delta] \) be its \( \delta \)-footprint. Refer to Figure 2 for an illustration. The size of the \( \delta \)-footprint is its total length \( ||R(\Sigma)|| \). The signatures \( \Sigma^\ell(P_i) \) returned by the algorithm of Theorem 3 all have a \( \delta \)-footprint of size at most \( 4 \alpha \delta + 4 \delta \). In Section 4 we use this property to transform the curves \( P_i \) and \( Q \) into new curves, whose free space is in a sense an \( O(\alpha) \)-approximation for the free space between \( P_i \) and \( Q \).

**Theorem 3.** Let \( \delta > 0 \), and let \( P, Q : [0, 1] \to \mathbb{R}^d \) be two curves, with \( m \) and \( n \) vertices, respectively. Let \( \alpha \in [1, n] \). In \( O((m + n) \log n) \) time, we can split \( P \) into subcurves \( P_1 \circ \cdots \circ P_k = P \), such that the vertices joining consecutive subcurves are \((n/\alpha)\)-sparse signature vertices. For all \( \ell \) and \( i \), we also compute an \((\ell, \delta)\)-signature of \( P_i \) with a \( \delta \)-footprint of size at most \( 4 \alpha \delta + 4 \delta \).

**Proof.** First we show that the subcurves \( P_i \) can be constructed in \( O((m + n) \log n) \) time. Using the result of Driemel et al. [13], we can compute a \( \delta \)-signature of \( P^\ell \), for any \( \ell \), in \( O(m) \) time. This algorithm is easily extended to return an \((\ell, \delta)\)-signature \( \Sigma^\ell(P) \) of \( P \) in \( O(m) \) time. Thus, we can compute a set of \((\ell, \delta)\)-signatures of \( P \), for \( \ell = 1, \ldots, d \), in \( O(m) \) time total.

We construct the data structure of Lemma 2 on \( Q \). This takes \( O(n \log n) \) time, and the data structure uses \( O(n) \) space. For each \( \ell \), we query the data structure with each vertex \( \sigma \) of \( \Sigma^\ell(P) \) to obtain the sparsity of \( \sigma \). Computing the sparsity of all signature vertices takes \( O(m \log n) \) time in total. From here, we can construct the subcurves \( P_i \) in a single scan of \( P \). The total time taken to construct the subcurves is \( O((m + n) \log n) \).

Because the endpoints of each \( P_i \) are signature vertices, the subcurve \( \Sigma^\ell(P_i) \) of \( \Sigma^\ell(P) \) whose endpoints correspond to those of \( P^\ell \) is an \((\ell, \delta)\)-signature of \( P_i \). We argue that the \( \delta \)-footprint of each \( \Sigma^\ell(P_i) \) has size at most \( 4 \alpha \delta + 4 \delta \), completing the proof.

Consider a signature \( \Sigma^\ell(P_i) \) and let \( L \) be the size of its \( \delta \)-footprint. Let \( I \) be the set indexing the vertices of \( \Sigma^\ell(P_i) \). We will select a subset \( I' \) of \( I \) such that \( L \leq 2 \delta |I'| \), and each vertex of \( Q \) is \( \delta \)-close to at most two signature vertices indexed by \( I' \). One can construct such a subset \( I' \) by initializing \( I' \) to be the empty set and sweeping over \( R(\Sigma^\ell(P_i)) \) from left to right: whenever we sweep over a point \( p \) that is not close to any signature vertex of \( I' \), insert the index of the rightmost signature vertex that is \( \delta \)-close to \( p \) into \( I' \). This way, each point (and in particular each vertex of \( Q \)) is \( \delta \)-close to at most two signature vertices indexed by \( I' \). Because at most two signature vertices (namely the start and end of \( P_i \)) are \((n/\alpha)\)-sparse, all but two points of \( I' \) are \( \delta \)-close to more than \( n/\alpha \) points of \( Q \). By the pigeonhole principle we have \( (|I'| - 2)(n/\alpha + 1) \leq 2n \), and hence \( |I'| \leq 2 \alpha + 2 \). Because \( L \leq 2 \delta |I'| \), we obtain \( L \leq 2 \delta |I'| \leq 2 \delta (2 \alpha + 2) = 4 \alpha \delta + 4 \delta \).

We give the main algorithm in the following theorem. This algorithm uses a procedure for computing exit sets as a black box.
**Theorem 4.** Let $\delta > 0$. Given an algorithm that computes a constant-complexity $(O(\alpha), \delta)$-exit set with respect to $P$ and $Q$ in $O(T(m, n))$ time, there is an algorithm for the $O(\alpha)$-approximate decision problem that takes $O((n/\alpha)\sum_i T(m_i, n) + \log n) + (m + n) \log n)$ time, where $m_i$ is the complexity of $P_i$.

**Proof.** First compute the subcurves $P_i = P[s_i, s_{i+1}]$ as above with the algorithm of Theorem 3, taking $O((m + n) \log n)$ time. We solve the decision problem on $P$ and $Q$ iteratively by computing an $(O(\alpha), \delta)$-exit set at $P(s_{i+1})$ given an $(O(\alpha), \delta)$-exit set at $P(s_i)$. For subcurve $P_i = P[s_i, s_{i+1}]$, let $S(s_i)$ be a subset of $O(\alpha \delta)$-reachable points in the column of $P(s_i)$ that are all $\delta$-close. Initially, $S(0) = \{(0, 0)\}$ or $S(0) = \emptyset$, depending on whether $(0, 0)$ is $\delta$-close or not. Compute an $(O(\alpha), \delta)$-exit set $E_\alpha(z)$ for each point $z \in S(s_i)$ in total $O(|S(s_i)|T(m_i, n))$ time. Let $E_\alpha(S(s_i))$ be the union of all computed exit sets. Since the resulting exit sets each have constant complexity, their total complexity is $O(|S(s_i)|)$ and hence we can compute $E_\alpha(S(s_i))$ in $O(|S(s_i)| \log |S(s_i)|)$ time.

Report the $O(n/\alpha)$ candidate $\delta$-passages of column $P(s_{i+1})$ in $O(\log n + (n/\alpha) \log (n/\alpha))$ time using the data structure of Lemma 2. We extract the bottom-most point of each connected component of the intersection between $E_\alpha(S(s_i))$ and the reported passages. This takes $O(|E_\alpha(S(s_i))|) = O(n/\alpha)$ extra time. The set $S(s_{i+1})$ is the set containing all these points.

Computing $E_\alpha(S(s_i))$ for all $i$ takes $O\left(\sum_i |S_i|T(m_i, n) + \log |S_i|\right) = O\left((n/\alpha)\sum_i T(m_i, n) + \log n\right)$ time. Once $E_\alpha(S(s_k))$ is computed, return YES if $(1, 1) \in E_\alpha(S(s_k))$, and NO otherwise. The claimed running time comes from computing the subcurves $P_i$ and the exit sets $E_\alpha(S(s_i))$. \hfill $\square$

We show how to compute an $(O(\alpha), \delta)$-exit set efficiently in Sections 3 and 4. In Section 3 we generalize the result of Gudmundsson et al. \cite{16} on curves with a large minimum edge length, to work on quasi-piecewise $(> 4\delta)$-monotone curves. This gives a linear-time algorithm for computing a $(3, \delta)$-exit set when one curve is quasi-piecewise $(> 4\delta)$-monotone. In Section 4 we approximate two given curves $P$ and $Q$ by two curves $P^*$ and $Q^*$, such that $P^*$ is quasi-piecewise $(> 4\delta)$-monotone. This algorithm implies the following linear-time algorithm for computing exit sets between $P$ and $Q$:

**Lemma 5.** Let $\delta > 0$ and $P, Q: [0, 1] \to \mathbb{R}^d$ be two curves with $m$ and $n$ vertices, respectively. Given an $(\ell, \delta)$-signature of $P$ for all $\ell$, whose $\delta$-footprints have a maximum size of $L$, we can compute a constant-complexity $(O(L/\delta), \delta)$-exit set of a point in $O(m + n)$ time.

Plugging the above result into the black box of Theorem 4 with $T(m, n) = O(m + n)$, we obtain an $O(L/\delta) = O(\alpha)$-approximate decision algorithm that takes $O((n/\alpha)\sum_i (m_i + n) + \log n) + (m + n) \log n) = O(mn^2/\alpha + n \log n)$ time. As this is super-quadratic, we show in the next section how to lower this time complexity to strongly subquadratic (for $\alpha = \Omega(n^\delta)$).

### 2.3 Achieving subquadratic running time

The dominating factor in the running times of the subproblems comes from the linear dependency on $n$ in Lemma 5. To reduce the running time, we use simplifications $\text{simp}(Q)_i$ of $Q$ that have complexities roughly equal to those of the subcurves $P_i$. To obtain a simplification that bounds the approximation ratio sufficiently, we use the data structure of Driemel and Har-Peled \cite{11} Section 6.2. We refer to this data structure as the simplified subcurve tree.

The simplified subcurve tree $T(Q)$ can be constructed in $O(n \log^2 n)$ time and uses $O(n \log n)$ space. Given a subcurve $Q'$ of $Q$ and a parameter $k \in \mathbb{N}$, we can in $O(k \log n)$ time extract a simplification $\text{simp}_k(Q')$ of $Q'$ with $O(k \log n)$ vertices, as well as a value $\tau(Q', k)$ that upper bounds the distance $d_F(Q', \text{simp}_k(Q'))$ between $Q'$ and the simplification. Let $\rho(Q, k)$ be the
Fréchet distance from $Q$ to the closest curve with at most $k$ vertices. Driemel and Har-Peled [11] show that $d_F(Q', \text{simpl}_k(Q')) \leq \tau(Q', k) \leq 11\rho(Q', k)$.\footnote{Driemel and Har-Peled [11] prove this property only for the case where $Q'$ is stored in a single node of $T(Q)$. However, it follows from the proof of Lemma 6.8 [11] that this property holds for general subcurves of $Q$ as well.}

Let a subcurve $P_i$ have $m_i$ vertices. We can directly apply the simplified subcurve tree on $Q$ with parameter $k = m_i$ to obtain a curve with $O(m_i \log n)$ vertices. However, the induced error $\tau(Q, k)$ will generally be too large. To obtain meaningful bounds on the induced error, we instead opt to compute a simplification $\text{simpl}_k(Q_z)$ of a subcurve $Q_z = Q[y, y^*]$ that depends on the point $z = (0, y)$ for which we compute an exit set, such that the Fréchet distance between $Q_z$ and $\text{simpl}_k(Q_z)$ is not too large (relative to $\delta$). To not lose relevant information on the free space between $P_i$ and $Q$, $Q_z$ must contain every subcurve $Q[y, y']$ that has distance $\delta$ to $P_i$, so that the points that are $\delta$-reachable from $z$ will be represented in the free space between $P_i$ and $Q_z$.

In the following theorem, show how to compute the simplification $\text{simpl}_k(Q_z)$ of such a curve in $O(k \log^2 n)$ time, given $T(Q)$.

**Theorem 6.** The simplified subcurve tree $T(Q)$ supports $O(k \log^2 n)$ time queries with parameters $y_0 \in [0, 1]$, $\delta \geq 0$ and $k \in \mathbb{N}$ to retrieve a curve $\text{simpl}_k(Q[y_0, y^*])$ for some $y^* \geq y_0$ that has $\tau(Q[y_0, y^*], k) \leq 11\delta$ and the following property. Let $y_1$ be the maximum value such that $\rho(Q[y_0, y_1], k) \leq \delta$, then $y^* \geq y_1$.

**Proof.** Let $\{y \in [y_0, 1] \mid \tau(Q[y_0, y], k) \leq 11\delta\}$. Note that $[y_0, y_1] \subseteq T_{11\delta}(y_0, k)$, since $\tau(Q[y_0, y], k) \leq 11\rho(Q[y_0, y], k) \leq 11\rho(Q[y_0, y_1]) \leq 11\delta$ for any $y \in [y_0, y_1]$. Hence, if $y^*$ is the supremum of a connected component of $T_{11\delta}(y_0, k)$, then $y^* \geq y_1$. Since $\tau(Q[y_0, y'], k) \leq 11\delta$ for any $y' \in T_{11\delta}(y_0, k)$, we have $d_F(Q[y_0, y'], \text{simpl}_k(Q[y_0, y'])) \leq 11\delta$, so $\text{simpl}_k(Q[y_0, y^*])$ has Fréchet distance at most $11\delta$ to $Q[y_0, y^*]$. See Figure 3.

We now show how to compute the maximum $y^*$ of some connected component of $T_{11\delta}(y_0, k)$ using a search procedure. For $y \geq y_0$, we can query $T(Q)$ with the subcurve $Q[y_0, y]$ and parameter $k$ to obtain a simplification $\text{simpl}_k(Q[y_0, y])$ of $Q[y_0, y]$, as well as a value $\tau := \tau(Q[y_0, y], k)$. If $\tau > 11\delta$, then $11\delta < \tau \leq 11\rho(Q[y_0, y], k)$ and hence $\rho(Q[y_0, y], k) > \delta$, from which it follows that $y > y_1$. If instead $\tau \leq 11\delta$, then $d_F(Q[y_0, y], \text{simpl}_k(Q[y_0, y])) \leq \tau \leq 11\delta$, and thus $y \in T_{11\delta}(y_0, k)$. In the first case, we search for a lower value of $y$. In the latter case, we remember the value $y$ and search for a higher value of $y$ (possibly in a different connected component of $T_{11\delta}(y_0, k)$).

When continuously varying $y$, the value of $\tau(Q[y_0, t], k)$ changes only when $y$ indexes a vertex. This follows from the construction of the simplifications, see Theorem 6.9 of Driemel and Har-Peled [11]. Hence we can restrict the search to values $y$ that define vertices of $Q$. Bisection search on the vertices performs $O(\log n)$ queries on $T(Q)$ and computes an edge of $Q$ whose indexing interval $Y$ contains a valid value $y^*$ in $O(k \log^2 n)$ time. Since $\tau(Q[y_0, t], k)$ changes only when $y$ indexes a vertex, we get that $y^*$ indexes a vertex of $Q$ where $\tau(Q[y_0, y^*], k) > 11\delta$ and $\tau(Q[y_0, y^* - \varepsilon], k) \leq 11\delta$ for an arbitrarily small $\varepsilon > 0$. We take a small enough $\varepsilon' > 0$ such that $Q(y^* - \varepsilon')$ lies on an edge incident to $Q(y^*)$, and return $\text{simpl}_k(Q[y_0, y^*]) := \text{simpl}_k(Q[y_0, y^* - \varepsilon'], k) \cap Q[y^* - \varepsilon', y^*]$.\hfill \qed
We augment Lemma 5 to use the above result on the simplified subcurve tree. Given a \( \delta \)-close point \( z = (s_i, y) \) in the column of \( P(s_i) \), we first extract a curve \( \text{simpl}_k(Q_z) := \text{simpl}_k(Q[y, y^*]) \) from \( T(Q) \) according to Theorem 6. We then construct \((\ell, 12\delta)\)-signatures of \( P_i \) that have a small \( 12\delta \)-footprint. Note that the result of Driemel et al. [13] Lemma 7.1] gives a method to construct \((\ell, 12\delta)\)-signatures of \( P_i \) that use subsets of the vertices of \( \Sigma^\ell(P_i) \). This process takes \( O(m_i \log m_i) \) time per signature, after \( O(m_i \log m_i) \) preprocessing \([13] \) Lemma 7.5]. The maximum size of the \( 12\delta \)-footprints of these signatures is at most \( 12 \) times as high as the maximum size of the \( \delta \)-footprints of the original signatures, since the new signatures use a subset of the vertices. Thus, their \( 12\delta \)-footprints have size at most \( 12L = O(\alpha \delta) \).

We apply Lemma 5 on the curves \( P_i \) and simpl\((Q_z)\), with the point \( z' = (0, 0) \) and value \( \delta' = 12\delta \). This gives an \((O(\alpha), \delta')\)-exit set \( E_\alpha(z') \) for \( z' \) in \( O(m_i \log n) \) time. Let \((s_i, y') \) be a point in the column of \( P(s_i, y') \) that is \( \delta \)-reachable from \( z \). By Theorem 6 we have that \( y' \leq y^* \) and hence that \( d_F(Q[y, y'], Q_z[0, y^*]) \leq 11\delta \) for some \( y'' \in [0, 1] \). By the triangle inequality, the point \((1, y'') \) is therefore a \( 12\delta \)-reachable point in \( F_{\leq 12\delta}(P_i, \text{simpl}_k(Q_z)) \). Every point in \( F_{\leq \delta}(P_i, Q) \) that is \( \delta \)-reachable from \( z \) thus corresponds to a point in \( F_{\leq 12\delta}(P_i, \text{simpl}_k(Q_z)) \) that is \( 12\delta \)-reachable from \( z' \). Note that it follows from the triangle inequality that every point in \( F_{\leq 12\delta}(P_i, \text{simpl}_k(Q_z)) \) that is \( 12\delta \)-reachable from \( z' \) corresponds to a point in \( F_{\leq \delta}(P_i, Q) \) that is \( 23\delta \)-reachable from \( z \). Thus we have that \( E_\alpha(z') \) corresponds to an \((O(\alpha), \delta)\)-exit set for \( z \).

Note that the set \( E_\alpha(z') \) is represented as a single vertical line segment. This implies that \( E_\alpha(z) \) can be represented by a single vertical line segment as well. We obtain the following result:

**Lemma 7.** Let \( \delta > 0 \) and \( P, Q : [0, 1] \rightarrow \mathbb{R}^d \) be two curves with \( m \) and \( n \) vertices, respectively. Given an \((\ell, \delta)\)-signature of \( P \) for all \( \ell \), whose \( \delta \)-footprints have a maximum size of \( L \), we can compute a constant-complexity \((O(L/\delta), \delta)\)-exit set of a point in \( O(m \log^2 n) \) time.

Plugging the above lemma into the black box of Theorem 4 with \( T(m, n) = O(m \log^2 n) \) gives our main result: an \((O(L/\delta), O(\alpha))\)-approximate decision algorithm that takes \( O((m + n) \log n + (n/\alpha)(\sum_i m_i \log^2 n)) = O((mn/\alpha) \log^2 n + n \log n) \) time, given \( T(Q) \). We plug this decision algorithm into the optimization procedure of Colomb and Fox [10], to turn it into an algorithm for computing the Fréchet distance. This does come at the cost of a logarithmic factor in running time, as well as a constant factor in the approximation ratio. Together with the \( O(n \log^3 n) \) preprocessing time to construct \( T(Q) \), we obtain the following result:

**Theorem 8.** Let \( P, Q : [0, 1] \rightarrow \mathbb{R}^d \) be two curves, with \( m \) and \( n \) vertices respectively, and let \( \delta > 0 \) and \( \alpha \in [1, n] \) be parameters. Then we can compute an \( O(\alpha) \)-approximation to \( d_F(P, Q) \) in \( O((n + mn/\alpha) \log^3 n) \) time.

### 3 Handling piecewise monotone curves

In this section we present an algorithm for computing a \((3, \delta)\)-exit set between a quasi-piecewise \((> 4\delta)\)-monotone curve \( P \) (see Definition 3) and an arbitrary curve \( Q \) in \( O(m + n) \) time.

**Definition 3 (Piecewise Monotone Curves).** Let \( P : [0, 1] \rightarrow \mathbb{R}^d \) be a curve. \( P \) is monotone if its projection on each of the \( d \) coordinate axes is monotonically increasing or decreasing. \( P \) is \((> \delta)\)-monotone if moreover \( \|P(0) - P(1)\|_\infty > \delta \). \( P \) is piecewise \((> \delta)\)-monotone if it is composed of \((> \delta)\)-monotone curves. \( P \) is quasi-piecewise \((> \delta)\)-monotone if it is the composition of a piecewise \((> \delta)\)-monotone curve with a monotone curve (so the last piece is not necessarily \((> \delta)\)-monotone).

The following two observations indicate that monotone curves behave similar to line segments.
Observation 1. Any $\delta$-ball of the $L_\infty$ norm intersects any monotone curve in at most one subcurve. So if $P$ is monotone, any horizontal line intersects the $\delta$-freespace of $P$ and $Q$ in a convex set.

Observation 2. If $P$ is monotone and $x \leq x'$, then $\|P(x) - P(1)\|_\infty \geq \|P(x') - P(1)\|_\infty$.

We now generalize the definition of the longest $\delta$-prefix \cite{16}.

Definition 4 (Longest $\delta$-Prefix). The longest $\delta$-prefix of a curve $Q$ with respect to a curve $P$, is the largest subcurve $Q[0, y]$ of $Q$ for which $d_F(P, Q[0, y]) \leq \delta$.

Note that the longest $\delta$-prefix of $Q$ with respect to $P$ exists if and only if $d_F(P, Q[0, y]) \leq \delta$ for some $y \in [0, 1]$.

3.1 The structure of an exit set

For the lemmas in this section, let $P, Q : [0, 1] \to \mathbb{R}^d$ be two curves, where $P$ is a quasi-piecewise (>4$\delta$)-monotone curve with $k$ pieces $P[x_0, x_1], \ldots, P[x_{k-1}, x_k]$. If $d_F(P[0, x_i], Q[0, y]) \leq \delta$ for some $y \in [0, 1]$, define $y_i$ such that $Q[1, y_i]$ is the longest $\delta$-prefix of $Q$ with respect to $P[0, x_i]$.

Observation 3. If $y_k$ is undefined, then the empty set is a $(1, \delta)$-exit set.

We prove in Lemma \cite{10} that a specific $(3, \delta)$-exit set can be represented as an empty set or an interval. For this, we first show in Lemma \cite{9} that if $P$ has the stronger property that it is piecewise (>4$\delta$)-monotone, then a specific $(1, \delta)$-exit set can be represented as an empty set, or an interval. We then show it Lemmas \cite{12} and \cite{13} how to compute the respective intervals in linear time. To compute the exit sets for piecewise (>4$\delta$)-monotone curves, we take advantage of the property (shown below) that any $\delta$-reachable point $(x_{i+1}, y)$ is reachable by a monotone path from $(x_i, y_i)$.

Lemma 9. Let $P$ be piecewise (>4$\delta$)-monotone. Assume that $y_k$ is defined and let $y_k^- \in [y_{k-1}, y_k]$ be the minimum value such that $y_k^- \geq y_{k-1}$ and $\|P(1) - Q(y_k^-)\|_\infty \leq \delta$. Then

(a) any $\delta$-reachable point $(1, y)$ has $y \geq y_{k-1},$

(b) for $y \in [y_{k-1}, y_k]$, if $(1, y)$ is $\delta$-close, then $(1, y)$ is $\delta$-reachable, and

(c) for $y \in [y_k^-, y_k]$, $(1, y)$ is $3\delta$-reachable.

Note that properties (a) and (b) simply state that $\{x_k\} \times [y_k^-, y_k]$ is a $(1, \delta)$-exit set.

Proof. If $y_k$ is defined, then so is $y_{k-1}$. We prove the properties (a-c) by induction on the number of pieces $k$. The proof for the base case $k = 1$ is a special case of $k > 0$, so we first prove the case $k > 1$ assuming that the lemma holds for curves with fewer than $k$ pieces.

Let $[y^*, y_{k-1}]$ be the exit set obtained by applying the lemma to first $k-1$ pieces of $P$. Let $(1, y)$ be any $\delta$-reachable point. Because there is no $\delta$-reachable point below $(x_{k-1}, y^*)$ and matchings are monotone, we have $y \geq y^*$. Assume for a contradiction that $y \in [y^*, y_{k-1}]$. By induction, $(x_{k-1}, y)$ is $3\delta$-reachable, so $(x_{k-1}, y)$ is $3\delta$-close. By (>4$\delta$)-monotonicity, we have $\|P(x_{k-1}) - P(x_k)\|_\infty > 4\delta$, so by triangle inequality,

$$\|P(x_k) - Q(y)\|_\infty \geq \|P(x_{k-1}) - P(x_k)\|_\infty - \|P(x_{k-1}) - Q(y)\|_\infty > 4\delta - \delta \geq 3\delta,$$

contradicting that $(1, y)$ is $\delta$-close, and hence that it is $\delta$-reachable, proving property (a) for $k > 1$. For $k = 1$, for $y \leq y_{k-1} = y_0$, because $(x_{k-1}, y)$ lies on the path that $\delta$-reaches $(x_0, y_0)$, we have that $(x_{k-1}, y)$ is $3\delta$-close. Now the rest of the proof of (a) for $k > 1$ also applies to $k = 1$.  

11
We now prove properties (b) and (c). These are trivial for \( k = 1 \), so let \( k > 1 \). Because \( Q[0, y_k] \) is a \( \delta \)-prefix with respect to \( P[1, x_k] \), there exists a monotone path \( \pi \) in \( \delta \)-free space from \((0, 0)\) to \((1, y_k)\). For any \( y \in [y_{k-1}, y_k] \), because \( \pi \) does not pass through \( \{x_{k-1}\} \times (y_{k-1}, 1] \), there exists a point \((x, y)\) on \( \pi \) with \( x \in [x_{k-1}, x_k] \).

By Observation 1 if \( \|P(1) - Q(y)\|_\infty \leq \delta \), then the horizontal segment connecting \((x, y)\) to \((x_k, y)\) lies in \( \delta \)-free space, which together with the part of \( \pi \) until \((x, y)\) shows that \((x_k, y)\) is \( \delta \)-reachable, proving property (b).

For property (c), let \( h^- \) be a horizontal segment in \( \delta \)-free space that ends at \((x_k, y^-_k)\) and starts at a point \((x^-, y^-_k)\) on \( \pi \). For any \( y \in [y^-_k, y_k] \), there exists a point \((x, y)\) on \( \pi \) with \( x \in [x^-, x_k] \). Now \((x, y), (x, y^-_k), (x_k, y^-_k)\) all lie in \( \delta \)-free space, so by triangle inequality \((x_k, y)\) lies in \( 3\delta \)-free space. Hence, by Observation 1 the segment between \((x, y)\) and \((x_k, y)\) lies in \( 3\delta \)-free space, so it is \( 3\delta \)-reachable using the part of \( \pi \) up to \((x, y)\) and the horizontal segment, proving property (c).

For the following lemma, we define \( y_{k-2} = 0 \) for the special case that \( k = 1 \).

**Lemma 10.** Let \( P \) be quasi-piecewise \((> 4\delta)\)-monotone. Assume that \( y_{k-1} \) is defined. Let \( y^-_{k-1} \in [y_{k-2}, y_{k-1}] \) be the minimum value such that \( y^-_{k-1} \geq y_{k-2} \) and \( \|P(x_{k-1}) - Q(y^-_{k-1})\|_\infty \leq \delta \). Let

\[
y^+ = \begin{cases} y_k & \text{if } y_k \text{ is defined and } y_k > y_{k-1}, \\ y_{k-1} & \text{otherwise.} \end{cases}
\]

Then \( \{x_k\} \times [y^-_{k-1}, y^+] \) is a \((3, \delta)\)-exit set.

**Proof.** Let \( y \in [y^-_{k-1}, y^+] \) and let \((1, y)\) be \( \delta \)-close. It suffices to show that \((1, y)\) is \( 3\delta \)-reachable. First consider the case that \( y \leq y_{k-1} \), then by Lemma 9 the point \((x_{k-1}, y)\) is \( 3\delta \)-reachable (or in the special case that \( k = 1 \), it is even \( \delta \)-reachable), so if \((x_k, y)\) is \( \delta \)-close, by Observation 1 we can \( 3\delta \)-reach it via the horizontal segment between \((x_{k-1}, y)\) and \((x_k, y)\).

Now consider the case that \( y > y_{k-1} \), then because \( y^+ \geq y \), so \( y^+ \neq y_{k-1} \) and hence \( y^+ = y_k \). Let \( \pi \) be a monotone path in \( \delta \)-freespace from \((0, 0)\) to \((1, y_k)\). Because \( \pi \) does not pass through \( \{x_{k-1}\} \times (y_{k-1}, 1] \), \( \pi \) contains a point \((x, y)\) with \( x \in [x_{k-1}, x_k] \). So by Observation 1 \((1, y)\) is \( \delta \)-reachable via \( \pi \) and the horizontal segment between \((x, y)\) and \((1, y)\).

We show how to compute the exit sets of Observation 3 and Lemmas 9 and 10 in linear time. The main idea of Lemma 9 is to greedily walk through the free space to incrementally compute exit sets of \( Q \) with respect to \( P[0, x_i] \) for increasing \( i \), until either the exit set is empty, or \( i = k \). The algorithm for quasi-piecewise \((> 4\delta)\)-monotone curves differs from that of piecewise \((> 4\delta)\)-monotone curves only in the last step. Figure 4 illustrates the general algorithm for a quasi-piecewise \((> 4\delta)\)-monotone curve, where the part of the figure left of \( x_{k-1} \) coincides with the algorithm for a piecewise \((> 4\delta)\)-monotone curve. To bound the running time, we linearly bound the running time for iteration \( i \) by the involved parts of \( P \) and \( Q \), and show that any part of \( P \) and \( Q \) is involved in at most a constant number of iterations.

**Lemma 11.** Let \( P \) be piecewise \((> 4\delta)\)-monotone. We can in \( O(m+n) \) time compute the \((1, \delta)\)-exit set \( \{x_k\} \times [y^-_k, y_k] \) defined in the statement of Lemma 9 or return the empty set if it does not exist.

**Proof.** Given \([y^-_{i-1}, y^-_i]\) as defined by Lemma 9 (where \( y^-_0 = 0 \)), let \( \hat{y}^-_i \) be the minimum value in \([y^-_{i-1}, 1]\) such that \( \|P(x_i) - Q(\hat{y}^-_i)\|_\infty \leq \delta \). If no such \( \hat{y}^-_i \) exists, we return the empty set as exit set.

The notation in this proof is illustrated in Figure 4. Let \( \hat{y}^+_i \geq \hat{y}^-_i \) be the maximum value such that \((x_i, y)\) is \( 3\delta \)-close for all \( y \in [\hat{y}^-_i, \hat{y}^+_i] \). By Lemma 9 the vertical segment from \((x_i, \hat{y}^-_i)\) to \((x_i, \hat{y}^+_i)\) contains all the \( \delta \)-reachable points at \( x_i \). To figure out if any of them are actually reachable,
we greedily construct a monotone walk $\gamma_i$ starting at the point $(x_{i-1}, y_{i-1})$. Let $\gamma_i$ greedily walk upwards through the $\delta$-freespace, advancing rightwards only when needed, and stopping only when it reaches some $(x, y^+)$, some $(x, y)$, or when we can no longer extend it. Suppose that this walk ends at $(x^*, y^*)$. Then for all $y \leq y^*$, if $(x, y)$ is $\delta$-close, then $(x, y)$ is $\delta$-reachable using Observation [1]. We now show that if $y > y^*$, then $(x, y)$ is not $\delta$-reachable. Indeed, if it were $\delta$-reachable by a path $\pi$, then $\pi$ contains a point $(x, y^*)$ with $x \in [x^*, x_i]$. If $x > x^*$, then Observation [1] contradicts that $\gamma_i$ stopped at $(x^*, y^*)$ because it can be extended to the right. Similarly if $x = x^*$, then $\gamma_i$ can be extended by following $\pi$. Hence, if $\hat{y}_i^- \leq y^*$, then $\hat{y}_i^- = y_i^-$ and $y^* = y_i$, as defined with respect to $P[0, x_i]$ by Lemma [9]. On the other hand, if $\hat{y}_i^- > y^*$, then we correctly return the empty exit set. Iterating this procedure for all $i$ (or until we return the empty exit set for some $i$), we compute the required exit set.

To help analyze the running time, we show that if $y_{i+1}$ is defined, then $y_{i+1} > \hat{y}_i^+$. Indeed, by definition, $\{x_i\} \times [\hat{y}_i^-, \hat{y}_i^+]$ lies in $3\delta$-freespace, and because by $(> \delta)$-monotonicity, $P(x_i)$ and $P(x_{i+1})$ differ by $4\delta$, $\{x_{i+1}\} \times [\hat{y}_i^-, \hat{y}_i^+]$ does not intersect $\delta$-freespace, so $y_{i+1} > \hat{y}_i^+$. The computation of $\hat{y}_i^-$ and $\hat{y}_i^+$ takes time linear in the number of vertices of $Q[y_{i-1}, \hat{y}_i^+]$. The computation of $\gamma_i$ takes time linear in the number of vertices of $P$ and $Q$ it passes through. $\gamma_i$ passes through vertices of $P[x_{i-1}, x_i]$ only, so the contribution of vertices of $P$ over all paths is linear. Similarly, because $y_{i+1} > \hat{y}_i^+$, the computations for $i$ and $i'$ charge different vertices of $Q$ if $|i - i'| \geq 2$, so every vertex is charged at most a constant number of times, bounding the running time by $O(n + m)$. \hfill $\Box$

If the last piece of $P$ is not $(> 4\delta)$-monotone, then it is possible that $y_k < y_{k-1}$ (as in Figure [4]), so the last step of the algorithm for quasi-piecewise $(> 4\delta)$-monotone curves is slightly more involved.

**Lemma 12.** Let $P$ be quasi-piecewise $(> 4\delta)$-monotone. We can in $O(m + n)$ time compute the $(3, \delta)$-exit set $\{x_k\} \times [y_{k-1}, y^+]$ defined in the statement of Lemma [9] or return the empty set if it does not exist.

**Proof.** Using Lemma [11], compute the $(1, \delta)$-exit set $\{x_{k-1}\} \times [y_{k-1}, y_{k-1}]$ with respect to the piecewise $(> 4\delta)$-monotone curve $P[0, x_{k-1}]$ and $Q$ (or return the empty set if it does not exist). It

---

**Figure 4:** An illustration of the algorithm for a quasi-piecewise monotone curve $P$. 

---
remains to compute $y^+$, for which we recall the definition

$$y^+ = \begin{cases} y_k & \text{if } y_k \text{ is defined and } y_k > y_{k-1}, \\ y_{k-1} & \text{otherwise.} \end{cases}$$

If $y_k$ is defined and $y_k \geq y_{k-1}$, then the algorithm for computing $\gamma_k$ from Lemma 11 will find it, in which case $y^+ = y_k$. If it is not defined or $y_k < y_{k-1}$, then the algorithm for computing $\gamma_k$ will return the empty set, in which case $y^+ = y_{k-1}$. The computations of $y_{k-1}$ and $\gamma_k$ both take $O(m + n)$ time.

We summarize Lemma 12 in Theorem 13.

**Theorem 13.** Let $\delta > 0$ and $P, Q : [0, 1] \to \mathbb{R}^d$ be two curves, where $P$ is quasi-piecewise ($> 4\delta$)-monotone. Let $m$ and $n$ be the number of vertices of $P$ and $Q$, respectively. We can compute a $(3, \delta)$-exit set with respect to $P$ and $Q$ in $O(m + n)$ time.

## 4 Constructing piecewise monotone curves with small error

Let $P, Q : [0, 1] \to \mathbb{R}^d$ be two polygonal curves with $m$ and $n$ vertices, respectively. In this section we show how to approximate $P$ and $Q$ with two curves $P^*$ and $Q^*$, where $P^*$ is quasi-piecewise ($> 4\delta$)-monotone. We can then apply Theorem 13 on these curves to compute an exit set with respect to $P$ and $Q$.

The algorithm assumes that we are given a set of $(\ell, \delta)$-signatures $\Sigma^\ell(P)$ of $P$, for $\ell = 1, \ldots, d$. The approximation quality of the algorithm depends on the maximum size of the $\delta$-footprints of these signatures. Recall from Section 2 that the $\delta$-footprint of an $(\ell, \delta)$-signature $\Sigma^\ell(P)$ is the region $R(\Sigma^\ell(P)) = \bigcup_i [\sigma_i^\ell - \delta, \sigma_i^\ell + \delta]$. Throughout this section, we let $L$ be the size of the largest $\delta$-footprint of any of the given signatures. Note that any signature contains at least one point, and hence we have $L \geq 2\delta$.

We construct $P^*$ and $Q^*$ by considering each dimension $\ell$ separately, and applying point-wise transformations to the $x^\ell$-coordinates of the points. These transformations induce transformations on the original curves $P$ and $Q$ that are independent of each other. We approximate $P^\ell$ and $Q^\ell$ with two curves $P^{*\ell}$ and $Q^{*\ell}$, which form the projections of $P^*$ and $Q^*$, such that $P^{*\ell}$ has edges of length greater than $4\delta$ only. If we have this property for all $\ell$, then $P^*$ is quasi-piecewise ($> 4\delta$)-monotone. Of course, we need that the distance between $P^*$ and $Q^*$ is an approximation for the distance between $P$ and $Q$. Specifically, if $d_F(P, Q) \leq \delta$, then we require $d_F(P^*, Q^*) \leq \delta$, and if $d_F(P, Q) > cL\delta$ for some $c$, we require $d_F(P^*, Q^*) > \delta$. Under the $\ell_\infty$ norm, this is achieved if it holds for all projections of $P^*$ and $Q^*$.

In the remainder of this section, we construct $P^{*\ell}$ and $Q^{*\ell}$ for a given dimension $\ell$. For ease of notation, we write $P = P^\ell$ and $Q = Q^\ell$, and also write $P^* = P^{*\ell}$ and $Q^* = Q^{*\ell}$. The $(\ell, \delta)$-signature $\Sigma^\ell(P)$ induces a $\delta$-signature $\Sigma$ for the curve $P^\ell$. It follows from the definition of footprints that the $\delta$-footprint of $\Sigma$ is equal to that of $\Sigma^\ell(P)$. Hence, the $\delta$-footprint of $\Sigma$ has size at most $L$.

### 4.1 Splitting up the curves

We transform $P$ and $Q$ by first partitioning $P$ and $Q$ into subcurves with favorable properties. In particular, these subcurves are very restricted in how they can be matched to each other, allowing us to apply the transformations that turn the subcurves into line segments. Since the subcurves have diameter greater than $4\delta$, these line segments will have length greater than $4\delta$ as well. To partition $P$, after which we partition $Q$, we define a *bridge*. 
Lemma 14. Given a signature curve $\Sigma$ of $P$, we can decompose $P$ and $Q$ into the subcurves $CP_i$, $BP_i$, $CQ_i$, and $BQ_i$ in $O(m + n)$ time.

The constructed subcurves have several useful properties. The compact curves $CP_i$ have small diameter, and the bridge curves $BP_i$ are $2\delta$-direction-preserving and have sufficiently large diameter.
We define a direction-preserving curve based on the definition of \(\delta\)-signatures by Driemel et al. [13], who require \(\delta\)-signatures to be direction-preserving.

**Definition 6 (\(\delta\)-Direction-Preserving).** Let \(\delta > 0\). A curve \(P: [0, 1] \rightarrow \mathbb{R}\) is \(\delta\)-direction-preserving in the positive direction if the following holds. Let \(0 = x_1 < \cdots < x_m = 1\) index the vertices of \(P\). Then \(P(x) - P(x') \leq \delta\) for all \(x, x' \in [x_i, x_{i+1}]\) with \(x < x'\). A \(\delta\)-direction-preserving curve in the negative direction is defined analogously.

The following two lemmas prove the properties of the compact and bridge curves. Lemma 15 states that each compact curve has diameter at most \(4L\). Lemma 16 states that the bridge curves are all \(2\delta\)-direction-preserving, and have a minimum diameter greater than \(4\delta\). The small diameter of the compact curves means that we can essentially ignore the entire subcurve, without incurring too much error in the approximation. The sufficiently large diameter of the bridge curves allow us to construct a curve \(P^*\) that has minimum edge length greater than \(4\delta\).

**Lemma 15.** Each \(CP_i\) has diameter at most \(4L - 2\delta\).

*Proof.* We constructed \(CP_i\) by traversing \(P_i\) over its signature vertices, and based on the distance between a signature vertex and the current curve, we either expand the curve to contain this signature vertex, or we stop expanding the curve. In both cases, the diameter of the curve grows by at most \(8\delta\) (in the latter case, we grow the curve slightly before we stop expanding it).

Let \(R(\Sigma)\) be the \(\delta\)-footprint of \(\Sigma\). Recall that the size of \(R(\Sigma)\) is at most \(L\). As each connected component of \(R(\Sigma)\) has size at least \(2\delta\), there can not be more than \(L/(2\delta)\) connected components in \(R(\Sigma)\). During construction of \(CP_i\), the curve always ends at a vertex of \(\Sigma\) (except possibly when the construction is done). We therefore have that each expansion of the curve either keeps the endpoint of the curve in the same connected component of \(R(\Sigma)\), or it jumps to another connected component. In the first case, the total expansion the curve can do is bounded by the size of \(R(\Sigma)\), which is \(L\). In the second case, the total expansion is bounded by \(8\delta\) times the number of connected components of \(R(\Sigma)\), which is at most \(L/(2\delta)\). This gives \(CP_i\) a diameter of at most \(4L\). Note that \(CP_i\) does not come strictly within distance \(\delta\) of the leftmost and rightmost points of \(R(\Sigma)\), as there are no signature vertices there. Thus we can subtract \(2\delta\) from the bound, giving \(CP_i\) a diameter of at most \(4L - 2\delta\). \(\Box\)

**Lemma 16.** Each \(BP_i\) is \(2\delta\)-direction-preserving and has diameter greater than \(4\delta\).

*Proof.* The curve \(BP_i\) is a subcurve of \(P(s_j, s_{j+1})\), where \(P(s_j)\) and \(P(s_{j+1})\) are two consecutive vertices of \(\Sigma\). This directly implies that \(BP_i\) is \(2\delta\)-direction-preserving. Because \(BP_i\) is constructed by taking a subcurve \(P[x, s_{j+1}]\) of \(P[s_j, s_{j+1}]\) that has diameter greater than \(\delta\), of which we set \(BP_i = P[x', s_{j+1}]\) such that \(P[x, x']\) has diameter \(4\delta\), it follows that the diameter of \(BP_i\) is greater than \(4\delta\). \(\Box\)

Transforming two curves \(P\) and \(Q\) is usually a global process; if nothing is known about what a point \(p \in P\) can and cannot be matched to, we cannot move \(p\) without potentially having to transform all of \(Q\) (and all of \(P\) because of this). The subcurves \(CP_i\), \(BP_i\), \(CQ_i\) and \(BQ_i\) are very restricted in what they can be matched to however, allowing us to make local changes to the curves. In Theorem 17 we give the various restrictions on matchings. We use these restrictions in Section 4.2 to apply local transformations to \(P\) that allow us to construct the transformations based on local properties. This is key in computing a curve \(P^*\) that has long edges.

**Theorem 17.** Assume that \(d_F(P, Q[0, y]) \leq \delta\) for some \(y \in [0, 1]\) and let \((f, g)\) be a \(\delta\)-matching between \(P\) and \(Q[0, y]\). Then \((f, g)\) matches the various subcurves to each other with the following restrictions:}
CP\i is matched to a subcurve of CQi, for all 1 ≤ i ≤ k.

BP\i is matched to a subcurve of CQi ∘ BQi ∘ CQi+1 for all 1 ≤ i ≤ k − 1.

Proof. Without loss of generality, we assume that BP\i and BQi exist. We give a proof by induction over i. First we prove the base case for CP\i. By construction, there is an ε > 0 such that the distance between any point on CP\1 and any point on BQi(0, ε] is greater than δ. Since P(0) = CP\1(0) must be matched to Q(0) = CQi(0), it follows that CP\1 must be matched to a subcurve of CQi, proving the base case for CP\i.

We now prove the base case for BP\i. Say there is a point p = BP\1(x) that is matched to CQi(0). Assume without loss of generality that BP\1 moves to the right. Then because BP\1 is 2δ-direction-preserving (Lemma 16), we have that BP\1[x, 1] ⊆ [p − 2δ, CP\2(0)] ⊆ [CP\2(0) − 4δ, CP\2(0)]. Note that |CQi(1) − CP\2(0)| ≥ 5δ, and therefore that there is an ε > 0 such that all points on BQi(0, ε] have distance greater than 5δ to CP\2(0). We obtain that no point on BQi(0, ε] can be matched to points on BP\1[x, 1]. It follows that points on BP\1 cannot be matched to points on BQi, and thus that BP\1 is matched to a subcurve of CQi ∘ BQi ∘ CQi+1. This proves the base case for BP\i.

We now prove the induction step, starting with CP\i. Similar to the base case, we have that there is an ε > 0 such that the distance between any point on CP\i and any point on BQi(0, ε] is greater than δ. It follows that CP\i must either be matched to a subcurve of Q\i−1 ∘ CQi or a subcurve of BQi(ε, 1] ∘ Q\i+1. By the induction hypothesis, BP\i−1 must be matched to a subcurve of CQi ∘ BQi−1 ∘ CQi+1. Hence, CP\i must be matched to a subcurve of CQi−1 ∘ BQi−1 ∘ CQi, as otherwise BQi(0, ε] would not be matched to anything. Note that BQi(0) does not lie close to any point on CQi+1 ∘ BQi−1. We get that CP\i must be matched to a subcurve of CQi, proving the induction step for CP\i.

We conclude the proof by proving the induction step for BP\i. Through similar reasoning as for the base case, it can be shown that if a point on BP\i is matched to CQi+1(0), then no point on BP\i can be matched to points on BQi+1. By the induction hypothesis, CP\i is matched to a subcurve of CQi. This means that either BP\i is matched to a subcurve of CQi, or there is a point on BP\i that is matched to CQi+1(0). In both cases, it holds that points on BP\i cannot be matched to points on BQi+1, from which it follows that BP\i is matched to a subcurve of CQi ∘ BQi ∘ CQi+1. This proves the induction step for BP\i.

4.2 Transforming the subcurves

We alter the curves CP\i, BP\i, CQi and BQi, so that we can concatenate them into two curves P* and Q* with the desired properties. Recall that we want P* to have edges that all have length greater than 4δ. Also, we want the Fréchet distance between P* and Q* to be approximately equal to the Fréchet distance between P and Q, so that we can use P* and Q* to compute an exit set with respect to P and Q.

We make the assumption that BQi is 4δ-direction-preserving. Observe that if it is not, then no subcurve Q[0, y] of Q exists with dF(P, Q[0, y]) ≤ δ, meaning that the empty set is a (1, δ)-exit set. Indeed, by Theorem 17, any δ-matching matches BQi to a subcurve of BP\i. Because BP\i is 2δ-direction-preserving (Lemma 16), it follows that BQi must be 4δ-direction-preserving for a subcurve Q[0, y] with dF(P, Q[0, y]) ≤ δ to exist.

We describe how to construct P* and Q*. We call the transformations that we apply to the curves straightenings. As the name implies, it straightens a curve, in the sense that it makes the curve move in only a single direction. Straightenings are formally defined as follows:
**Definition 7.** Given a curve $P : [0, 1] \to \mathbb{R}$ and interval $I = [a, b]$, the I-straightening of $P$ in the positive direction is the curve $P^I : [0, 1] \to \mathbb{R}$, where

$$P^I(x) = \begin{cases} \min\{\max_{x' \leq x} P(x'), b\} & \text{if } P(x) \in I. \\ P(x) & \text{otherwise.} \end{cases}$$

The I-straightening in the negative direction is defined analogously, with the roles of min and max, as well as the roles of $a$ and $b$, switched.

![Figure 6](image)

**Figure 6:** The $\mathbb{R}$-straightening $Q$ (red) of a curve $P$ (blue) in the positive direction. The curve $P$ moves horizontally in one dimension, but the vertical segments are added for clarity. Points on the blue curve are mapped to points on the red curve $Q$ at the same height.

See Figure 6 for an illustration of a straightening. Note that the straightening of a curve in one dimension yields merely a line segment. However, due to its point-wise definition, straightenings can be applied to curves in higher dimensions as well, which is required for our case.

The straightening of a curve can be computed in time linear in the number of vertices of the curve by performing the straightening operation over the vertices and connecting the new vertices with line segments. We transform the curves $CP_i$, $BP_i$, $CQ_i$, and $BQ_i$ using straightenings. We iterate over the indices $i = 1, \ldots, k$. Assume that $CP_{i+1}(0) > CP_i(1)$. The other case is handled symmetrically. Let $\text{Im}(CP_i) = [a, b]$ and set $I = [a - \delta, CP_{i+1}(0) - 4\delta]$. We construct the I-straightenings $CP'_i \circ BP'_i$ and $CQ'_i \circ BQ'_i$ of $CP_i \circ BP_i$ and $CQ_i \circ BQ_i$, respectively, in the positive direction. Also, if $i \leq k - 1$, let $I' = [CP_{i+1}(0) - 4\delta, CP_{i+1}(0)]$. If $CP_{i+1}(1) > CP_{i+1}(0)$, then we construct the I'-straightenings $BP''_i$ and $BQ''_i$ of $BP_i'$ and $BQ_i'$, respectively, in the positive direction. Else we construct the I'-straightenings in the negative direction.

We choose the straightening intervals in such a way that both the compact curves $CP_i$ and bridge curves $BP_i$ are straightened into line segments. (For $CP_i$, this requires only the I-straightening, but for $BP_i$ this requires both the I- and I'-straightenings.) The properties of $CP_i$ and $BP_i$ give all resulting line segments lengths greater than $4\delta$ (except for the last). Hence, the curve $P^* = CP'_i \circ \bigcap_{2 \leq i \leq k}(BP''_{i-1} \circ CP'_i)$ has long edges only, except for the last. This is proved in the following lemma.

**Lemma 18.** All edges of $P^*$, except possibly the last edge, have length greater than $4\delta$.

**Proof.** First we show that $CP'_i = CP_i(0)CP_i(1)$ for all $i$. Let $\text{Im}(CP_i) = [a, b]$, and assume without loss of generality that $CP_{i+1}(0) > CP_i(1)$. Note that this also means that $CP_i(1) \geq CP_i(x)$ for all $x \in [0, 1]$. Let $I = [a - \delta, CP_{i+1}(0) - 4\delta]$. Then $CP'_i$ is the I-straightening of $CP_i$. Since $CP_i \subseteq I$, we have that $CP'_i(x) = \max_{x' \leq x} CP_i(x)$ for all $x \in [0, 1]$, and thus $CP'_i(0) \leq CP'_i(x) \leq CP'_i(1)$. Note that $CP'_i(0) = CP_i(0)$ and $CP'_i(1) = CP_i(1)$. It follows that $CP'_i = CP_i(0)CP_i(1)$.

Similar reasoning as above shows that $BP'_i \cap I$ forms a single line segment, oriented in the same direction as $CP'_i$. Because $BP'_i \setminus I \subseteq I'$ for $I' = [CP_{i+1}(0) - 4\delta, CP_{i+1}(0)]$, similar reasoning as above again shows that either $BP''_i = BP'_i$, or $BP''_i \setminus BP'_i$ is a single line segment. In both cases, $BP''_i$ is a single line segment. Thus, $CP'_i \circ BP''_i$ is a single line segment.
Because $P^* = CP_i \circ \bigcirc_{2 \leq i \leq k}(BP''_{i-1} \circ CP_i)$, it follows that if every $CP_i \circ BP''_i$ has length greater than $4\delta$, except for when $i = k$, then all but the last edge of $P^*$ has length greater than $4\delta$. That $CP_i \circ BP''_i$ has length greater than $4\delta$ follows from the fact that $CP_i = CP_i(0)CP_i(1)$, and because we constructed $CP_i$ such that $|CP_i(0) - CP_i(1)| \geq 4\delta$.

Let $Q^* = CQ_i \circ \bigcirc_{2 \leq i \leq k}(BQ'^*_{i-1} \circ CQ_i)$. In Lemma 21, we show that a $(c, \delta)$-exit set with respect to $P^*$ and $Q^*$, with $c = O(1)$, is also a $(4L/\delta, \delta)$-exit set with respect to $P$ and $Q$. This theorem follows from the following two lemmas, which show that the Fréchet distance between $P$ and subcurves of $Q$ is approximately preserved between $P^*$ and subcurves of $Q^*$.

**Lemma 19.** If $d_F(P, Q[0, y^*]) \leq \delta$ for some $y^* \in [0, 1]$, then $d_F(P^*, Q^*[0, y^*]) \leq \delta$.

**Proof.** We show that the matching $(f, g)$ between $P^*$ and $Q^*[0, y^*]$ induced by (the same) $\delta$-matching $(f, g)$ between $P$ and $Q[0, y^*]$ has cost at most $\delta$. Let a point $p$ on $P$ be matched to a point $q$ on $Q$ by $(f, g)$, and let $p'$ and $q'$ be the respective points after straightening. Without loss of generality, assume that $p \in CP_i \circ BP_i$ for some $i$ (that is, $BP_i$ exists). By Theorem 17, we have that $q \in CQ_i \circ BQ_i \circ CQ_{i+1}$.

Let $I$ and $I'$ be the regions used for straightening $CP_i \circ BP_i$ and $CQ_i \circ BQ_i$. We first prove the statement for when $q \in CQ_i \circ BQ_i$. Note that if the $I$- and $I'$-straightenings happen in opposite directions, then the $I$-straightening happens in the direction of $I'$ and vice versa. So if $p$ and $q$ move in opposite directions during straightening, they move towards each other and we have $|p' - q'| \leq |p - q| \leq \delta$.

We argue that if $p$ and $q$ move in the same direction, then $|p' - q'| \leq \delta$. Without loss of generality, assume that both the $I$- and $I'$-straightenings happen in the positive direction. Since $I \cap I' \neq \emptyset$, performing the two straightenings separately is equivalent to performing the $(I \cup I')$-straightening of the curves. Let $p = (CP_i \circ BP_i)(x)$ and $q = (CQ_i \circ BQ_i)(y)$. We have that $p' = \min\{\max_{x' \leq x}(CP_i \circ BP_i)(x'), CP_{i+1}(0)\}$ and $q' = \min\{\max_{y' \leq y}(CQ_i \circ BQ_i)(y'), CP_{i+1}(0)\}$. If $|p' - q'| > \delta$, then there is either a point on $(CP_i \circ BP_i)[0, x]$ that is not close to any point on $(CQ_i \circ BQ_i)[0, y]$, or vice versa. Our construction of the subcurves gives us that all points on $CP_i$ are close to points on $CQ_i$ and vice versa, and symmetrically for $BP_i$ and $BQ_i$. Thus we have a contradiction, from which it follows that $|p' - q'| \leq \delta$ when $q \in CQ_i \circ BQ_i$.

Now assume that $q \in CQ_{i+1}$. Theorem 17 then gives us that $p \in BP_i$. Note that by the same theorem, there must now be a point $p'' \in BP_i$ before $p$, that is matched to $CQ_{i+1}(0)$. We again assume without loss of generality that $CP_{i+1}(0) > CP_i(1)$. This means that $BP_i$ is $2\delta$-direction-preserving in the positive direction, which implies that $p \geq p'' - 2\delta \geq CP_{i+1}(0) - 3\delta \geq CP_{i+1}(0) - 4\delta$. By construction we also have $p \leq CP_{i+1}(0)$, and thus $p \in I'$.

Let $p = BP_i(x)$ and $q = CQ_{i+1}(y)$. Let $I''$ be the interval $I'$ expanded on both sides by $\delta$. Then $p' = \min\{\max_{x' \leq x} BP_i(x'), CQ_{i+1}(0)\}$ and $q' = \min\{\max_{y' \leq y} CQ_{i+1}(y'), CP_{i+2}(0) - 4\delta\} \geq \min\{\max_{y' \leq y} CQ_{i+1}(y'), CP_{i+1}(0)\}$. Like before, if $|p' - q'| > \delta$, then there is either a point on $BP_i[0, x] \cap I''$ that is not close to any point on $CQ_{i+1}[0, y] \cap CQ_{i+1} \cap I''$, or vice versa. Note that $BP_i[0, x] \cap I'' = BP_i[0, x] \cap I'$, as $BP_i'$ is the $I$-straightening of $BP_i$, where $I \cap I' = \emptyset$. By our construction, we have that all points in $BP_i \cap I'$ are close to points on $CQ_{i+1} \cap I''$ and vice versa. This gives a contradiction, from which it follows that $|p' - q'| \leq \delta$ when $q \in CQ_{i+1}$.

**Lemma 20.** If $d_F(P^*, Q^*[0, y^*]) \leq \delta$ for some $y^* \in [0, 1]$, then $d_F(P, Q[0, y^*]) \leq 4L$.

**Proof.** We prove that $|P(x) - P^*(x)| \leq 4L - 2\delta$, and similarly that $|Q(y) - Q^*(y)| \leq 4L$, for any $x \in [0, 1]$ and $y \in [0, t]$. It then follows that any $\delta$-matching between $P^*$ and $Q^*$ is also an $4L$-matching between $P$ and $Q$. 

19
Let $x \in [0, 1]$, and assume without loss of generality that $P(x) \in CP_i \circ BP_i$ for some $i$. Let $I$ be the interval such that $CP_i'$ and $BP_i'$ are the $I$-straightenings of $CP_i$ and $BP_i$, respectively. Also, let $I'$ be the interval such that $BP_i''$ is the $I$-straightening of $BP_i'$.

Without loss of generality, assume that the $I$-straightenings happen in the positive direction. First assume that $P(x) = CP_i(x')$ for some $x' \in [0, 1]$. Then because $CP_i \subseteq I$, we have that $P^*(x) = CP_i'(x') = \max_{x'' \leq x'} CP_i''(x'')$. By Lemma 15, the diameter of $CP_i$ is at most $4L - 2\delta$. It follows that $|P(x) - P^*(x)| \leq 4L - 2\delta$ if $P(x) \in CP_i$.

Now assume that $P(x) = BP_i(x')$ for some $x' \in (0, 1)$. Note that because the $I$-straightening happens in the positive direction, we have that $BP_i$ is a direction-preserving in the positive direction. Thus, if $P(x) \in I$, we have that $BP_i'(x') = \max_{x'' \leq x} BP_i''(x)$, and therefore that $|P(x) - P^*(x)| \leq 2\delta < 4L - 2\delta$. If $P(x) \in I'$ instead, then because $I'$ has diameter $4\delta$, it follows that $|P(x) - P^*(x)| \leq 4\delta < 4L - 2\delta$. We therefore have that $|P(x) - P^*(x)| \leq 4L - 2\delta$ if $P(x) \in BP_i$.

Now, to see that $|Q(y) - Q^*(y)| \leq 4L$, note that because the diameter of $CP_i$ is at most $4L - 2\delta$, our construction leads to $Q_i$ having a diameter of at most $4L$. Also, recall that we assumed $BQ_i$ to be a direction-preserving, as otherwise $d_F(P, Q) > \delta$ and we terminate the algorithm immediately. Using these two properties, we can apply the above proof for $|P(x) - P^*(x)|$ here as well. 

From Lemma 19 we obtain that any $\delta$-reachable point in $F_{\leq \delta}(P, Q)$ is also a $\delta$-reachable point in $F_{\leq \delta}(P^*, Q^*)$, and hence must be contained in any exit set. Lemma 20 gives us that any $c\delta$-reachable point in $F_{\leq \delta}(P^*, Q^*)$ is also $4cL$-reachable in $\mathcal{F}_{\leq \delta}(P, Q)$. The above directly implies the following theorem:

**Lemma 21.** Let $P, Q : [0, 1] \to \mathbb{R}^d$ be two curves with $m$ and $n$ vertices, respectively. Let $\delta > 0$ and $\alpha \in [1, n]$, and let $c \geq 1$ be a constant. In $O(m + n)$ time, we can construct two curves $P^*$ and $Q^*$, where $P^*$ is quasi-piecewise $(\geq 4\delta)$-monotone, such that a $(c, \delta)$-exit set with respect to $P^*$ and $Q^*$ is an $(O(L/\delta), \delta)$-exit set with respect to $P$ and $Q$.

Together with the algorithm of Section 3 for computing exit sets in linear time if one of the curve is quasi-piecewise $(\geq 4\delta)$-monotone, the above lemma immediately implies an algorithm for computing an $(O(L/\delta), \delta)$-exit set with respect to $P$ and $Q$ in $O(m + n)$ time. This algorithm is used in the black box procedure of Theorem 4 in Section 2 to obtain an approximate decision algorithm for the Fréchet distance.

**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. [10.1137/130920526](https://doi.org/10.1137/130920526)

[2] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995. [10.1142/S0218195995000064](https://doi.org/10.1142/S0218195995000064)

[3] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In Yossi Azar and Thomas Erlebach, editors, *14th Annual European Symposium on Algorithms*, volume 4168 of *Lecture Notes in Computer Science*, pages 52–63. Springer, 2006. [10.1007/11841036\_8](https://doi.org/10.1007/11841036\_8)
[4] Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly sub-quadratic algorithms unless SETH fails. In 55th IEEE Annual Symposium on Foundations of Computer Science, pages 661–670. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.76.

[5] Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds. International Journal of Computational Geometry & Applications, 27(1-2):85–120, 2017. doi:10.1142/S0218195917600056.

[6] 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.

[7] Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four soviets walk the dog: Improved bounds for computing the Fréchet distance. Discrete & Computational Geometry, 58(1):180–216, 2017. doi:10.1007/s00454-017-9878-7.

[8] 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, pages 2887–2901. SIAM, 2019. doi:10.1137/1.9781611975482.179.

[9] Timothy M. Chan and Zahed Rahmati. An improved approximation algorithm for the discrete Fréchet distance. Information Processing Letters, 138:72–74, 2018. doi:10.1016/j.ipl.2018.06.011.

[10] 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, pages 26:1–26:14, 2021. doi:10.4230/LIPIcs.SoCG.2021.26.

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

[12] Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. Discrete & Computational Geometry, 48(1):94–127, 2012. doi:10.1007/s00454-012-9402-z.

[13] Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 766–785, 2016. doi:10.1137/1.9781611974331.ch55.

[14] Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. 1994.

[15] Michael Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In Christian Choffrut and Matthias Jantzen, editors, 8th Annual Symposium on Theoretical Aspects of Computer Science, volume 480 of Lecture Notes in Computer Science, pages 127–136, 1991. doi:10.1007/BFb0020793.

[16] Joachim Gudmundsson, Majid Mirzanezhad, Ali Mohades, and Carola Wenk. Fast Fréchet distance between curves with long edges. International Journal of Computational Geometry & Applications, 29(2):161–187, 2019. doi:10.1142/S0218195919500043.
[17] Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30(4):852–865, 1983. doi:10.1145/2157.322410