

The Geodesic Fréchet Distance Between Two Curves Bounding a Simple Polygon

Thijs van der Horst  

Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Marc van Kreveld  

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Tim Ophelders  

Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Bettina Speckmann  

Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Abstract

The Fréchet distance is a popular similarity measure that is well-understood for polygonal curves in \mathbb{R}^d : near-quadratic time algorithms exist, and conditional lower bounds suggest that these results cannot be improved significantly, even in one dimension and when approximating with a factor less than three. We consider the special case where the curves bound a simple polygon and distances are measured via geodesics inside this simple polygon. Here the conditional lower bounds do not apply; Efrat et al. (2002) were able to give a near-linear time 2-approximation algorithm.

In this paper, we significantly improve upon their result: we present a $(1 + \varepsilon)$ -approximation algorithm, for any $\varepsilon > 0$, that runs in $\mathcal{O}(\frac{1}{\varepsilon}(n + m \log n) \log nm \log \frac{1}{\varepsilon})$ time for a simple polygon bounded by two curves with n and m vertices, respectively. To do so, we show how to compute the reachability of specific groups of points in the free space at once, by interpreting the free space as one between separated one-dimensional curves. We solve this one-dimensional problem in near-linear time, generalizing a result by Bringmann and Künnemann (2015). Finally, we give a linear time exact algorithm if the two curves bound a convex polygon.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Fréchet distance, approximation, geodesic, simple polygon

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.35

Related Version *Full Version*: <https://arxiv.org/abs/2501.03834>

Funding *Tim Ophelders*: partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.

1 Introduction

The Fréchet distance is a well-studied similarity measure for curves in a metric space. Most results so far concern the Fréchet distance between two polygonal curves R and B in \mathbb{R}^d with n and m vertices, respectively. Then the Fréchet distance between two such curves can be computed in $\tilde{\mathcal{O}}(nm)$ time (see e.g. [1, 5]). There is a closely matching conditional lower bound: If the Fréchet distance between polygonal curves can be computed in $\mathcal{O}((nm)^{1-\varepsilon})$ time (for any constant $\varepsilon > 0$), then the Strong Exponential Time Hypothesis fails [3]. This lower bound extends to curves in one dimension, and holds even when approximating to a factor less than three [6].

Because it is unlikely that exact strongly subquadratic algorithms exist, approximation algorithms have been developed [9, 15, 8]. The authors [15] were the first to present an algorithm which results in an arbitrarily small polynomial approximation factor (n^ε for any



© Thijs van der Horst, Marc van Kreveld, Tim Ophelders, and Bettina Speckmann;
licensed under Creative Commons License CC-BY 4.0

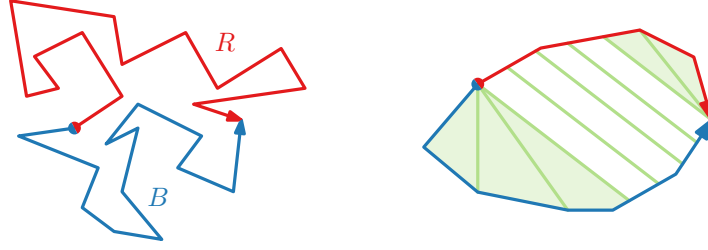
33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 35; pp. 35:1–35:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** (left) Curves R and B bound a simple polygon. (right) A maximally-parallel matching.

$\varepsilon \in (0, 1]$ in strongly subquadratic time ($\tilde{O}(n^{2-\varepsilon})$). Very recently, Cheng et al. [8] gave the first (randomized) constant-factor approximation algorithm with a strongly subquadratic running time. Specifically, it computes a $(7 + \varepsilon)$ -approximation in $\mathcal{O}(nm^{0.99} \log(n/\varepsilon))$ time.

For certain families of “realistic” curves, the SETH lower bound does not apply. For example, when the curves are *c-packed*, Driemel et al. [11] gave an $(1 + \varepsilon)$ -approximation algorithm, for any $\varepsilon \in (0, 1)$, that runs in $\tilde{O}(cn/\varepsilon)$ time. Bringmann and Künnemann [4] improved the running time to $\tilde{O}(cn/\sqrt{\varepsilon})$ time.

For curves in one dimension with an imbalanced number of vertices, the Fréchet distance can be computed in strongly subquadratic time without making extra assumptions about the shape of the curves. This was recently established by Blank and Driemel [2], who give an $\tilde{O}(n^{2\alpha} + n)$ -time algorithm when $m = n^\alpha$ for some $\alpha \in (0, 1)$.

If the two polygonal curves R and B lie inside a simple polygon P with k vertices and we measure distances by the geodesic distance inside P , then neither the upper nor the conditional lower bound change in a fundamental way. Specifically, Cook and Wenk [10] show how to compute the Fréchet distance in this setting in $\mathcal{O}(k + N^2 \log kN \log N)$ time, with $N = \max\{n, m\}$. For more general polygonal obstacles, Chambers et al. [7] give an algorithm that computes the *homotopic* Fréchet distance in $\mathcal{O}(N^9 \log N)$ time, where $N = m + n + k$ is the total number of vertices on the curves and obstacles.

Har-Peled et al. [13] investigate the setting where R and B are simple, interior-disjoint curves on the boundary of a triangulated topological disk. If the disk has k faces, their algorithm computes a $\mathcal{O}(\log k)$ -approximation to the homotopic Fréchet distance in $\mathcal{O}(k^6 \log k)$ time. Efrat et al. [12] consider a more geometric setting, where R and B bound a simple polygon (see Figure 1 (left)). Here, the SETH lower bound does not apply; a 2-approximation to the geodesic Fréchet distance can be computed in $\mathcal{O}((n + m) \log nm)$ time [12]. Moreover, the authors [14] recently gave an $\mathcal{O}((n + m) \log^4 nm)$ -time exact algorithm for a similar setting, where distances are measured under the L_1 -geodesic distance. Their result implies a $\sqrt{2}$ -approximation algorithm for the geodesic Fréchet distance.

Organization and results. In this paper, we significantly improve upon the results of Efrat et al. [12] and the authors [14]: we present a $(1 + \varepsilon)$ -approximation algorithm for the geodesic Fréchet distance, for any $\varepsilon \in (0, 1]$, that runs in $\mathcal{O}(\frac{1}{\varepsilon}(n + m \log n) \log nm \log \frac{1}{\varepsilon})$ time when R and B together bound a simple polygon. We give an overview of our algorithm in Section 2. Our algorithm relies on an interesting connection between matchings and nearest neighbors and is described in Section 3. There we also explain how to transform the decision problem for *far* points on B (those who are not a nearest neighbor of any point on R) into a problem between separated one-dimensional curves. This problem is about computing, given a set of “entrances” and “potential exits” (points anywhere in the parameter space), which potential exits are δ -reachable from entrances. Bringmann and Künnemann [4] study a

restricted variant of this problem in the context of Fréchet distance between c -packed curves. In addition to giving a near-linear time algorithm for the general problem, we also compute the Fréchet distance between two separated one-dimensional curves in linear time.

In the full version of this paper, we additionally describe a simple linear-time algorithm for when P is a convex polygon. We show that in this setting there is a Fréchet matching with a specific structure: a *maximally-parallel matching* (see Figure 1 (right)). We compute the orientation of the parallel part from up to $O(n + m)$ different tangents, which we find using “rotating calipers.”

Preliminaries. A (polygonal) *curve* R with n vertices is a continuous map $R: [1, n] \rightarrow \mathbb{R}^d$ that is linear on each interval $[i, i + 1]$ where i is an integer. Its vertices are the points $R(i)$. If the vertices lie in the plane, then we say R is two-dimensional.¹ A one-dimensional curve is defined analogously.

We denote by $R[x, x']$ the subcurve of R over the domain $[x, x']$, and abuse notation slightly to let $R[r, r']$ to also denote this subcurve when $r = R(x)$ and $r' = R(x')$. The *edges* of R are the directed line segments $R[i, i + 1]$ for integers $i \in [1, n - 1]$. We write $|R|$ to denote the number of vertices of R . In this work, we consider two simple, interior-disjoint curves $R: [1, n] \rightarrow \mathbb{R}^2$ and $B: [1, m] \rightarrow \mathbb{R}^2$ with $R(1) = B(1)$ and $R(n) = B(m)$, and say that they bound a simple polygon P .

A *reparameterization* of $[1, n]$ is a non-decreasing surjection $f: [0, 1] \rightarrow [1, n]$. Two reparameterizations f and g of $[1, n]$ and $[1, m]$, describe a *matching* (f, g) between two curves R and B with n and m vertices, respectively, where any point $R(f(t))$ is matched to $B(g(t))$. The matching (f, g) is said to have *cost*

$$\max_t d(R(f(t)), B(g(t))),$$

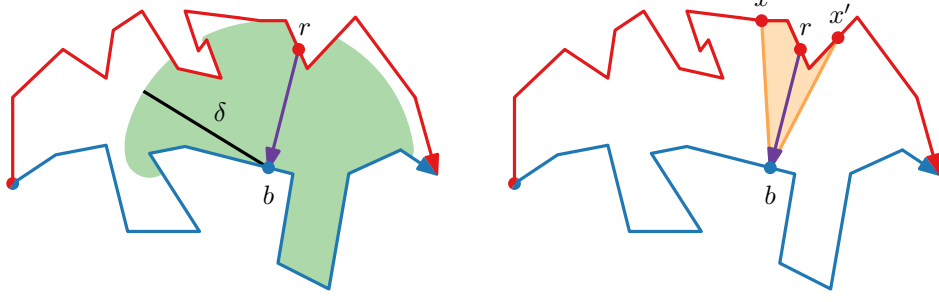
where $d(\cdot, \cdot)$ is the geodesic distance between points in P . A matching with cost at most δ is called a δ -*matching*. The (continuous) *geodesic Fréchet distance* $d_F(R, B)$ between R and B is the minimum cost over all matchings. The corresponding matching is a *Fréchet matching*.

The *parameter space* of R and B is the axis-aligned rectangle $[1, n] \times [1, m]$. Any point (x, y) in the parameter space corresponds to the pair of points $R(x)$ and $B(y)$ on the two curves. A point (x, y) in the parameter space is δ -*close* for some $\delta \geq 0$ if $d(R(x), B(y)) \leq \delta$. The δ -*free space* $\mathcal{F}_\delta(R, B)$ of R and B is the set of δ -close points. Alt and Godau [1] observed that there is a one-to-one correspondence between δ -matchings between subcurves $R[x, x']$ and $B[y, y']$, and bimonotone paths from (x, y) to (x', y') through $\mathcal{F}_\delta(R, B)$. We abuse terminology slightly and refer to such paths as δ -matchings as well.

A point $q = (x', y') \in \mathcal{F}_\delta(R, B)$ is δ -*reachable* from a point $p = (x, y)$ if $x \leq x'$ and $y \leq y'$, and there exists a bimonotone (i.e., monotone in both coordinates) path in $\mathcal{F}_\delta(R, B)$ from p to q . That is, if the subcurves $R[x, x']$ and $B[y, y']$ have Fréchet distance at most δ between them. Points that are δ -reachable from $(1, 1)$ are simply called δ -reachable points.

Let $\varepsilon \in (0, 1]$ be a parameter. A $(1 + \varepsilon)$ -*approximate decision algorithm* for our problem takes a decision parameter $\delta \geq 0$, and outputs either that $d_F(R, B) \leq (1 + \varepsilon)\delta$ or that $d_F(R, B) > \delta$. It may report either answer if $\delta < d_F(R, B) \leq (1 + \varepsilon)\delta$.

¹ Curves are inherently one-dimensional objects. We abuse terminology slightly to refer to the ambient dimension as the dimension of a curve.



■ **Figure 2** (left) Points r and b with $b \in NN(r)$. The region shaded in green consists of all points within geodesic distance δ of b . (right) The (r, b, δ) -nearest neighbor fan (orange).

2 Algorithmic outline

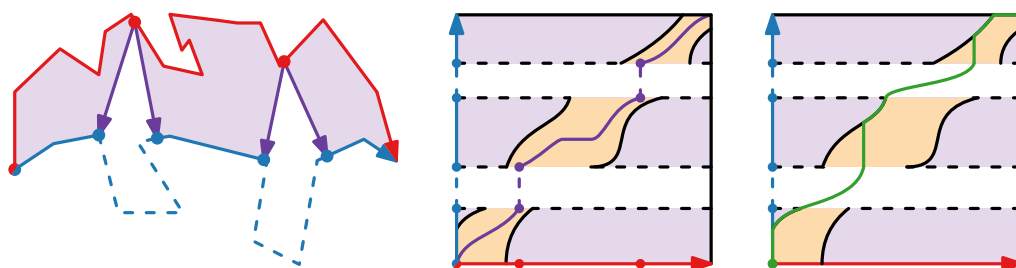
In this section we sketch the major parts of our algorithm that approximates the geodesic Fréchet distance $\delta_F := d_F(R, B)$ between curves R and B . We approximate δ_F using a $(1 + \varepsilon)$ -approximate decision algorithm, and use binary search to find the correct decision parameter. In Section 3.4 we relate the Fréchet distance to the geodesic Hausdorff distance δ_H , which allows us to bound the number of iterations of the binary search. In particular, we show that δ_F lies in the range $[\delta_H, 3\delta_H]$. The Hausdorff distance δ_H can be computed in $\mathcal{O}((n + m) \log nm)$ time [10, Theorem 7.1]. A binary search over $[\delta_H, 3\delta_H]$ results in a $(1 + \varepsilon)$ -approximation of the Fréchet distance after $\mathcal{O}(\log \frac{1}{\varepsilon})$ calls to the decision algorithm. Theorem 1 follows.

► **Theorem 1.** *For any $\varepsilon \in (0, 1]$, we can compute a $(1 + \varepsilon)$ -approximation to $d_F(R, B)$ in $\mathcal{O}(\frac{1}{\varepsilon}(n + m \log n) \log nm \log \frac{1}{\varepsilon})$ time.*

The approximate decision algorithm. In the remainder of this section we outline the approximate decision algorithm, which is presented in detail in Section 3. At its heart lies a useful connection between matchings and nearest neighbors. A *nearest neighbor* of a point r on R is any point b on B that among all points on B is closest to r . We denote the nearest neighbor(s) of r by $NN(r)$. We prove in Section 3.1 that any δ -matching matches each nearest neighbor b of r relatively close to r . Specifically, b must be matched to only points r' for which the entire subcurve of R between r and r' is within distance δ of b . We introduce the concept of a (r, b, δ) -nearest neighbor fan to capture the candidate locations for r' .

The (r, b, δ) -nearest neighbor fan $F_{r,b}(\delta)$ contains the point b and the maximal subcurve $R[x, x']$ that contains r and is within geodesic distance δ of b ; it is the union of geodesics between b and points on $R[x, x']$, see Figure 2. We call b the *apex* of $F_{r,b}(\delta)$ and $R[x, x']$ the *leaf* of the fan. We prove in Section 3.1 that any δ -matching must match the apex b to a point in the leaf $R[x, x']$.

As r moves forward along R , so do its nearest neighbors b along B . Their nearest neighbor fans $F_{r,b}(\delta)$ move monotonically through the polygon P bounded by R and B . However, while r moves continuously along R , the points b and their nearest neighbor fans might jump discontinuously. Such discontinuities occur due to points b that are not a nearest neighbor of any point on R , and thus at points that are not the apex of a nearest neighbor fan. We classify the points on B accordingly: we call a point b on B a *near* point if it is a nearest neighbor of at least one point on R , and call b a *far* point otherwise.



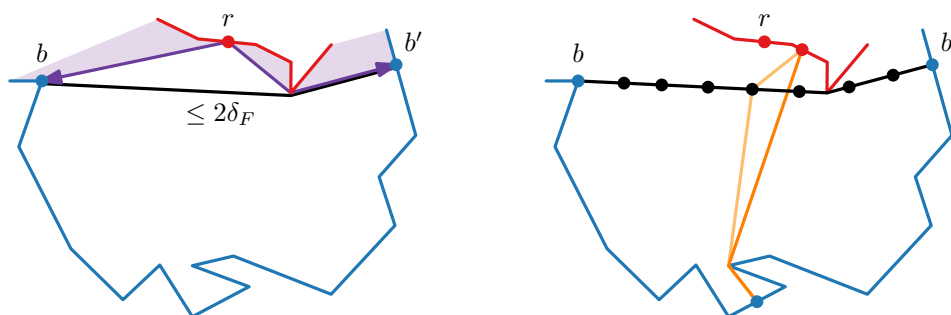
■ **Figure 3** (left) The mappings (purple) from points on R to their nearest neighbor(s) on B . (middle) The partition of the parameter space based on near and far points on B . The partly-dashed purple curve indicates the nearest neighbor(s) on B of points on R . The beige regions correspond to the (r, b, δ) -nearest neighbor fans. (right) A δ -matching that is greedy on B inside the regions.

The distinction between near and far points induces a partition \mathcal{H} of the parameter space into horizontal slabs. We consider these slabs from bottom to top, and iteratively construct a δ -matching (provided that one exists). Recall that a δ -matching is a bimonotone path in $\mathcal{F}_\delta(R, B)$ from the bottom-left corner of the parameter space to the top-right corner.

Inside a slab $H_{\text{near}} \in \mathcal{H}$ corresponding to a subcurve of B with only near points, the nearest neighbor fans correspond to a connected, x -monotone and y -monotone region \mathcal{R} spanning the entire height of H_{near} . These regions are illustrated in Figure 3. The intersection between any δ -matching and H_{near} is contained in \mathcal{R} . The structure of \mathcal{R} implies that if a δ -matching between R and B exists, there exists one which moves vertically up inside \mathcal{R} whenever this is possible. Geometrically, this corresponds to greedily traversing the near points on B , and traversing parts of R only when necessary. We formalize this in Section 3.2.

Slabs whose corresponding subcurves of B have only far points pose the greatest technical challenge for our algorithm; we show how to match far points in an approximate manner in Section 3.3. Specifically, let $H_{\text{far}} \in \mathcal{H}$ correspond to a subcurve $B[b, b']$ with only far points on its interior. We compute a suitable subcurve of R that can be $(1 + \varepsilon)\delta$ -matched to $B[b, b']$ in the following manner. First we argue that $d(b, b') \leq 2\delta_F$. In other words, the geodesic from b to b' is short and separates R from the subcurve $B[b, b']$. We use this separating geodesic to transform the problem of creating a matching for far points into $K = \mathcal{O}(1/\varepsilon)$ one-dimensional problems.

Specifically, we discretize the separator with K points, which we call *anchors*, and ensure that consecutive anchors are at most $\varepsilon\delta$ apart. We snap our geodesics to these anchors (see Figure 4), which incurs a small approximation error. Based on which anchor a geodesic



■ **Figure 4** (left) The points b and b' are both nearest neighbor of some point r , implying a short separator. (right) Adding anchor points to the separator and snapping the orange geodesic (between arbitrary points on R and $B[b, b']$) to one.

snaps to, we partition the parameter space of R and $B[b, b']$ into regions, one for each anchor. For each anchor point, the lengths of the geodesics snapped to it can be expressed as distances between points on two separated one-dimensional curves; this results in a one-dimensional problem which we can solve exactly. In Section 3.3 we present the transformation into one-dimensional curves, and in Section 4 we present an efficient exact algorithm for the resulting one-dimensional problem.

3 Approximate geodesic Fréchet distance

3.1 Nearest neighbor fans and partitioning the parameter space

We first present useful properties of nearest neighbor fans and their relation to matchings. In Lemma 3 we give a crucial property of nearest neighbor fans, namely that *any* δ -matching between R and B matches b to only points in the leaf of the fan. For the proof, we make use of the following auxiliary lemma:

► **Lemma 2.** *Let $r \in R$ and $b \in NN(r)$. For any points $r' \in R$ and $b' \in B$ on opposite sides of the geodesic $\pi(r, b)$ between r and b , we have $d(r', b) \leq d(r', b')$.*

► **Lemma 3.** *Let $r \in R$ and $b \in NN(r)$. For any $\delta \geq 0$, every δ -matching between R and B matches b to only points in the leaf of $F_{r,b}(\delta)$.*

Proof. Fix a δ -matching (f, g) and let r' be a point matched to b by this matching. Assume without loss of generality that r' comes before r along R . Let \hat{r} be a point between r' and r . The δ -matching (f, g) matches \hat{r} to at least one point \hat{b} after b . Thus we obtain from Lemma 2 that $d(\hat{r}, b) \leq d(\hat{r}, \hat{b}) \leq \delta$. This proves that all points between r' and r are included in the leaf of $F_{r,b}(\delta)$. ◀

We partition the parameter space into (closed) maximal horizontal slabs, such that for each slab $[1, n] \times [y, y']$, either the subcurve $B[y, y']$ contains only near points, or its interior contains only far points. Let \mathcal{H} be the resulting partition. Each slab $H \in \mathcal{H}$ has two horizontal line segments, one on its bottom and one on its top side, that correspond to nearest neighbor fans. We refer to these line segments as the *entrance* and *exit intervals* of H , and refer to points on them as the *entrances* and *exits* of H . A consequence of Lemma 3 is that any δ -matching enters and leaves H through an entrance and an exit, respectively. We compute δ -safe entrances and exits: δ -reachable points from which (n, m) is δ -reachable. Each such point is used by a δ -matching, and is used to iteratively determine if such a matching exists.

It proves sufficient to consider only a discrete set of entrances and exits for each slab. For each slab, we identify (implicitly) a set of $\mathcal{O}(n)$ entrances and exits that contain δ -safe entrances and exits (if any exist at all). We define these entrances and exits using locally closest points and will call them transit points.

A point r on R is *locally closest* to a point b on B if perturbing r infinitesimally while staying on R increases its distance to b . The *transit* entrances and exits are those entrances and exits (x, y) where $R(x)$ is either a vertex or locally closest to $B(y)$. We show that it is sufficient to consider only transit entrances and exits:

► **Lemma 4.** *If there exists a δ -matching, then there exists one that enters and leaves each slab through a transit entrance and exit.*

Our algorithm computes a δ -safe transit exit for each slab. To do so, it requires the explicit entrance and exit intervals. We compute these using a geodesic Voronoi diagram.

► **Lemma 5.** *The partition \mathcal{H} consists of $\mathcal{O}(m)$ slabs. We can compute \mathcal{H} , together with the entrance and exit interval of each slab, in $\mathcal{O}((n+m)\log nm)$ time in total.*

3.2 Slabs of near points

Let $H_{\text{near}} \in \mathcal{H}$ be a slab corresponding to a subcurve \hat{B} of B with only near points. We use properties of nearest neighbor fans to determine a δ -safe transit exit of H_{near} . A crucial property is that the nearest neighbor fans behave monotonically with respect to their apexes, if δ is large enough. Specifically, this is the case if $\delta \geq \delta_H$, the geodesic Hausdorff distance between R and B . This is the maximum distance between a point on $R \cup B$ and its closest point on the other curve.

► **Lemma 6.** *Suppose $\delta \geq \delta_H$. Let b and b' be near points on B and let $R[x_1, x'_1]$ and $R[x_2, x'_2]$ be the leaves of their respective nearest neighbor fans. If b comes before b' , then $x_1 \leq x_2$ and $x'_1 \leq x'_2$.*

The monotonicity of the nearest neighbor fans, together with the fact that each point on \hat{B} corresponds to such a fan, ensures that we can determine such an exit in logarithmic time (see Lemma 8). We make use of the following data structure that reports transit exits:

► **Lemma 7.** *Given the exit interval $[x, x'] \times \{y\}$ of H_{near} , we can report the at most three transit exits on a horizontal line segment $[i, i+1] \times \{y\}$, for any integer $1 \leq i < n$, in $\mathcal{O}(\log nm)$ time, after $\mathcal{O}(n+m)$ preprocessing time.*

► **Lemma 8.** *Suppose $\delta \geq \delta_H$. Let $H_{\text{near}} \in \mathcal{H}$ be a slab corresponding to a subcurve of B with only near points. Given the exit interval of H_{near} , together with a δ -safe transit entrance, we can compute a δ -safe transit exit in $\mathcal{O}(\log nm)$ time, after $\mathcal{O}(n+m)$ preprocessing time.*

Proof (sketch). Given a δ -safe transit entrance p_{enter} , the leftmost transit exit to the right of p_{enter} is δ -safe. We report it with the data structure of Lemma 7. ◀

3.3 Slabs of far points

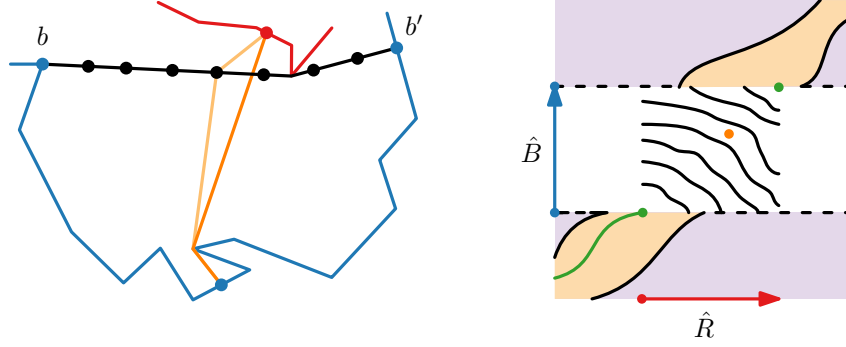
Next we give an algorithm for computing a δ -safe transit exit of a given slab $H_{\text{far}} \in \mathcal{H}$ that corresponds to a subcurve of B with only far points on its interior. Our algorithm is approximate: given $\varepsilon \in (0, 1]$, it computes an (ε, δ) -safe transit exit (if one exists). This is a $(1 + \varepsilon)\delta$ -reachable point from which (n, m) is δ -reachable. Such a transit exit behaves like a δ -safe transit exit for the purpose of iteratively constructing a matching.

To compute an (ε, δ) -safe transit exit, we make use of an approximate decision algorithm that uses the fact that \hat{B} has only far points on its interior. We present this algorithm in Section 3.3.1. In Section 3.3.2 we then apply this approximate decision algorithm in a search procedure, where we search over the $\mathcal{O}(n)$ transit exits to compute an (ε, δ) -safe one.

3.3.1 Approximate decision algorithm for far points

Let \hat{B} be the subcurve of B that corresponds to H_{far} , so its interior has only far points. Let \hat{R} be an arbitrary subcurve of R , for which we seek to approximately determine whether $d_F(\hat{R}, \hat{B}) \leq \delta$. We report either that $d_F(\hat{R}, \hat{B}) \leq (1 + \varepsilon)\delta$, or that $d_F(\hat{R}, \hat{B}) > \delta$.

For our algorithm, we first discretize the space of geodesics between points on \hat{R} and points on \hat{B} , by grouping the geodesics into few ($\mathcal{O}(1/\varepsilon)$) groups and rerouting the geodesics in a group through some representative point. Let \hat{b} and \hat{b}' be the first, respectively last,



■ **Figure 5** (left) Snapping a geodesic (orange) to an anchor. (right) The eight regions in the parameter space of \hat{R} and \hat{B} corresponding to the first eight (out of nine) anchors. The orange geodesic lies in region \mathcal{R}_5 .

endpoints of \hat{B} . There is a point r on R with $NN(r) = \{\hat{b}, \hat{b}'\}$. We observe that this implies the geodesic $\pi(\hat{b}, \hat{b}')$ that connects \hat{b} to \hat{b}' is short with respect to the Fréchet distance, and thus with respect to any relevant value for δ :

► **Lemma 9.** $d(\hat{b}, \hat{b}') \leq 2d_F(\hat{R}, \hat{B})$.

We assume for the remainder that $d(\hat{b}, \hat{b}') \leq 2\delta$; if this is not the case, we immediately report that $d_F(\hat{R}, \hat{B}) > \delta$. This assumption means that the geodesic $\pi(\hat{b}, \hat{b}')$ is a short separator between \hat{R} and \hat{B} . That is, any geodesic between a point on \hat{R} and a point on \hat{B} intersects $\pi(\hat{b}, \hat{b}')$. For clarity, we denote by π_{sep} the separator $\pi(\hat{b}, \hat{b}')$. We use the short separator to formulate the (exact) reachability problem as $\mathcal{O}(1/\varepsilon)$ subproblems involving one-dimensional curves. This is where we incur a small approximation error.

We discretize π_{sep} with $K + 1 = \mathcal{O}(1/\varepsilon)$ points $\hat{b} = a_1, \dots, a_{K+1} = \hat{b}'$, which we call *anchors*, and ensure that consecutive anchors at most a distance $\varepsilon\delta$ apart, see Figure 5 (left). We assume that no anchor coincides with a vertex of $\hat{R} \cup \hat{B}$ (except for a_1 and a_{K+1}).

We route geodesics between points on \hat{R} and points on \hat{B} through these anchors. Specifically, for points $\hat{r} \in \hat{R}$ and $\hat{b} \in \hat{B}$, if $\pi(\hat{r}, \hat{b})$ intersects π_{sep} between consecutive anchors a_k and a_{k+1} , then we “snap” $\pi(\hat{r}, \hat{b})$ to a_k ; that is, we replace it by the union of $\pi(\hat{r}, a_k)$ and $\pi(a_k, \hat{b})$ (see Figure 5 (right)). This creates a new path between \hat{r} and \hat{b} that goes through a_k and has length at most $d(\hat{r}, \hat{b}) + \varepsilon\delta$.

We associate an anchor a_k with the points (x, y) in the parameter space for which the geodesic $\pi(\hat{R}(x), \hat{B}(y))$ is snapped to a_k . These points form a region \mathcal{R}_k that is connected and monotone in both coordinates (see Figure 5 (right)). We iteratively compute, for each region \mathcal{R}_k , a set of $(1 + \varepsilon)\delta$ -reachable points on its boundary, such that the decision question can be answered by checking if the last set contains $(|\hat{R}|, |\hat{B}|)$. We later formulate these subproblems in terms of pairs of one-dimensional curves that are separated by a point.

We first discretize the problem. For this, we identify sets of points on the shared boundaries between the pairs of adjacent regions, such that there exists a δ -matching that enters and exits each region through such a set. Specifically, for $k = 2, \dots, K$, we let the set S_k contain those points $(x, y) \in \mathcal{R}_{k-1} \cap \mathcal{R}_k$ for which one of $R(x)$ and $B(y)$ is a vertex or locally closest to a_k (so perturbing the point infinitesimally along its curve increases its distance to a_k). We set $S_1 = \{(1, 1)\}$. In the following lemmas, we show that these sets suit our needs and are efficiently constructable.

► **Lemma 10.** *If there exists a δ -matching between \hat{R} and \hat{B} , then there exists one that goes through a point in S_k for every anchor a_k .*

► **Lemma 11.** *Each set S_k contains $\mathcal{O}(|\hat{R}| + |\hat{B}|)$ points and can be constructed in $\mathcal{O}((|\hat{R}| + |\hat{B}|) \log nm)$ time, after $\mathcal{O}(n + m)$ preprocessing time.*

Having constructed the sets S_k for all anchors in $\mathcal{O}(\frac{1}{\varepsilon}(|\hat{R}| + |\hat{B}|) \log nm)$ time altogether, we move to computing subsets $S_k^* \subseteq S_k$ containing all δ -reachable points, and only points that are $(1 + \varepsilon)\delta$ -reachable. We proceed iteratively, constructing S_{k+1}^* from S_k^* . For this, observe that for any point (x, y) in the interior of \mathcal{R}_k , the geodesic $\pi(\hat{R}(x), \hat{B}(y))$ was snapped to a_k . We use this fact to construct a pair of one-dimensional curves that approximately describe the lengths of these geodesics.

Let $\bar{R}_k: [1, |\hat{R}|] \rightarrow \mathbb{R}$ and $\bar{B}_k: [1, |\hat{B}|] \rightarrow \mathbb{R}$ be one-dimensional curves, where we set $\bar{R}_k(x) = -d(\hat{R}(x), a_k)$ and $\bar{B}_k(y) = d(\hat{B}(y), a_k)$ (note the difference in sign). These curves encode the distances between points on \hat{R} and \hat{B} when snapping geodesics to a_k . That is, $|\bar{R}_k(x) - \bar{B}_k(y)|$ is the length of $\pi(\hat{R}(x), \hat{B}(y))$ after snapping to a_k . Hence, for any pair of points $p \in S_k^*$ and $q \in S_{k+1}$, we have the following relations:

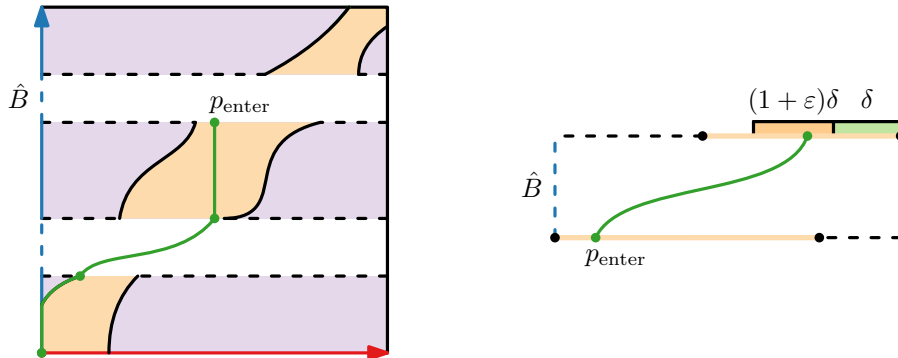
- If q is δ -reachable from p in the parameter space of \hat{R} and \hat{B} , then it is $(1 + \varepsilon)\delta$ -reachable in the parameter space of \bar{R}_k and \bar{B}_k .
- Conversely, if q is $(1 + \varepsilon)\delta$ -reachable from p in the parameter space of \bar{R}_k and \bar{B}_k , then it is $(1 + \varepsilon)\delta$ -reachable in the parameter space of \hat{R} and \hat{B} .

We define S_{k+1}^* as the points in S_{k+1} that are $(1 + \varepsilon)\delta$ -reachable from points in S_k^* , in the parameter space of \bar{R}_k and \bar{B}_k . Computing these points is the problem involving one-dimensional curves that we alluded to earlier.

► **Lemma 12.** *Given S_k^* , we can compute S_{k+1}^* in $\mathcal{O}((|\hat{R}| + |\hat{B}|) \log nm)$ time, after $\mathcal{O}(n + m)$ preprocessing time.*

We apply the above procedure iteratively, computing S_k^* for each anchor a_k . These sets take a total of $\mathcal{O}(\frac{1}{\varepsilon}(|\hat{B}| + |\hat{R}|) \log nm)$ time to construct. Afterwards, if $(|\hat{R}|, |\hat{B}|) \in S_K^*$, we report that $d_F(\hat{R}, \hat{B}) \leq (1 + \varepsilon)\delta$. Otherwise, we report that $d_F(\hat{R}, \hat{B}) > \delta$. We obtain:

► **Lemma 13.** *Let \hat{R} be an arbitrary subcurve of R , and let \hat{B} be a maximal subcurve of B with only far points on its interior. We can decide whether $d_F(\hat{R}, \hat{B}) \leq (1 + \varepsilon)\delta$ or $d_F(\hat{R}, \hat{B}) > \delta$ in $\mathcal{O}(\frac{1}{\varepsilon}(|\hat{R}| + |\hat{B}|) \log nm)$ time, after $\mathcal{O}(n + m)$ preprocessing time.*



■ **Figure 6** The subproblem of matching far points. The exit interval on the right is divided into three regions, based on reachability of points. The aim is to compute a $(1 + \varepsilon)\delta$ -reachable transit exit to the left of all δ -reachable transit exits.

3.3.2 Computing a good exit

Recall that we set out to compute an (ε, δ) -safe transit exit of H_{far} . We assume we are given an (ε, δ) -safe entrance $p_{\text{enter}} = (x, y)$. Since the entire exit interval of H_{far} lies in $\mathcal{F}_\delta(R, B)$, it suffices to compute a transit exit q_{exit} that is $(1 + \varepsilon)\delta$ -reachable from p_{enter} and that lies to the left of all transit exits that are δ -reachable from p_{enter} , see Figure 6. We compute such a transit exit q_{exit} through a search procedure, combined with the decision algorithm.

There are $\mathcal{O}(n)$ transit exits of H_{far} . To avoid running the decision algorithm for each of these, we use exponential search. The choice for exponential search over, e.g., binary search comes from the fact that the running time of the decision algorithm depends on the location of the transit exit, with transit exits lying further to the right in the exit interval of H_{far} needing more time for the decision algorithm. Exponential search ensures that we do not consider transit exits that are much more to the right than needed.

► **Lemma 14.** *Let $H_{\text{far}} \in \mathcal{H}$ be a slab corresponding to a subcurve \hat{B} of B with only far points on its interior. Given an (ε, δ) -safe transit entrance $p = (x, y)$ of H_{far} , we can compute an (ε, δ) -safe transit exit $q = (x', y')$ in $\mathcal{O}(\frac{1}{\varepsilon}(|R[x, x']| + |\hat{B}|) \log n \log nm)$ time.*

3.4 The approximate optimization algorithm

We combine the algorithms of Sections 3.2 and 3.3 to obtain a $(1 + \varepsilon)$ -approximate decision algorithm, which we then turn into an algorithm that computes a $(1 + \varepsilon)$ -approximation of the geodesic Fréchet distance between R and B . Given $\delta \geq 0$ and $\varepsilon \in (0, 1]$, the approximate decision algorithm reports that $d_F(R, B) \leq (1 + \varepsilon)\delta$ or $d_F(R, B) > \delta$.

Let δ_H be the geodesic Hausdorff distance between R and B . This distance, which is the maximum distance between a point on $R \cup B$ to its closest point on the other curve, is a natural lower bound on the geodesic Fréchet distance. If $\delta < \delta_H$, we therefore immediately return that $d_F(R, B) > \delta$. We can compute δ_H in $\mathcal{O}((n + m) \log nm)$ time [10].

Next suppose $\delta \geq \delta_H$. For our approximate decision algorithm, we first compute the partition \mathcal{H} and the entrance and exit intervals of each of its slabs. By Lemma 5, this takes $\mathcal{O}((n + m) \log nm)$ time. We iterate over the $\mathcal{O}(m)$ slabs of \mathcal{H} from bottom to top. Once we consider a slab $H \in \mathcal{H}$, we have computed an (ε, δ) -safe transit entrance $p_{\text{enter}} = (x, y)$ (except if H is the bottom slab, in which case we set $p_{\text{enter}} = (1, 1)$).

Let \hat{B} be the subcurve corresponding to H . If \hat{B} contains only near points, we compute a (ε, δ) -safe transit exit $q_{\text{exit}} = (x', y')$ of H in $\mathcal{O}(\log nm)$ time with the algorithm of Section 3.2. Otherwise, we use the algorithm of Section 3.3, which takes $\mathcal{O}(\frac{1}{\varepsilon}(|R[x, x']| + |\hat{B}|) \log n \log nm)$ time. Both algorithms require $\mathcal{O}(n + m)$ preprocessing time. Taken over all slabs in \mathcal{H} , the total complexity of the subcurves \hat{B} is $\mathcal{O}(m)$. This gives the following result:

► **Theorem 15.** *For any $\varepsilon \in (0, 1]$, there is a $(1 + \varepsilon)$ -approximate decision algorithm running in $\mathcal{O}(\frac{1}{\varepsilon}(n + m) \log n \log nm)$ time.*

We turn the decision algorithm into an approximate optimization algorithm with a simple binary search. For this, we show that the geodesic Fréchet distance is not much greater than δ_H . This gives an accurate “guess” of the actual geodesic Fréchet distance.

► **Lemma 16.** $\delta_H \leq d_F(R, B) \leq 3\delta_H$.

For our approximate optimization algorithm, we perform binary search over the values $\delta_H, (1 + \varepsilon)\delta_H, \dots, 3\delta_H$ and run our approximate decision algorithm with each encountered parameter. This procedure yields a $(1 + \varepsilon)^2$ -approximation to the geodesic Fréchet distance. Since $(1 + \varepsilon)^2 \leq 1 + 3\varepsilon$ for $\varepsilon \in (0, 1]$, scaling ε by a factor of $1/3$ gives our main result:

► **Theorem 1.** *For any $\varepsilon \in (0, 1]$, we can compute a $(1 + \varepsilon)$ -approximation to $d_F(R, B)$ in $\mathcal{O}(\frac{1}{\varepsilon}(n + m \log n) \log nm \log \frac{1}{\varepsilon})$ time.*

4 Separated one-dimensional curves and propagating reachability

In this section we consider the following problem: Let \bar{R} and \bar{B} be two one-dimensional curves with n and m vertices, respectively, where \bar{R} lies left of the point 0 and \bar{B} right of it. We are given a set $S \subseteq \mathcal{F}_\delta(\bar{R}, \bar{B})$ of $\mathcal{O}(n+m)$ “entrances,” for some $\delta \geq 0$. Also, we are given a set $E \subseteq \mathcal{F}_\delta(\bar{R}, \bar{B})$ of $\mathcal{O}(n+m)$ “potential exits.” We wish to compute the subset of potential exits that are δ -reachable from an entrance. We call this procedure *propagating reachability information* from S to E . We assume that the points in S and E correspond to pairs of vertices of \bar{R} and \bar{B} . This assumption can be met by introducing $\mathcal{O}(n+m)$ vertices, which does not increase our asymptotic running times. Additionally, we may assume that all vertices of \bar{R} and \bar{B} have unique values, for example by a symbolic perturbation.

The problem of propagating reachability information has already been studied by Bringmann and Künnemann [4]. In case S lies on the left and bottom sides of the parameter space and E lies on the top and right sides, they give an $\mathcal{O}((n+m) \log nm)$ time algorithm. We are interested in a more general case however, where S and E may lie anywhere in the parameter space. We make heavy use of the concept of *prefix-minima* to develop an algorithm for our more general setting that has the same running time as the one described by Bringmann and Künnemann [4]. Furthermore, our algorithm is able to actually compute a Fréchet matching between \bar{R} and \bar{B} in linear time (see the full version of this paper), whereas Bringmann and Künnemann require near-linear time for only the decision version.

As mentioned above, we use prefix-minima extensively for our results in this section. Prefix-minima are those vertices that are closest to the separator 0 among those points before them on the curves. In the full version of this paper, we prove that a Fréchet matching exists that matches subcurves between consecutive prefix-minima to prefix-minima of the other curve, see Figure 7 for an illustration. We call these matchings *prefix-minima matchings*. This matching will end in a bichromatic closest pair of points, and so we can compose the matching with a symmetric matching for the reversed curves.

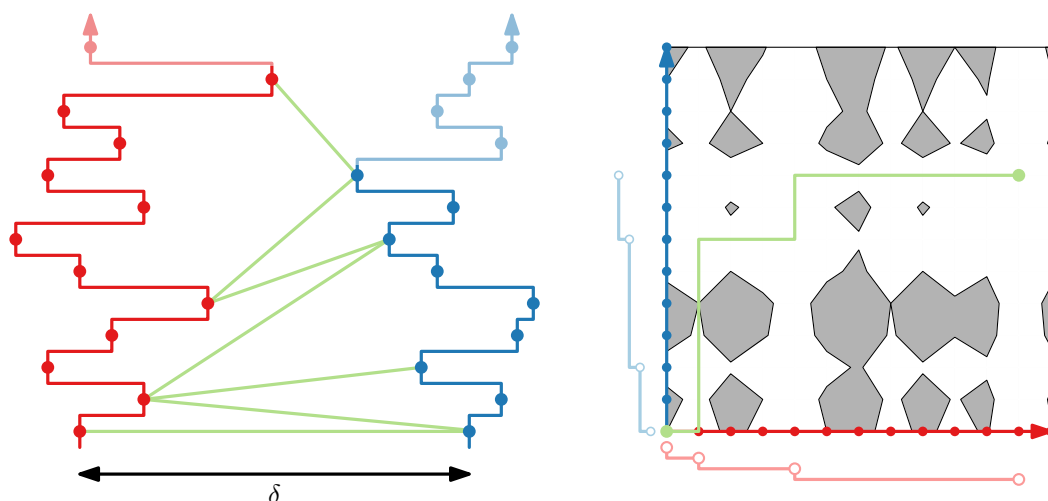


Figure 7 (left) A pair of separated, one-dimensional curves \bar{R} and \bar{B} , drawn stretched vertically for clarity, together with a prefix-minima matching. (right) The path in $\mathcal{F}_\delta(\bar{R}, \bar{B})$ corresponding to the matching.

In Section 4.1 we introduce two geometric forests in $\mathcal{F}_\delta(\bar{R}, \bar{B})$, with leaves at S , that capture multiple prefix-minima matchings at once. It is based on *horizontal-greedy* and *vertical-greedy* matchings. We show that these forests have linear complexity and can be computed efficiently.

In Section 4.2 we do not only go forward from points in S , but also backwards from points in E using *suffix-minima*. Again we have horizontal-greedy and vertical-greedy versions. Intersections between the two prefix-minima forests and the two suffix minima forests show the existence of a δ -free path of the corresponding points in S and E , so the problem reduces to a bichromatic intersection algorithm.

4.1 Greedy paths in the free space

We wish to construct a set of canonical prefix-minima δ -matchings in the free space from which we can deduce which points in E are reachable. Naturally, we want to avoid constructing a path between every point in S and every point in E . Therefore, we investigate certain classes of prefix-minima δ -matchings that allows us to infer reachability information with just two paths per point in S and two paths per point in E . Furthermore, these paths have a combined $\mathcal{O}(n + m)$ description complexity.

We first introduce one of the greedy matchings and prove a useful property. A *horizontal-greedy δ -matching* π_{hor} is a prefix-minima δ -matching starting at a point $s = (i, j)$ that satisfies the following property: Let (i', j') be a point on π_{hor} where $\bar{R}(i')$ and $\bar{B}(j')$ are prefix-minima of $\bar{R}[i, n]$ and $\bar{B}[j, m]$. If there exists a prefix-minimum $\bar{R}(\hat{i})$ of $\bar{R}[i, n]$ after $\bar{R}(i')$, and the horizontal line segment $[i', \hat{i}] \times \{j'\}$ lies in $\mathcal{F}_\delta(\bar{R}, \bar{B})$, then either π_{hor} traverses this line segment, or π_{hor} terminates in (i', j') .

For an entrance $s \in S$, let $\pi_{\text{hor}}(s)$ be the maximal horizontal-greedy δ -matching. See Figure 8 for an illustration. The path $\pi_{\text{hor}}(s)$ serves as a canonical prefix-minima δ -matching, in the sense that any point t that is reachable from s by a prefix-minima δ -matching is reachable from a point on $\pi_{\text{hor}}(s)$ through a single vertical segment:

► **Lemma 17.** *Let $s \in S$ and let t be a point that is reachable by a prefix-minima δ -matching from s . A point $\hat{t} \in \pi_{\text{hor}}(s)$ vertically below t exists for which the segment $\hat{t}t$ lies in $\mathcal{F}_\delta(\bar{R}, \bar{B})$.*

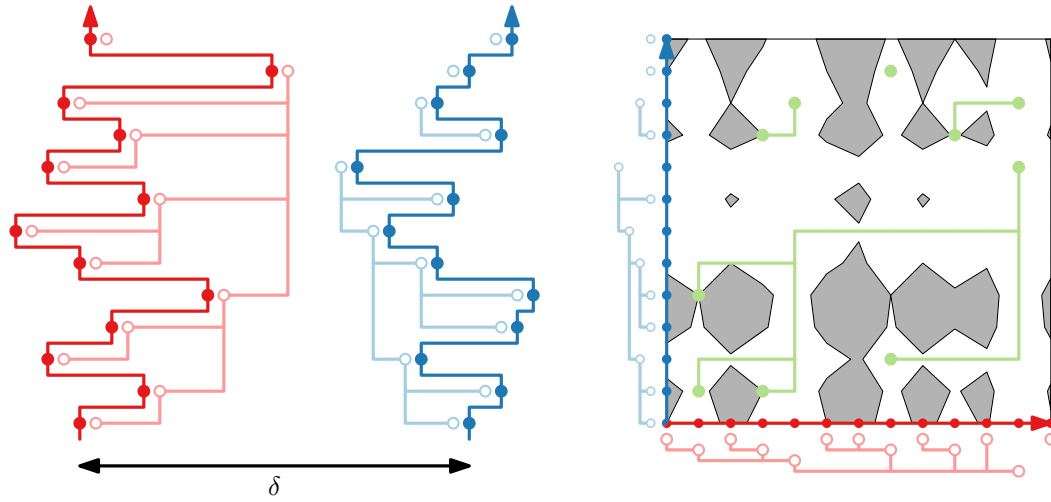
A single path $\pi_{\text{hor}}(s)$ may have $\mathcal{O}(n + m)$ complexity. We would like to construct the paths for all entrances, but this would result in a combined complexity of $\mathcal{O}((n + m)^2)$. However, due to the definition of the paths, if two paths $\pi_{\text{hor}}(s)$ and $\pi_{\text{hor}}(s')$ have a point (i, j) in common, then the paths are identical from (i, j) onwards. Thus, rather than explicitly describing the paths, we instead describe their union. Specifically, the set $\bigcup_{s \in S} \pi_{\text{hor}}(s)$ forms a geometric forest $\mathcal{T}_{\text{hor}}(S)$ whose leaves are the points in S , see Figure 8. This forest has low complexity and can be constructed efficiently:

► **Lemma 18.** *The forest $\mathcal{T}_{\text{hor}}(S)$ has $\mathcal{O}(n + m)$ vertices.*

► **Lemma 19.** *We can construct a geometric graph for $\mathcal{T}_{\text{hor}}(S)$ in $\mathcal{O}((n + m) \log nm)$ time.*

4.2 Propagating reachability

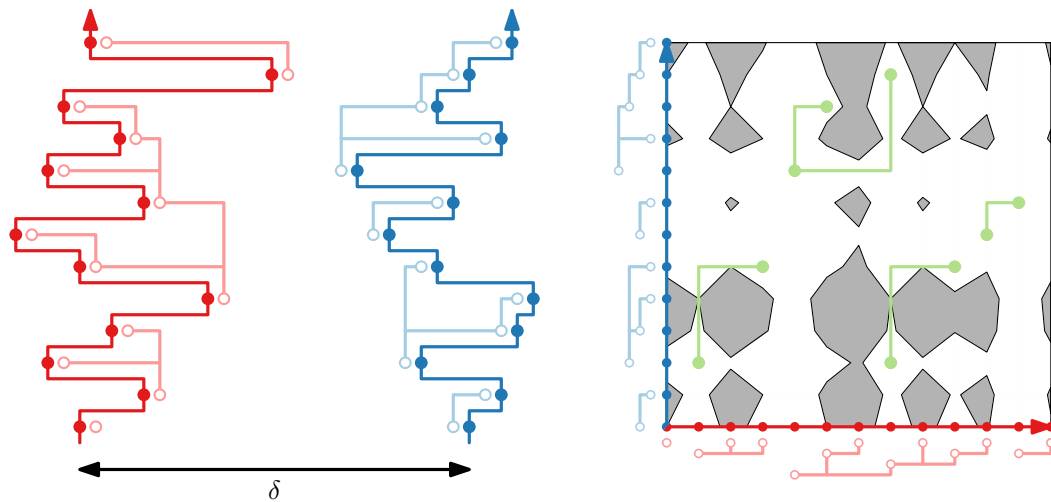
Next we give an algorithm for propagating reachability information. For the algorithm, we consider three more δ -matchings that are symmetric in definition to the horizontal-greedy δ -matchings. The first is the maximal *vertical-greedy* δ -matching $\pi_{\text{ver}}(s)$, which, as the name suggests, is the maximal prefix-minima δ -matching starting at an entrance



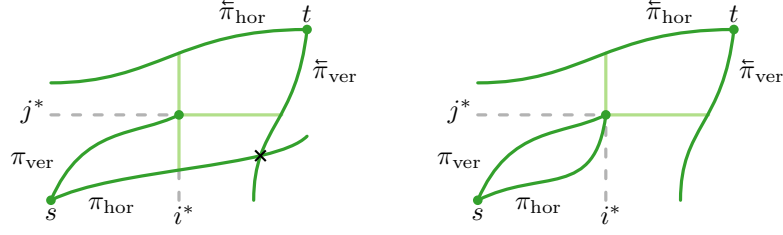
■ **Figure 8** (left) For every vertex, its next prefix-minimum is depicted as its parent in the respective tree. (right) The horizontal-greedy δ -matchings. Paths move monotonically to the right and up.

$s \in S$ that prioritizes vertical movement over horizontal movement. The other two require a symmetric definition to prefix-minima, namely *suffix-minima*. These are the vertices closest to 0 compared to the suffix of the curve after the vertex. The maximal *reverse* horizontal- and vertical-greedy δ -matchings $\bar{\pi}_{\text{hor}}(t)$ and $\bar{\pi}_{\text{ver}}(t)$, starting at a potential exit $t \in E$, are symmetric in definition to the maximal horizontal- and vertical-greedy δ -matching, except that they move backwards, to the left and down, and their vertices correspond to suffix-minima of the curves (see Figure 9).

Consider a point $s = (i, j) \in S$ and let $t = (i', j') \in E$ be δ -reachable from s . Let $\bar{R}(i^*)$ and $\bar{B}(j^*)$ form a bichromatic closest pair of $\bar{R}[i, i']$ and $\bar{B}[j, j']$. Note that these points are unique, by our general position assumption. We prove in the full version of this paper that (i^*, j^*) is δ -reachable from s , and that t is δ -reachable from (i^*, j^*) .



■ **Figure 9** (left) For every vertex, its previous suffix-minimum is shown as its parent in the tree. (right) The reverse horizontal-greedy δ -matchings. Paths move monotonically to the left and down.



■ **Figure 10** Two possible situations following from Lemmas 17 and 20. The paths starting at s or t are the four greedy matchings. The horizontal and vertical light green segments lie in $\mathcal{F}_\delta(\bar{R}, \bar{B})$. On the right, the extensions of $\pi_{\text{hor}}(s)$ and $\pi_{\text{ver}}(s)$ respectively intersect $\tilde{\pi}_{\text{ver}}(t)$ and $\tilde{\pi}_{\text{hor}}(t)$.

From Lemma 17 we have that $\pi_{\text{hor}}(s)$ has points vertically below (i^*, j^*) , and the vertical segment between $\pi_{\text{hor}}(s)$ and (i^*, j^*) lies in $\mathcal{F}_\delta(\bar{R}, \bar{B})$. We extend the property to somewhat predict the movement of $\pi_{\text{hor}}(s)$ near t :

► **Lemma 20.** *Either $\pi_{\text{hor}}(s)$ terminates in (i^*, j^*) , or it contains a point vertically below t or horizontally left of t .*

Based on Lemmas 17 and 20 and their symmetric counterparts, $\pi_{\text{hor}}(s)$ and $\pi_{\text{ver}}(s)$ satisfy the properties below, and $\tilde{\pi}_{\text{hor}}(t)$ and $\tilde{\pi}_{\text{ver}}(t)$ satisfy symmetric properties, see Figure 10.

- $\pi_{\text{hor}}(s)$ has a point vertically below (i^*, j^*) , and the vertical segment between $\pi_{\text{hor}}(s)$ and (i^*, j^*) lies in $\mathcal{F}_\delta(\bar{R}, \bar{B})$.
- $\pi_{\text{ver}}(s)$ has a point horizontally left of (i^*, j^*) , and the horizontal segment between $\pi_{\text{ver}}(s)$ and (i^*, j^*) lies in $\mathcal{F}_\delta(\bar{R}, \bar{B})$.
- $\pi_{\text{hor}}(s)$ and $\pi_{\text{ver}}(s)$ both either terminate in (i^*, j^*) , or contain a point vertically below t or horizontally left of t .

These properties mean that either $\pi_{\text{hor}}(s) \cup \pi_{\text{ver}}(s)$ intersects $\tilde{\pi}_{\text{hor}}(t) \cup \tilde{\pi}_{\text{ver}}(t)$, or the following extensions do: Let $\pi_{\text{hor}}^+(s)$ be the path obtained by extending $\pi_{\text{hor}}(s)$ with the maximum horizontal line segment in $\mathcal{F}_\delta(\bar{R}, \bar{B})$ whose left endpoint is the end of $\pi_{\text{hor}}(s)$. Define $\pi_{\text{ver}}^+(s)$ symmetrically, by extending $\pi_{\text{ver}}(s)$ with a vertical segment. Also define $\tilde{\pi}_{\text{hor}}^+(s)$ and $\tilde{\pi}_{\text{ver}}^+(s)$ analogously. By Lemma 17, $\pi_{\text{hor}}^+(s)$ or $\pi_{\text{ver}}^+(s)$ must intersect $\tilde{\pi}_{\text{hor}}^+(t)$ or $\tilde{\pi}_{\text{ver}}^+(t)$. Furthermore, if $\pi_{\text{hor}}^+(s)$ or $\pi_{\text{ver}}^+(s)$ intersects $\tilde{\pi}_{\text{hor}}^+(t')$ or $\tilde{\pi}_{\text{ver}}^+(t')$ for some potential exit $t' \in E$, then the bimonotonicity of the paths implies that t' is δ -reachable from s . Thus:

► **Lemma 21.** *A point $t \in E$ is δ -reachable from a point $s \in S$ if and only if $\pi_{\text{hor}}^+(s) \cup \pi_{\text{ver}}^+(s)$ intersects $\tilde{\pi}_{\text{hor}}^+(t) \cup \tilde{\pi}_{\text{ver}}^+(t)$.*

Recall that $\mathcal{T}_{\text{hor}}(S)$ represents all paths $\pi_{\text{hor}}(s)$. We augment $\mathcal{T}_{\text{hor}}(S)$ to represent all paths $\pi_{\text{hor}}^+(s)$. For this, we take each root vertex p and compute the maximal horizontal segment $\overline{pq} \subseteq \mathcal{F}_\delta(\bar{R}, \bar{B})$ that has p as its left endpoint. We compute this segment in $\mathcal{O}(\log n)$ time after $\mathcal{O}(n \log n)$ time preprocessing (for details, refer to the full version of this paper). We then add q as a vertex to $\mathcal{T}_{\text{hor}}(S)$, and add an edge from p to q .

Let $\mathcal{T}_{\text{hor}}^+(S)$ be the augmented graph. We define the graphs $\mathcal{T}_{\text{ver}}^+(S)$, $\tilde{\mathcal{T}}_{\text{hor}}^+(E)$ and $\tilde{\mathcal{T}}_{\text{ver}}^+(E)$ analogously. The four graphs have a combined complexity of $\mathcal{O}(n+m)$ and can be constructed in $\mathcal{O}((n+m) \log nm)$ time. Our algorithm computes the edges of $\tilde{\mathcal{T}}_{\text{hor}}^+(E)$ and $\tilde{\mathcal{T}}_{\text{ver}}^+(E)$ that intersect an edge of $\mathcal{T}_{\text{hor}}^+(S)$ or $\mathcal{T}_{\text{ver}}^+(S)$. We do so with a standard sweepline algorithm:

► **Lemma 22.** *Given sets of n “red” and m “blue,” axis-aligned line segments in \mathbb{R}^2 , we can report all segments that intersect a segment of the other color in $\mathcal{O}((n+m) \log nm)$ time.*

Suppose we have computed the set of edges \mathcal{E} of $\tilde{\mathcal{T}}_{\text{hor}}^+(E)$ and $\tilde{\mathcal{T}}_{\text{ver}}^+(E)$ that intersect an edge of $\mathcal{T}_{\text{hor}}^+(S)$ or $\mathcal{T}_{\text{ver}}^+(S)$. We store \mathcal{E} in a red-black tree, so that we can efficiently retrieve and remove edges from this set. Let $e \in \mathcal{E}$ and suppose e is an edge of $\tilde{\mathcal{T}}_{\text{hor}}^+(E)$. Let μ be the top-right vertex of e . All potential exits of E that are stored in the subtree of μ are reachable from a point in S . We traverse the entire subtree of μ , deleting every edge we find from \mathcal{E} . Every point in E we find is marked as reachable. In this manner, we obtain:

► **Theorem 23.** *Let \bar{R} and \bar{B} be two separated one-dimensional curves with n and m vertices. Let $\delta \geq 0$, and let $S, E \subseteq \mathcal{F}_\delta(\bar{R}, \bar{B})$ be sets of $\mathcal{O}(n + m)$ points. We can compute the set of all points in E that are δ -reachable from points in S in $\mathcal{O}((n + m) \log nm)$ time.*

5 Conclusion

We studied computing the approximate geodesic Fréchet distance of two curves R and B that bound a simple polygon P , one clockwise and one counterclockwise, whose endpoints meet. Our algorithm is approximate, though the only approximate parts are for matching the far points and turning the decision algorithm into an optimization algorithm. Doing so exactly and in strongly subquadratic time remains an interesting open problem.

Our algorithm extends to the case where R and B do not cover the complete boundary of the polygon. In other words, the start and endpoints of R and B need not coincide. Geodesics between points on R and B must stay inside P . In this case, $k = |P|$ can be much greater than $n + m - 2$, which influences the preprocessing and query times of various data structures we use. The running time then becomes: $\mathcal{O}(k + \frac{1}{\epsilon}(n + m \log n) \log k \log \frac{1}{\epsilon})$.

References

- 1 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995. doi:10.1142/S0218195995000064.
- 2 Lotte Blank and Anne Driemel. A faster algorithm for the Fréchet distance in 1D for the imbalanced case. In *Proc. 32nd Annual European Symposium on Algorithms (ESA)*, volume 308 of *LIPICs*, pages 28:1–28:15, 2024. doi:10.4230/LIPICs.ESA.2024.28.
- 3 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–670, 2014. doi:10.1109/FOCS.2014.76.
- 4 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.
- 5 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.
- 6 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 *Proc. 30th Annual Symposium on Discrete Algorithms (SODA)*, pages 2887–2901, 2019. doi:10.1137/1.9781611975482.179.
- 7 Erin W. Chambers, Éric Colin de Verdière, Jeff Erickson, Sylvain Lazard, Francis Lazarus, and Shripad Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010. doi:10.1016/J.COMGEO.2009.02.008.
- 8 Siu-Wing Cheng, Haoqiang Huang, and Shuo Zhang. Constant approximation of Fréchet distance in strongly subquadratic time. *CoRR*, abs/2503.12746, 2025. doi:10.48550/arXiv.2503.12746.

- 9 Connor Colombe and Kyle Fox. Approximating the (continuous) Fréchet distance. In *Proc. 37th International Symposium on Computational Geometry (SoCG)*, pages 26:1–26:14, 2021. doi:10.4230/LIPIcs.SoCG.2021.26.
- 10 Atlas F. Cook IV and Carola Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms*, 7(1):9:1–9:19, 2010. doi:10.1145/1868237.1868247.
- 11 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.
- 12 Alon Efrat, Leonidas J. Guibas, Sariel Har-Peled, Joseph S. B. Mitchell, and T. M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, 28(4):535–569, 2002. doi:10.1007/S00454-002-2886-1.
- 13 Sariel Har-Peled, Amir Nayyeri, Mohammad R. Salavatipour, and Anastasios Sidiropoulos. How to walk your dog in the mountains with no magic leash. *Discrete & Computational Geometry*, 55(1):39–73, 2016. doi:10.1007/S00454-015-9737-3.
- 14 Thijs van der Horst, Marc van Kreveld, Tim Ophelders, and Bettina Speckmann. A near-linear time exact algorithm for the l_1 -geodesic Fréchet distance between two curves on the boundary of a simple polygon. To appear at the Algorithms and Data Structures Symposium (WADS), 2025.
- 15 Thijs van der Horst, Marc J. van Kreveld, Tim Ophelders, and Bettina Speckmann. A subquadratic n^ε -approximation for the continuous Fréchet distance. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1759–1776, 2023. doi:10.1137/1.9781611977554.ch67.