International Journal of Computational Geometry & Applications

Fast Fréchet Distance Between Curves with Long Edges*

Joachim Gudmundsson[†]

University of Sydney, Australia
joachim.gudmundsson@sydney.edu.au

Majid Mirzanezhad[‡]

Tulane University, New Orleans, USA
mmirzane@tulane.edu

Ali Mohades

Amirkabir University of Technology, Iran
mohades@aut.ac.ir

Carola Wenk[‡]

Tulane University, New Orleans, USA
cwenk@tulane.edu

ABSTRACT

Computing the Fréchet distance between two polygonal curves takes roughly quadratic time. In this paper, we show that for a special class of curves the Fréchet distance computations become easier. Let P and Q be two polygonal curves in $\mathbb{R}^d$ with n and m vertices, respectively. We prove four results for the case when all edges of both curves are long compared to the Fréchet distance between them: (1) a linear-time algorithm for deciding the Fréchet distance between two curves, (2) an algorithm that computes the Fréchet distance in $O((n+m)\log(n+m))$ time, (3) a linear-time $\sqrt{d}$-approximation algorithm, and (4) a data structure that supports $O(m\log^2 n)$-time decision queries, where m is the number of vertices of the query curve and n the number of vertices of the preprocessed curve.

Keywords: The Fréchet distance, Approximation algorithm, Data structure.

1. Introduction

Measuring the similarity between two curves is an important problem that has applications in many areas, e.g., in morphing,[3] movement analysis,[4] handwriting recognition[5] and protein structure alignment.[6] Fréchet distance is one of the most

*Preliminary versions of this paper appeared in[1, 2]

[†]This research was supported under the Australian Research Councils Discovery Projects funding scheme (DP150101134).

[‡]This research was supported by the National Science Foundation under the grant (CCF-1637576).

popular similarity measures which has received considerable attentions in recent years. It is intuitively the minimum length of the leash that connects a man and dog walking across the curves without going backward. The classical algorithm for computing the Fréchet distance between curves with total complexity n runs in $O(n^2 \log n)$ time.[7] The major goal of this paper is to focus on computing the Fréchet distance for a reasonable special class of curves in significantly faster than quadratic time.

1.1. *Related Work*

Buchin et al.[8] gave an $\Omega(n \log n)$ lower bound for computing the Fréchet distance. Then Bringmann[9] showed that, assuming the Strong Exponential Time Hypothesis, the Fréchet distance cannot be computed in strongly subquadratic time, i.e., in time $O(n^{2-\epsilon})$ for any $\epsilon > 0$. For the *discrete* Fréchet distance, which considers only distances between the vertices, Agarwal et al.[10] gave an algorithm with a (mildly) subquadratic running time of $O(n^2 \frac{\log \log n}{\log n})$. Buchin et al.[11] showed that the continuous Fréchet distance can be computed in $O(n^2 \sqrt{\log n}(\log \log n)^{3/2})$ expected time. Bringmann and Mulzer[12] gave an $O(n^2/\phi + n \log n)$-time algorithm to compute a ϕ-approximation of the discrete Fréchet distance for any integer $1 \leq \phi \leq n$. Therefore, an n^ϵ-approximation, for any $\epsilon > 0$, can be computed in (strongly) subquadratic time.

For the continuous Fréchet distance, there are also a few subquadratic algorithms known for restricted classes of curves such as κ-bounded, backbone and c-packed curves. Alt et al.[13] considered κ-bounded curves and they gave an $O(n \log n)$ time algorithm to $(\kappa + 1)$-approximate the Fréchet distance. A curve P is κ-bounded if for any two points $x, y \in P$, the union of the balls with radii r centered at x and y contains the whole $P[x, y]$ where r is equal to $(\kappa/2)$ times the Euclidean distance between x and y. For any $\epsilon > 0$, Aronov et al.[14] provided a near-linear time $(1+\epsilon)$-approximation algorithm for the discrete Fréchet distance for so-called backbone curves that have essentially constant edge length and require a minimum distance between non-consecutive vertices. For c-packed curves a $(1 + \epsilon)$-approximation can be computed in $O(cn/\epsilon + cn \log n)$ time.[15] A curve is c-packed if for any ball B, the length of the portion of P contained in B is at most c times the diameter of B.

1.2. *Our Contribution*

In this paper, we study a new class of curves, namely curves with long edges, and we show that for these curves the Fréchet distance can be computed significantly faster than quadratic time. In a particular application, one might be interested in detecting groups of different movement patterns in migratory birds that fly very long distances. As shown in Fig. 1, different flyways are comparatively straight and the trajectory data of individual birds usually consists of only one GPS sample per day in order to conserve battery power. Infrequent sampling and the straight flyways therefore result in curves with long edges, and it is desirable to compare the routes of different animals in order to identify common flyways.

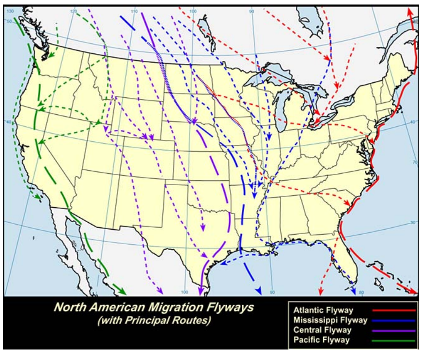


Fig. 1. There are four typical flyways across the US. Clustering the trajectories by similarity between them allows us to detect the most common movement pattern.[16]

We consider the decision, optimization, approximation and data structure problems for the Fréchet distance between two polygonal curves P and Q in $\mathbb{R}^d$ with n and m vertices, respectively, all for the case where all edges of both curves are long compared to the Fréchet distance between them. In Section 3 we present a greedy linear-time algorithm for deciding whether the Fréchet distance is at most ε, as long as all edges in P are longer than 2ε and edges in Q are longer than $(1+\sqrt{d})\varepsilon$. In Section 4 we give an algorithm for computing the Fréchet distance in $O((n+m)\log(n+m))$ time and a linear-time algorithm for approximating the Fréchet distance up to a factor of $\sqrt{d}$. In Section 5 we present a data structure that decides whether the Fréchet distance between a preprocessed curve P and a query curve Q is at most ε or not, in $O(m\log^2 n)$ query time using $O(n\log n)$ space and preprocessing time.

2. Preliminaries

In this section we provide notations and definitions that will be required in the next sections. Let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves with vertices $p_1,\dots,p_n$ and $q_1,\dots,q_m$, respectively. We treat a *polygonal curve* as a continuous map $P : [1,n] \to \mathbb{R}^d$ where $P(i) = p_i$ for an integer i, and the i-th edge is linearly parametrized as $P(i+\lambda) = (1-\lambda)p_i + \lambda p_{i+1}$, for integer i and $0 < \lambda < 1$. A *re-parametrization* $\sigma : [0,1] \to [1,n]$ of P is any continuous, non-decreasing function such that $\sigma(0) = 1$ and $\sigma(1) = n$. We denote a re-parametrization of Q by $\theta : [0,1] \to [1,m]$. We denote the length of the shortest edge in P and the length of the shortest edge in Q by l_P and l_Q, respectively. For two points $x, y \in \mathbb{R}^d$, let $\|x-y\|$ denote the Euclidean distance between the points and $\overline{xy}$ the straight line segment connecting x to y. The Euclidean distance between $x \in \mathbb{R}^d$ and an edge $e : [1, 2] \to \mathbb{R}^d$ is denoted as $\|x, e\| = \min_{1 \le t \le 2} \|x - e(t)\|$. For $1 \le a \le b \le n$, $P[a, b]$ denotes the subcurve of P starting in $P(a)$ and ending in $P(b)$. Let $\varepsilon > 0$ be a real number. Consider an edge $e : [1, 2] \to \mathbb{R}^d$ of length $\|e\| > 2\varepsilon$ whose endpoints are e_1 and e_2. The *direction vector* of e is the vector from e_1 to e_2. Now let $B(p, \varepsilon) = \{x \in \mathbb{R}^d \mid \|p - x\| \le \varepsilon\}$ be the ball with radius ε that is centered at a point p. The *cylinder* $C(e, \varepsilon)$ is the set of points in $\mathbb{R}^d$ within distance ε from e, i.e., $C(e, \varepsilon) = \cup_{x \in e} B(x, \varepsilon)$. We say P is (e, ε)-*monotone* if (1) $p_1 \in B(e_1, \varepsilon)$ and $p_n \in B(e_2, \varepsilon)$, (2) $P \subseteq C(e, \varepsilon)$, and (3) P is monotone with respect to the line supporting e. A curve is monotone with respect to a line l if it intersects any hyperplane perpendicular to l in at most one component.

2.1. *Fréchet Distance and Free-Space Diagram*

To compute the Fréchet distance between P and Q, Alt and Godau[7] introduced the notion of *free-space diagram*. For any $\varepsilon > 0$, we denote the free-space diagram between P and Q by $\mathcal{FSD}_{\le\varepsilon}(P, Q)$. This diagram has the domain $[1, n] \times [1, m]$ and it consists of $(n - 1) \times (m - 1)$ cells, where each point (s, t) in the diagram corresponds to two points $P(s)$ and $Q(t)$. A point (s, t) in $\mathcal{FSD}_{\le\varepsilon}(P, Q)$ is called *free* if $\|P(s) - Q(t)\| \le \varepsilon$ and *blocked*, otherwise. The union of all free points is referred to as the *free space*. A *monotone matching* between P and Q is a pair of re-parameterizations (σ, θ) corresponding to an xy-monotone path from $(1, 1)$ to (n, m) within the free space in $\mathcal{FSD}_{\le\varepsilon}(P, Q)$. The *Fréchet distance* between two

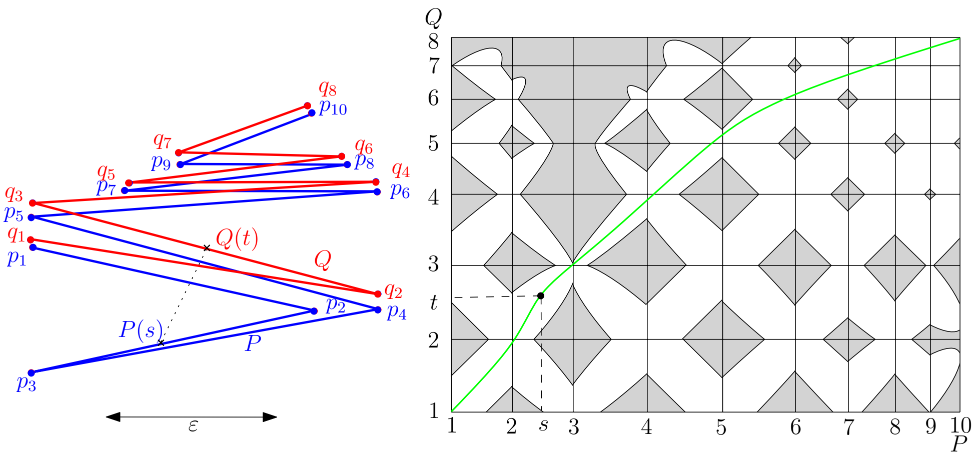


Fig. 2. Two curves P, Q and $\varepsilon > 0$ on the left, and the free space diagram $\mathcal{FSD}_{\le\varepsilon}(P, Q)$ on the right showing free space in white and blocked space in gray. A reachable path is shown in green. The point (s, t) lies in free space. There is a quadratic number of cells containing free space as well as a quadratic number of cells containing blocked space in $\mathcal{FSD}_{\le\varepsilon}(P, Q)$ and all of them may need to be checked to decide reachability for (n, m). Note that both P and Q contain short edges as well as long edges compared to ε.

curves is defined as $\delta_F(P,Q) = \inf_{(\sigma,\theta)} \max_{0\leq t\leq 1} \|P(\sigma(t)) - Q(\theta(t))\|$, where (σ,θ) is a monotone matching and $\max_{0\leq t\leq 1} \|P(\sigma(t)) - Q(\theta(t))\|$ is called the *width* of the matching. A monotone matching realizing $\delta_F(P,Q)$ is called a *Fréchet matching.* A point (s,t) is *reachable* if there exists a Fréchet matching from $(1,1)$ to (s,t) in $\mathcal{FSD}_{\leq\varepsilon}(P,Q)$. A Fréchet matching in $\mathcal{FSD}_{\leq\varepsilon}(P,Q)$ from $(1,1)$ to (s,t) is also called a *reachable path* for (s,t) (see Fig. 2). Alt and Godau[7] compute a reachable path by propagating reachable points across free space cell boundaries in a dynamic programming manner, which requires the exploration of the entire $\mathcal{FSD}_{\leq\varepsilon}(P,Q)$ and takes $O(mn)$ time.

2.2. *The Main Idea*

We set out to provide faster algorithms for the Fréchet distance using implicit structural properties of the free-space diagram of curves with long edges. These properties allow us to develop greedy algorithms that construct valid re-parameterizations by repeatedly computing a maximally reachable subcurve on one of the curves. Like the greedy algorithm proposed by Bringmann and Mulzer,[12] we compute prefix subcurves that have a valid Fréchet distance. However, while the approximation ratio of their greedy algorithm is exponential, the approximation ratio of the algorithm we present in Section 4.2 is constant, because we can take advantage of the curves having long edges. Our assumption on edge lengths is more general than backbone curves, since we do not require that non-consecutive vertices be far away from each other and we do not require any upper bound on the length of the edges.

The free space diagram for curves with long edges is simpler, and intuitively seems to have fewer reachable paths (see Fig. 3). In the remainder of this paper we show that indeed we can exploit this simpler structure to compute reachable paths in a simple greedy manner which results in runtimes that are significantly faster than quadratic.

3. A Greedy Decision Algorithm

In this section we give a linear time algorithm for deciding whether the Fréchet distance between two polygonal curves P and Q in $\mathbb{R}^d$ with relatively long edges is at most ε. In Section 3.1, we first prove a structural property for the case that each edge in P is longer than 2ε and Q is a single segment. Afterwards in Section 3.2, we consider the extension to the case that P and Q are two polygonal curves and we show some extended structural property of free space induced by two curves with long edges. In Section 3.3, we present our greedy algorithm, which is based on computing longest reachable prefixes in P with respect to each segment in Q. We consider three different variants of edge lengths assumption when $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{d})\varepsilon$ (Section 3.3.1), $l_P \geq 2\varepsilon$ and $l_Q \geq (1+\sqrt{d})\varepsilon$ (Section 3.3.2), and $l_P > 0$ and $l_Q > 4\varepsilon$ (Section 3.3.3). In Section 3.4, we provide a critical example for which our greedy algorithm fails when the assumption on the edge lengths does not hold.

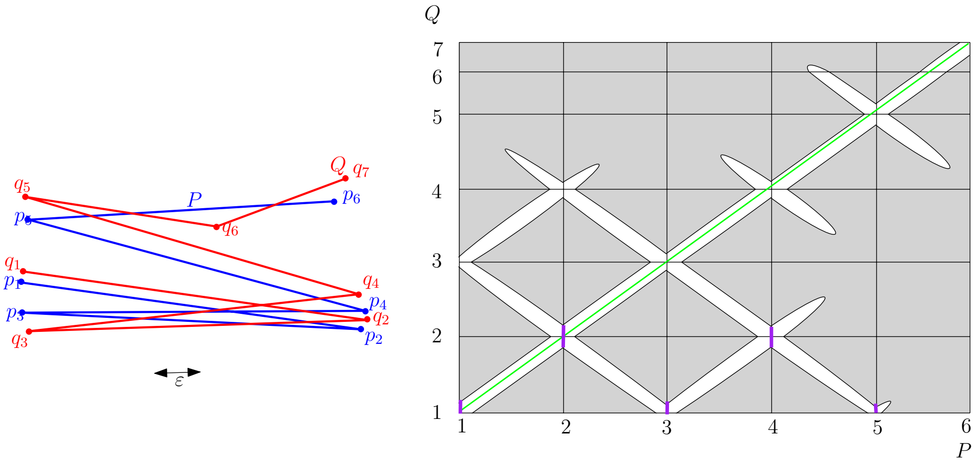


Fig. 3. $\mathcal{FSD}_{\leq\varepsilon}(P,Q)$ for curves with long edges results in fewer reachable paths for (n,m). Consider the vertical free intervals (shown in purple) in the first row of the free space diagram. Since $l_P > 2\varepsilon$, no consecutive purple intervals intersect which is a property we exploit. One can use such a property to find a reachable path without needing to check the entire free space diagram.

3.1. *A Simple Fréchet Matching for a Single Segment*

In this section we start by introducing the crucial notion of *orthogonal matching* between a polygonal curve P and a single line segment e. An orthogonal matching projects each point from P to its closest point on e. In particular, it maps vertices of P either orthogonally to the segment e or directly to the endpoints of e.

Definition 1 (Orthogonal Matching). Let $\varepsilon > 0$, $P : [1, n] \to \mathbb{R}^d$ be a polygonal curve, and $e : [1, 2] \to \mathbb{R}^d$ be a line segment. A Fréchet matching (σ, θ) realizing $\delta_F(P, e) \leq \varepsilon$ is called an *orthogonal matching* of width at most ε if $\sigma(t) = 1$ for $t \in [0, a]$, $\|P(\sigma(t)) - e(\theta(t))\| = \|P(\sigma(t)), e\| \leq \varepsilon$ for $t \in (a, b)$, and $\sigma(t) = n$ for $t \in [b, 1]$ for some $0 \leq a \leq b \leq 1$; see Fig. 4(a).

Now we state a key lemma that demonstrates that if P has long edges, then the orthogonal matching of width at most ε between P a segment e exists if and only if $\delta_F(P, e) \leq \varepsilon$, and this is equivalent to P being (e, ε)-monotone.

Lemma 1 (Orthogonal Matching and Monotonicity). *Let* $\varepsilon > 0$, $P : [1, n] \to \mathbb{R}^d$ *be a polygonal curve and* $e : [1, 2] \to \mathbb{R}^d$ *be a line segment. Consider the following statements:*

(1) $\delta_F(P, e) \leq \varepsilon$,
(2) P *is* (e, ε)*-monotone,*
(3) P *and* e *admit an* orthogonal matching *of width at most* ε.

In general, (2) $\Leftrightarrow$ *(3) and (3)* $\Rightarrow$ *(1). In addition, if* $l_P > 2\varepsilon$ *then (1)* $\Rightarrow$ *(2), i.e., all three statements are equivalent.*

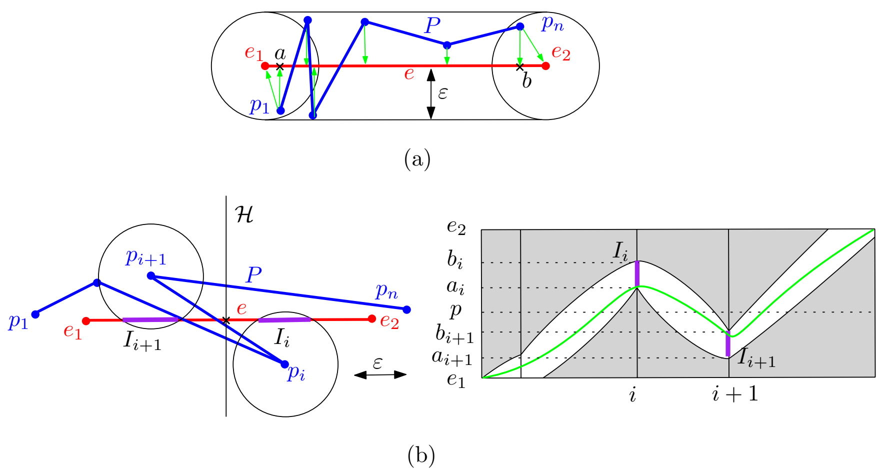


Fig. 4. (a) In this example P is (e, ε)-monotone and the green arrows indicate an orthogonal matching between P and e. (b) An illustration of the case $\mathcal{H} \cap e \neq \emptyset$ in the proof of (1) $\Rightarrow$ (2) in Lemma 1. Note that the consecutive purple intervals I_i and I_{i+1} do not intersect because $l_P > 2\varepsilon$.

Proof. We immediately have (3) $\Rightarrow$ (1) by Definition 1. To prove (2) $\Rightarrow$ (3), assume P is (e, ε)-monotone. We can construct an orthogonal matching by mapping each p_i to its nearest neighbor $e(1 + t_i)$ on e, with $0 \leq t_{i-1} \leq t_i \leq 1$. We set $\sigma(t_i) = i$ and $\theta(t_i) = 1 + t_i$ for all $i = 1, \ldots, n$, and we set $a = t_1$, $b = t_n$, $\sigma(t) = 1$ for $t \in [0, a]$, $\sigma(t) = n$ for $t \in [b, 1]$, and $\theta(0) = 1$ and $\theta(1) = 2$. The matching (σ, θ) is obtained by linearly interpolating between these values. The function $\sigma(t)$ is monotone by construction, and $\theta(t)$ is monotone because P is monotone with respect to the line supporting e. And all distances $\|P(\sigma(t)) - e(\theta(t))\| \leq \varepsilon$ because P is (e, ε)-monotone. Thus (σ, θ) is an orthogonal matching of width at most ε. To prove (3) $\Rightarrow$ (2), let (σ, θ) be an orthogonal matching of width at most ε. Then clearly $p_1 \in B(e_1, \varepsilon)$, $p_n \in B(e_2, \varepsilon)$, and $P \subseteq C(e, \varepsilon)$. Let $t_1, \ldots, t_n$ be such that $P(\sigma(t_i)) = p_i$ for $i = 1, \ldots, n$. Since (σ, θ) is a (monotone) Fréchet matching, $\theta(t_1), \ldots, \theta(t_n)$ is a monotone increasing sequence. And since (σ, θ) is orthogonal, the line segments $\overline{p_i\theta(t_i)}$ are all monotone to the line ℓ supporting e. Therefore, P is monotone with respect to ℓ and thus P is (e, ε)-monotone.

Now assume $l_P > 2\varepsilon$. In order to prove (1) $\Rightarrow$ (2), if $\delta_F(P, e) \leq \varepsilon$ then clearly $p_1 \in B(e_1, \varepsilon)$, $p_n \in B(e_2, \varepsilon)$, and $P \subseteq C(e, \varepsilon)$. It remains to show that P is monotone with respect to the line ℓ supporting e. For all $i = 1, \ldots, n$, define $I_i = B(p_i, \varepsilon) \cap e = e[a_i, b_i]$. Because $l_P > 2\varepsilon$, we know that $I_i \cap I_{i+1} = \emptyset$. Let (σ, θ) be a monotone matching realizing $\delta_F(P, e) \leq \varepsilon$. For the sake of contradiction assume there exists a hyperplane $\mathcal{H}$ perpendicular to ℓ such that P intersects $\mathcal{H}$ in at least two points $P(x)$ and $P(y)$, where $x < y$. Let p_i be the last vertex along $P[x, y]$, and recall that e_1 and e_2 are the two vertices of e. First assume that $\mathcal{H} \cap e \neq \emptyset$. Then p_i lies on the e_2-side of $\mathcal{H}$ and p_{i+1} lies on the e_1-side of $\mathcal{H}$. Therefore, because $I_i \cap I_{i+1} = \emptyset$, we know that $a_i > b_{i+1}$. Let $t_i, t_{i+1} \in [0,1]$ be two values such that $p_i = P(\sigma(t_i))$ and $p_{i+1} = P(\sigma(t_{i+1}))$, where $t_i < t_{i+1}$. From $\sigma(t_i) \geq a_i$ and $\sigma(t_{i+1}) \leq b_{i+1}$, we know that $\sigma(t_i) > \sigma(t_{i+1})$, which violates the monotonicity of (σ, θ), see Fig. 4(b). Now consider the case that $\mathcal{H} \cap e = \emptyset$. Then p_i lies on one side of $\mathcal{H}$, and e lies entirely on the other side. If $\mathcal{H} \cap B(e_1, \varepsilon) \neq \emptyset$, then we know that $P[1, y] \subseteq B(e_1, \varepsilon)$. But this is not possible since all edges of P are longer than 2ε. The same argument holds if $\mathcal{H} \cap B(e_2, \varepsilon) \neq \emptyset$. □

In fact Lemma 1 shows that for a curve P with long edges, the Fréchet distance to a line segment e is determined by examining whether P is (e, ε)-monotone or not.

3.2. *A Simple Fréchet Matching for More than One Segment*

In this section, we extend the matching between a curve P and a single line-segment e to a matching between two curves P and Q.

Definition 2 (Longest ε-Prefix). Let $\varepsilon > 0$, $P : [1, n] \to \mathbb{R}^d$ be a polygonal curve, and $e : [1, 2] \to \mathbb{R}^d$ be a line segment. Define $\gamma = \max\{ t \mid 1 \leq t \leq n \text{ and } \delta_F(P[1, t], e) \leq \varepsilon\}$. We call $P[1, \gamma]$ the *longest ε-prefix of P with respect to e.*

We now use the longest ε-prefix to define an extension of the matching introduced in Definition 1. Definition 2 is the basis of our greedy algorithm (Algorithm 1) which is presented in the next section. We show that if there exists a matching between two curves, then one can necessarily cut it into $m - 1$ orthogonal matchings between each segment in Q and the corresponding longest ε-prefix. Before we reach this property, we need the following technical lemma:

Lemma 2 ($(\sqrt{d}\varepsilon)$-Ball). *Let $\varepsilon > 0$ and let $P : [1, n] \to \mathbb{R}^d$ be a polygonal curve such that $l_P > 2\varepsilon$. Let $e : [1, 2] \to \mathbb{R}^d$ where $\|e\| > 2\varepsilon$. Assume that $P[1, \gamma]$ is the longest ε-prefix of P with respect to e, and let α be a parameter such that $P(\alpha)$ is the first point along P that intersects $B(e_2, \varepsilon)$. Then $P[\alpha, \gamma] \subseteq B(e_2, \sqrt{d}\varepsilon)$.*

Proof. By assumption $\|e\| > 2\varepsilon$, we know that $B(e_1, \varepsilon) \cap B(e_2, \varepsilon) = \emptyset$, thus α exists. Notice that $P[\alpha, \gamma] \subseteq C(e, \varepsilon)$. Let $\mathcal{H}$ be the hyperplane that is intersecting and perpendicular to e and is tangent to $B(e_2, \varepsilon)$. Hence $\mathcal{H}$ splits $P[1, \gamma]$ into two parts, the part on the e_1-side and the part that on the e_2-side. Let $P(x)$ be the last vertex before $P(\gamma)$ along P. By Definition 2, $\delta_F(P[1, \gamma], e) \leq \varepsilon$, and (1) if $P(x) \in B(e_2, \varepsilon)$, then Lemma 1 implies that $P[1, x]$ is (e, ε)-monotone. Thus $P[\alpha, \gamma]$ must lie on the e_2-side of $\mathcal{H}_2$, and in particular inside the cube enclosing $B(e_2, \varepsilon)$, see Fig. 5. Therefore the maximum possible distance between any point in $P[\alpha, \gamma]$ and e_2 is $\sqrt{d}\varepsilon$. (2) If $P(x) \notin B(e_2, \varepsilon)$, we first show that $P[1, x]$ is monotone with respect to the line supporting e and then we use the similar argument as in (1) to imply the maximum possible distance between any point in $P[\alpha, \gamma]$ and e_2 is $\sqrt{d}\varepsilon$. Now let (σ, θ) be a Fréchet matching between $P[1, \gamma]$ and e. For the sake of contradiction assume there exists an edge $P[i, i + 1]$ such that the angle between the direction vectors of $P[i, i+1]$ and e is greater than $\pi/2$ with $i < x$. Let $t_i, t_{i+1} \in [0,1]$ be two real values with $t_i < t_{i+1}$ such that $\sigma(t_i) = i$ and $\sigma(t_{i+1}) = i+1$ and let $I_i = B(p_i, \varepsilon) \cap e = e[a_i, b_i]$ and $I_{i+1} = B(p_{i+1}, \varepsilon) \cap e = e[a_{i+1}, b_{i+1}]$. Now from $B(p_i, \varepsilon) \cap B(p_{i+1}, \varepsilon) = \emptyset$ follows that $I_i \cap I_{i+1} = \emptyset$. Note that the angle between the direction vectors of $P[i, i+1]$ and e is greater than $\pi/2$ which indicates that $b_{i+1} < a_i$. Therefore $a_{i+1} \leq \theta(t_{i+1}) \leq b_{i+1} < a_i \leq \theta(t_i) \leq b_i$. Now three following cases are expected: (i) if $i+1 < \alpha$, then γ does not exist since (σ, θ) is not monotone and this would be a contradiction. Therefore $P[1, x]$ is monotone with respect to the line supporting e. (ii) If $\alpha < i \leq x$, then $\gamma < x$ since $i < \gamma < i+1$ which is a contradiction with $\gamma > x$. Hence $P[1, x]$ is monotone with respect to the line supporting e. (iii) if $i = x \leq \alpha$, then $P[\alpha, \gamma]$ is only a subsegment of $P[i, i+1]$ and trivially lies within $B(e_2, \varepsilon)$. This completes the proof. □

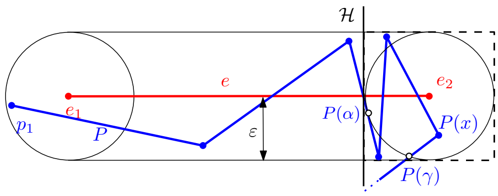


Fig. 5. The farthest point in $P[\alpha, \gamma]$ from e_2 must lie inside the cube enclosing $B(e_2, \varepsilon)$.

Lemma 3 ((3ε)-Ball). *Let $\varepsilon > 0$ and let $P : [1, n] \to \mathbb{R}^d$ be a polygonal curve. Let $e : [1, 2] \to \mathbb{R}^d$ where $\|e\| > 2\varepsilon$. Assume that $P[1, \gamma]$ is the longest ε-prefix of P with respect to e and $P(\alpha)$ is the first point along P that intersects $B(e_2, \varepsilon)$. Then $P[\alpha, \gamma] \subseteq B(e_2, 3\varepsilon)$.*

Proof. Although the proof of Lemma 11 in Gudmundsson and Smid[17] is similar, we describe a slight modification of the proof that is necessary for our setting. Suppose (σ, θ) is a Fréchet matching realizing $\delta_F(P[1, \gamma], e) \leq \varepsilon$. Let $x \in [\alpha, \gamma]$ such that $P(x)$ is the farthest point to e_2. We need to show that $\|P(x) - e_2\| \leq 3\varepsilon$ which implies $P[\alpha, \gamma] \subseteq B(e_2, 3\varepsilon)$. Let $t_\alpha, t_\gamma \in [0, 1]$ be two values such that $\alpha = \sigma(t_\alpha)$ and $\gamma = \sigma(t_\gamma)$. Note that there exists some $t_x \in [t_\alpha, t_\gamma]$ such that $x = \sigma(t_x)$. By the triangle inequality we have:

$$\|P(x) - e_2\| \leq \|P(x) - e(\theta(t_x))\| + \|e(\theta(t_x)) - e_2\| \leq \varepsilon + \|e(\theta(t_x)) - e_2\|.$$

Note that $t_x > t_\alpha$ and we can have $\|e(\theta(t_x)) - e_2\| \leq \|e(\theta(t_\alpha)) - e_2\|$, hence:

$$\|P(x) - e_2\| \leq \varepsilon + \|e(\theta(t_\alpha)) - e_2\|.$$

By applying the triangle inequality once more we have:

$$\|P(x) - e_2\| \leq \varepsilon + \|e(\theta(t_\alpha)) - P(\alpha)\| + \|P(\alpha) - e_2\| \leq 3\varepsilon. \qquad \square$$

Now we show that if $\delta_F(P,Q) \le \varepsilon$, then the two polygonal curves P and Q admit a piecewise orthogonal matching, which can be obtained by computing longest ε-prefixes of P with respect to each segment of Q. This lemma is the foundation of our greedy algorithm (Algorithm 1).

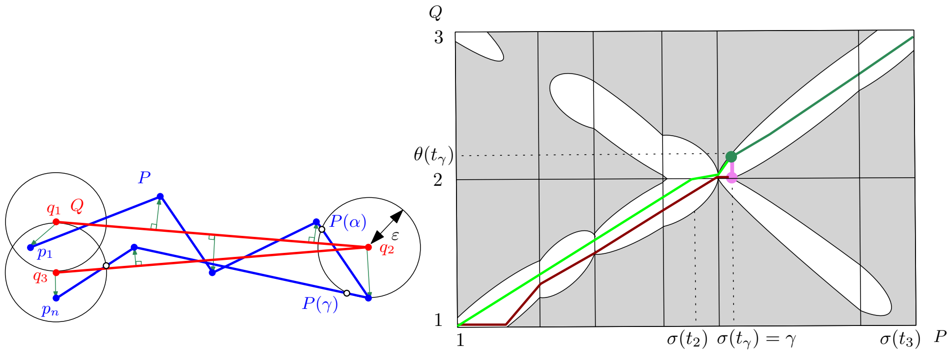


Fig. 6. Given an arbitrary matching (the concatenation of the light and dark green reachable paths), the orthogonal matching (the brown reachable path) between $P[1,\gamma]$ and $Q[1,2]$ exists. We construct a matching realizing $\delta_F(P[\gamma,n],Q[2,m]) \le \varepsilon$ as the concatenation of the pink and the dark green reachable paths.

Lemma 4 (Cutting Lemma). *Let $\varepsilon > 0$, and let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves such that $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{d})\varepsilon$. If $\delta_F(P,Q) \le \varepsilon$, then $P[1,\gamma]$ as the longest ε-prefix of P with respect to $Q[1,2]$ exists, $\delta_F(P[1,\gamma],Q[1,2]) \le \varepsilon$ and $\delta_F(P[\gamma,n],Q[2,m]) \le \varepsilon$.*

Proof. Let (σ,θ) be any Fréchet matching realizing $\delta_F(P,Q) \le \varepsilon$. This corresponds to a reachable path, which is shown as the concatenation of the light and dark green paths in the example in Fig. 6. Let $t_2 \in [0,1]$ be the largest value such that $Q(\theta(t_2)) = q_2$, hence $\delta_F(P[1,\sigma(t_2)],Q[1,2]) \le \varepsilon$. By Definition 2, γ exists with $\gamma \ge \sigma(t_2)$, and $\delta_F(P[1,\gamma],Q[1,2]) \le \varepsilon$. See the brown reachable path corresponding to the orthogonal matching realizing $\delta_F(P[1,\gamma],Q[1,2]) \le \varepsilon$ in Fig. 6. In the remainder of this proof we construct a matching to prove that $\delta_F(P[\gamma,n],Q[2,m]) \le \varepsilon$ (the concatenation of the pink and dark green paths).

Let $t_\gamma \in [0,1]$ be the largest value such that $P(\sigma(t_\gamma)) = P(\gamma)$. By Lemma 2, $P[\sigma(t_2),\gamma] \subseteq B(q_2,\sqrt{d}\varepsilon)$. Now let $t_3 \in [0,1]$ be the smallest value such that $Q(\theta(t_3)) = q_3$. We have $\|q_2 - q_3\| > (1+\sqrt{d})\varepsilon$, therefore $B(q_2,\sqrt{d}\varepsilon) \cap B(q_3,\varepsilon) = \emptyset$ and thus (σ,θ) cannot match q_3 to any point in $P[\sigma(t_2),\gamma]$. Therefore, $\sigma(t_2) \le \gamma = \sigma(t_\gamma) < \sigma(t_3)$, and correspondingly $\theta(t_2) \le \theta(t_\gamma) < \theta(t_3)$.

Now we construct a new matching $(\bar{\sigma},\bar{\theta})$ realizing $\delta_F(P[\gamma,n],Q[2,m]) \le \varepsilon$ as follows: $\bar{\sigma}(t) = \sigma(t)$ and $\bar{\theta}(t) = \theta(t)$ for all $t_\gamma \le t \le 1$ (dark green reachable path). On the other hand, since $\|P(\gamma) - q_2\| \le \varepsilon$ (pink point) and $\|P(\gamma) - Q(\theta(t_\gamma))\| \le \varepsilon$

(dark green point), we know that $Q[2, \theta(t_\gamma)] \subseteq B(P(\gamma), \varepsilon)$, i.e., the pink vertical segment is free. We set, $\bar{\sigma}(t) = \gamma$ and $\bar{\theta}(t) = \frac{t_\gamma - t}{t_\gamma} \cdot 2 + \frac{t}{t_\gamma} \cdot \theta(t_\gamma)$ for all $t_2 \leq t \leq t_\gamma$ (pink reachable path). Therefore, we have $\delta_F(P[\gamma, n], Q[2, m]) \leq \varepsilon$, which completes the proof. □

Now since by Lemma 4 we have $\delta_F(P[1, \gamma], Q[1, 2]) \leq \varepsilon$, Lemma 1 implies that the matching between $P[1, \gamma]$ and $Q[1, 2]$ is orthogonal. Let $P(x)$ be the last vertex of $P[1, \gamma]$ and let $Q(x')$ be its closest point on $Q[1, 2]$, for some $x < \gamma$ and $x' \leq 2$. Note that if $\|P(\gamma) - P(x)\|$ is shorter than 2ε, we can adjust the orthogonal matching by simply mapping all points on $P[x, \gamma]$ to $Q[x', 2]$. In addition, if P and Q have long edges then the free-space diagram is simpler than in the general case, since the entire vertical space (the pink segment in Fig. 6) between the two points $(\gamma, 2)$ and $(\gamma, \theta(t_\gamma))$ has to be free and cannot contain any blocked points.

3.3. *The Decision Algorithm*

In this section we present a linear time decision algorithm using the properties provided in Section 3.1 and Section 3.2. In Section 3.3.1 we consider the case that $l_P > 2\varepsilon$ and $l_Q > (1 + \sqrt{d})\varepsilon$. In Section 3.3.2 we show that this approach can be generalized to the case that $l_P \geq 2\varepsilon$ and $l_Q \geq (1 + \sqrt{d})\varepsilon$, and in Section 3.3.3 we generalize the approach to the case that there is only an edge length assumption on Q.

3.3.1. *Long Edges with $l_P > 2\varepsilon$ and $l_Q > (1 + \sqrt{d})\varepsilon$*

At the heart of our decision algorithm is the greedy algorithm presented in Algorithm 1. The input to this DecisionAlgorithm are two polygonal curves P and Q, and $\varepsilon > 0$. The algorithm assumes that P and Q have long edges. In each iteration the function LongestEpsilonPrefix returns γ, where $P[s, \gamma]$ is the longest ε-prefix of $P[s, n]$ with respect to $Q[i-1, i]$, if it exists. Here, s is the parameter where $P(s)$ is the endpoint of the previous longest ε-prefix with respect to $Q[i-2, i-1]$. At any time in the algorithm, if $\gamma = null$, this means that the corresponding longest ε-prefix does not exist and then "No" is returned. Otherwise, the next edge of Q is processed. This continues iteratively until all edges have been processed, or γ_i does not exist for some $i = 2, \cdots, m$.

The LongestEpsilonPrefix$(P[\gamma_{i-1}, n], Q[i-1, i], \varepsilon)$ procedure is implemented as follows: We use Alt and Godau's[7] dynamic programming algorithm to compute the reachability information in $\mathcal{FSD}_{\leq\varepsilon}(P[\gamma_{i-1}, n], Q[i-1, i])$, which computes all (s, t) for which $\delta_F(P[\gamma_{i-1}, s], Q[i-1, t]) \leq \varepsilon$. This takes linear time in the complexity of $P[\gamma_{i-1}, n]$ since $Q[i-1, i]$ is a single segment. Then γ_i is the largest s for which $\delta_F(P[\gamma_{i-1}, s], Q[i-1, i]) \leq \varepsilon$. Note that $P(s)$ has to lie on the boundary of $B(q_i, \varepsilon)$. If no such s exists then $\gamma_i = null$. We now prove the correctness of our decision algorithm.

Algorithm 1: Decide whether $\delta_F(P,Q) \le \varepsilon$

DecisionAlgorithm($P[1,n], Q[1,m], \varepsilon$)
    // Assumes $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{d})\varepsilon$
    $\gamma_1 \leftarrow 1$
    for $i \leftarrow 2$ **to** m **do**
        $\gamma_i \leftarrow$LongestEpsilonPrefix($P[\gamma_{i-1}, n], Q[i-1, i], \varepsilon$)
        if $\gamma = null$ **then return** *"No"*
        $s \leftarrow \gamma$
    if $\gamma < n$ **then return** *"No"*
    return *"Yes"*

Theorem 1 (Correctness). *Let $\varepsilon > 0$, and let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves such that $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{d})\varepsilon$. Then* DecisionAlgorithm*(P,Q,ε) returns "Yes" if and only if $\delta_F(P,Q) \le \varepsilon$.*

Proof. If the algorithm returns "Yes" then the sequence $\{(q_i,\gamma_i)\}$ for all $i = 1,\ldots,m$ with $\gamma_1 = 1$ and $\gamma_m = n$ describes a monotone matching that realizes $\delta_F(P,Q) \le \varepsilon$.

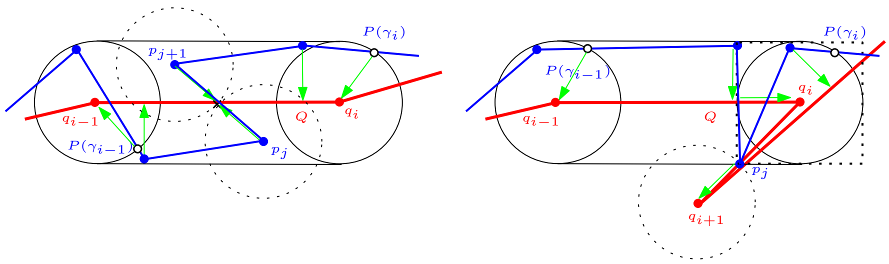
If $\delta_F(P,Q) \le \varepsilon$, then we prove by induction on i that the algorithm returns "Yes", i.e., all longest ε-prefixes $(P[1,\gamma_2], P[\gamma_2,\gamma_3], \ldots, P[\gamma_{m-1},\gamma_m])$ of P with respect to the corresponding segments of Q exist. For $i = 2$, following Lemma 4, γ_2 exists and can be found by the algorithm. For any $i > 2$, the algorithm has determined $\gamma_2,\ldots,\gamma_{i-1}$ already and by Lemma 4, $\delta_F(P[\gamma_{i-1},n], Q[i-1,m]) \le \varepsilon$. Another application of Lemma 4 yields that $\delta_F(P[\gamma_{i-1},\gamma_i], Q[i-1,i]) \le \varepsilon$ and $\delta_F(P[\gamma_i,n], Q[i,m]) \le \varepsilon$.

In the case that $i = m-1$ it remains to prove that $\gamma_{i+1} = \gamma_m = n$. For the sake of contradiction, assume $\gamma_m < n$. Since $P[\gamma_{m-1},\gamma_m]$ is the longest ε-prefix, there is no other $\gamma'_m \in (\gamma_m, n]$ such that $\delta_F(P[\gamma_{m-1},\gamma'_m], Q[m-1,m]) \le \varepsilon$. Consequently, $\delta_F(P[\gamma_{m-1},\gamma'_m], Q[m-1,m]) > \varepsilon$ and therefore $\delta_F(P[\gamma_{m-1},n], Q[m-1,m]) > \varepsilon$. Applying the contrapositive of Lemma 4 to $P[\gamma_{m-1},n]$ and $Q[m-1,m]$ yields $\delta_F(P,Q) > \varepsilon$, which is a contradiction. Therefore $\gamma_m = n$ and the algorithm returns "Yes" as claimed. □

Observation 2 (Piecewise Orthogonal Matching). *If $\delta_F(P,Q) \le \varepsilon$, then the sequence $\{\gamma_1,\gamma_2,\ldots,\gamma_n\}$ computed by Algorithm 1 induces a Fréchet matching that maps $P(\gamma_i)$ to q_i, and therefore $\delta_F(P,Q) \le \varepsilon$ for all $i = 2,\ldots,m$. Lemma 1 implies that the matching between $P[\gamma_{i-1},\gamma_i]$ and $Q[i-1,i]$ is orthogonal.*

We summarize this section with the following theorem:

Theorem 3 (Runtime). *Let $\varepsilon > 0$, and let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves such that $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{d})\varepsilon$. Then there exists*



(a) (b)

Fig. 7. Two examples of matchings between $P[\gamma_{i-1}, \gamma_i]$ and $Q[i-1, i]$ that are not (piecewise) orthogonal. Matchings are indicated with green arrows. All balls have radius ε. (a) An example where $l_P = \|p_{j+1} - p_j\| = 2\varepsilon$ and γ_i exists, but the induced Fréchet matching is not orthogonal. (b) An example where $\|q_{i+1} - q_i\| = (1 + \sqrt{d})\varepsilon$ and $B(q_{i+1}, \varepsilon) \cap B(q_i, \sqrt{d}\varepsilon) \cap P = p_j$. Although γ_i exists, a matching that is not piecewise orthogonal of width exactly ε exists.

a greedy decision algorithm, Algorithm 1, that can determine whether $\delta_F(P, Q) \leq \varepsilon$ in $O(n + m)$ time.

Proof. The number of vertices in $P[\gamma_{i-1}, \gamma_i]$ is at most $\lceil \gamma_i - \gamma_{i-1} \rceil + 1$. The algorithm greedily finds the longest ε-prefix per edge $Q[i-1, i]$ by calling LongestEpsilon-Prefix$(P[s, n], Q[i-1, i], \varepsilon)$ in $O(\lceil \gamma_i - \gamma_{i-1} \rceil + 1)$ time. The for-loop iterates over $m - 1$ edges, thus the runtime is $\sum_{i=2}^{m}(\lceil \gamma_i - \gamma_{i-1} \rceil + 1) < \sum_{i=2}^{m}(\gamma_i - \gamma_{i-1} + 2) = \gamma_m - \gamma_1 + 2(m-1) = n - 1 + 2m - 2 = O(n + m)$. □

3.3.2. *Long Edges with $l_P \geq 2\varepsilon$ and $l_Q \geq (1 + \sqrt{d})\varepsilon$*

We now consider the slightly more general case that $l_P \geq 2\varepsilon$ and $l_Q \geq (1 + \sqrt{d})\varepsilon$. The optimization algorithm presented in Section 4.1 makes use of this case. Clearly, if $l_P > 2\varepsilon$ and $l_Q > (1 + \sqrt{d})\varepsilon$ then Theorem 1 applies as usual. If $l_P = 2\varepsilon$ or $l_Q = (1 + \sqrt{d})\varepsilon$ then Algorithm 1 can still be run, however the Fréchet matching induced by the γ_i is not necessarily a piecewise orthogonal matching anymore, which means Observation 2 may not hold, see Fig. 7. However, we can still prove a slightly modified correctness theorem.

Theorem 4. *Let $\varepsilon > 0$, and $l_P \geq 2\varepsilon$ and $l_Q \geq (1 + \sqrt{d})\varepsilon$. If* DecisionAlgorithm$(P, Q, \varepsilon)$ *returns "Yes" then $\delta_F(P, Q) \leq \varepsilon$. If it returns "No" then $\delta_F(P, Q) \geq \varepsilon$.*

Proof. Let $\varepsilon^* = \delta_F(P, Q)$. If $l_P > 2\varepsilon$ and $l_Q > (1 + \sqrt{d})\varepsilon$ then Theorem 1 applies as usual. So, assume $l_P = 2\varepsilon$ or $l_Q = (1 + \sqrt{d})\varepsilon$. If the algorithm returns "Yes", then we know that $\delta_F(P[\gamma_{i-1}, \gamma_i], Q[i-1, i]) \leq \varepsilon_0$ for all $i = 2, \ldots, m$, and therefore $\varepsilon^* \leq \varepsilon$.

In the remainder of this proof we show the contrapositive of the second part:

If $\varepsilon^* = \delta_F(P,Q) < \varepsilon$ then DecisionAlgorithm(P,Q,ε) returns "Yes". So, assume $\varepsilon^* < \varepsilon$. Then, by Theorem 1, DecisionAlgorithm(P,Q,ε^*) returns "Yes", which means that all $\gamma_i^* =$ LongestEpsilonPrefix$(P[\gamma_{i-1}^*, n], Q[i-1,i], \varepsilon^*)$ exist for all $i = 2,\dots,m$, and $\gamma_1^* = 1$. We prove by induction that all $\gamma_i =$ LongestEpsilonPrefix$(P[\gamma_{i-1}, n], Q[i-1,i], \varepsilon)$ exist as well. The inductive base is trivial to show since $\gamma_1 = \gamma_1^* = 1$. Now as an inductive hypothesis let $i > 1$ be the largest integer value for which γ_{i-1} exists and is computed. In the following we show that $\gamma_i =$ LongestEpsilonPrefix$(P[\gamma_{i-1}, n], Q[i-1,i], \varepsilon)$ exists and can be computed. Let $P(x)$ be the first point along $P[\gamma_i^*, n]$ on the boundary of $B(q_i,\varepsilon)$. We have $\gamma_{i-1}^* < \gamma_{i-1} < \gamma_i^* < x$, where the first inequality follows from $B(q_{i-1},\varepsilon^*) \subset B(q_{i-1},\varepsilon)$, and the second inequality follows from $B(q_{i-1},\varepsilon) \cap B(q_i,\varepsilon^*) = \emptyset$ because $l_Q > 2\varepsilon$. Now let (σ,θ) be the Fréchet matching realizing $\delta_F(P[\gamma_{i-1}^*, \gamma_i^*], Q[i-1,i]) \leq \varepsilon^*$, and let $t \in [0,1]$ such that $\sigma(t) = \gamma_{i-1}$. Then from $\gamma_{i-1}^* < \gamma_{i-1} < \gamma_i^*$ follows that $i-1 \leq \theta(t) \leq i$. We can therefore construct a piecewise re-parameterization for $P[\gamma_{i-1}, x]$ and $Q[i-1,i]$ which yields:

$$\begin{aligned}\delta_F(P[\gamma_{i-1},x],Q[i-1,i]) \leq \max\{\ &\delta_F\big(P(\gamma_{i-1}),Q[i-1,\theta(t)]\big),\\ &\delta_F\big(P[\gamma_{i-1},\gamma_i^*],Q[\theta(t),i]\big),\ \delta_F\big(P[\gamma_i^*,x],q_i\big)\ \} \leq \varepsilon.\end{aligned}$$

Since $\gamma_i \geq x$, this implies that all γ_i exist for all $i = 2,\dots,m$. Note that the procedure LongestEpsilonPrefix$(P[\gamma_{i-1}, n], Q[i-1,i])$ can compute γ_i by finding the reachable path for (γ_i, i) across $\mathcal{FSD}_{\leq\varepsilon}(P[\gamma_{i-1}, n], Q[i-1,i])$. Therefore DecisionAlgorithm(P,Q,ε) returns "Yes". □

3.3.3. *Long Edges with $l_Q > 4\varepsilon$*

Our algorithm also can be applied to the case that one curve has arbitrary edge lengths and the other curve has edge lengths greater than 4ε.

Theorem 5 (Single Curve with Long Edges). *Let $\varepsilon > 0$, and let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves such that $l_P > 0$ and $l_Q > 4\varepsilon$. Then there exists a greedy decision algorithm, Algorithm 1, that can determine whether $\delta_F(P,Q) \leq \varepsilon$ or $\delta_F(P,Q) > \varepsilon$ in $O(n+m)$ time.*

Proof. In the proof of Lemma 4, we can replace Lemma 2 with Lemma 3, and realize that Lemma 4 also holds for the case $l_P > 0$ and $l_Q > 4\varepsilon$. The rest follows from Theorem 1 and Theorem 3. □

3.4. *Necessity of the Assumption*

As we have seen so far, Algorithm 1 greedily constructs a feasible Fréchet matching by linearly walking on curve P to find all longest ε-prefixes on it with respect to the corresponding edges of Q. Unfortunately, this property is not always true for curves with short edges. In general, there can be a quadratic number of blocked regions in the free space diagram of two curves; see Fig. 8 as an example of two curves in

$\mathbb{R}^2$ that have edges of length exactly equal 2ε except for some edges with lengths in $[2\varepsilon, (1+\sqrt{2})\varepsilon]$. This example demonstrates that our simple greedy construction of a Fréchet matching is unlikely to work if the edges are shorter than the assumptions we made. It also shows that our greedy construction does not work if both curves have edge lengths of at least 2ε.

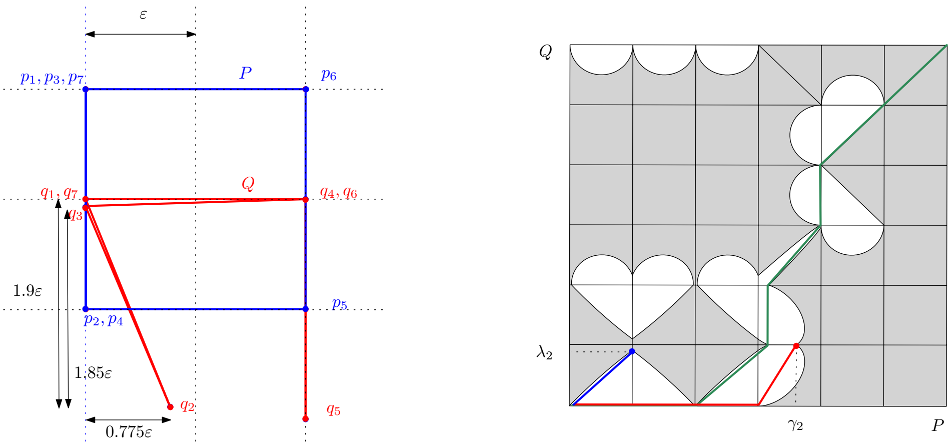


Fig. 8. An example in which the greedy algorithm fails to realize the Fréchet matching highlighted in green. Here, $P[1, \gamma_2]$ is the longest ε-prefix in P with respect to $Q[1, 2]$, as illustrated by the red reachable path. Also $Q[1, \lambda_2]$ is the longest ε-prefix in Q with respect to $P[1, 2]$ as illustrated by the blue reachable path. Every edge is 2ε long, except for the edges $Q[1, 2]$ and $Q[2, 3]$ that have lengths 2.02ε and 2.005ε, respectively. The latter values are still in the range $[2\varepsilon, (1+\sqrt{2})\varepsilon]$.

4. Optimization and Approximation

In this section, we present two algorithms for computing and approximating the Fréchet distance between two curves with long edges, respectively. First we give an exact algorithm which runs in $O((n+m)\log(n+m))$ time. Afterwards, we present a linear time algorithm which is similar to the greedy decision algorithm, but it uses the notion of minimum prefix to approximate the Fréchet distance.

4.1. *Optimization*

The main idea of our algorithm is that we compute critical values of the Fréchet distance between two curves and then perform binary search on these to find the optimal value acquired by the decision algorithm. In general, there are a cubic number of *critical values*, which are candidate values for the Fréchet distance between two polygonal curves. These critical values are those ε for which $p_1 \in B(q_1, \varepsilon)$ or $p_n \in B(q_m, \varepsilon)$, or when decreasing ε slightly a free space interval disappears on the boundary of a free space cell or a monotone path in the free space becomes non-monotone. See Alt and Godau[7] for more details on critical values. In our case we can show that it suffices to consider only a linear number of critical values, because the assumption on the edge lengths of the curves implies that a piecewise orthogonal matching exists, which reduces the number of possible critical values. Our optimization algorithm consists of the following four steps:

(1) Run DecisionAlgorithm(P, Q, ε_0) with $\varepsilon_0 = \min\{l_P/2, l_Q/(1+\sqrt{d})\}$ and store all γ_i = LongestEpsilonPrefix$(P[\gamma_{i-1}, n], Q[i-1, i], \varepsilon_0)$ for all $i = 2, \ldots, m$. Only proceed if DecisionAlgorithm(P, Q, ε_0) returns "Yes".
(2) If $P[\gamma_{i-1}, \gamma_i]$ is not $(Q[i-1, i], \varepsilon_0)$-monotone for some $i = 2, \ldots, m$ then return $\delta_F(P, Q) = \varepsilon_0$.
(3) Compute $C := \cup_{i=2}^{m} C_i \cup \{\varepsilon_0\}$, where C_i is the set of all critical values for $P[\alpha_{i-1}, \gamma_i]$ and $Q[i-1, i]$. Here, $P(\alpha_i)$ is the first point along $P[\gamma_{i-1}, n]$ that intersects $B(q_i, \varepsilon_0)$ and $\alpha_1 = 1$.
(4) Sort C and perform binary search on C using DecisionAlgorithm$(P, Q, \cdot)$ to find $\delta_F(P, Q)$.

In step (1) we set $\varepsilon_0 = \min\{l_P/2, l_Q/(1+\sqrt{d})\}$. This means that $l_P \geq 2\varepsilon_0$ and $l_Q \geq (1+\sqrt{d})\varepsilon_0$. Step (2) handles the case that the matching induced by the γ_i may not be a piecewise orthogonal matching. But once the algorithm proceeds to step (3), there exists a piecewise orthogonal matching between P and Q. This restricts the set of critical values we have to consider in step (3) as follows: Let $\varepsilon^* \leq \varepsilon_0$ and assume $\varepsilon^* = \delta_F(P, Q)$. Let γ_i^* = LongestEpsilonPrefix$(P[\gamma_{i-1}^*, n], Q[i-1, i], \varepsilon^*)$, for $i = 2, \ldots, m$, and let $P(\alpha_i^*)$ be the first intersection point between $P[\gamma_{i-1}^*, n]$ and $B(q_i, \varepsilon^*)$, and $\alpha_1^* = \gamma_1^* = 1$. From $B(q_i, \varepsilon^*) \subseteq B(q_i, \varepsilon_0)$ follows that $\alpha_i \leq \alpha_i^* \leq \gamma_i^* \leq \gamma_i$. And since $\gamma_{i-1} \leq \gamma_i$, we know that $P[\gamma_{i-1}^*, \gamma_i^*] \subseteq P[\alpha_{i-1}, \gamma_i]$. We thus have observed the following, see Fig. 9:

Observation 6. *Let $\varepsilon^* \leq \varepsilon_0$. For all $i = 2, \ldots, m$:*
(1) $\alpha_i \leq \alpha_i^* \leq \gamma_i^* \leq \gamma_i$, (2) $P[\gamma_{i-1}^*, \gamma_i^*] \subseteq P[\alpha_{i-1}, \gamma_i]$.

Therefore all critical values for $P[\gamma_{i-1}^*, \gamma_i^*]$ and $Q[i-1, i]$ must be contained in the set C_i which are the critical values for $P[\alpha_{i-1}, \gamma_i]$ and $Q[i-1, i]$, and the binary search in step (4) will identify ε^*.

Lemma 5 (Correctness). *Let $\varepsilon_0 = \min\{l_P/2, l_Q/(1+\sqrt{d})\}$ and let $\varepsilon^* = \delta_F(P, Q)$. If in step (1) of the optimization algorithm* DecisionAlgorithm(P, Q, ε_0) *returns "Yes", then the optimization algorithm returns ε^* and $\varepsilon^* \leq \varepsilon_0$. Otherwise $\varepsilon^* \geq \varepsilon_0$.*

Proof. If DecisionAlgorithm(P, Q, ε_0) returns "No" then Theorem 4 implies that $\delta_F(P, Q) = \varepsilon^* \geq \varepsilon_0$. Now suppose, for the remainder of this proof, that DecisionAlgorithm(P, Q, ε_0) returns "Yes". Then we know that all γ_i exist and $\delta_F(P[\gamma_{i-1}, \gamma_i], Q[i-1, i]) \leq \varepsilon_0$ for all $i = 2, \ldots, m$, and therefore $\varepsilon^* \leq \varepsilon_0$, see also

Theorem 4. This implies that ε_0 is an upper bound on all critical values in C. It remains to show that the optimization algorithm returns ε^*.

If in step (2) there is an $i = 2, \ldots, m$ such that $P[\gamma_{i-1}, \gamma_i]$ is not $(Q[i-1,i], \varepsilon)$-monotone, then there must exist an edge $P[j, j+1]$, for $\gamma_{i-1} \leq j < \gamma_i$, such that the angle between the direction vectors of $P[j, j+1]$ and $Q[i-1, i]$ is greater than $\pi/2$. The length of all edges in P must be at least $2\varepsilon_0$. But for this edge, the only way a (monotone) Fréchet matching between $P[\gamma_{i-1}, \gamma_i]$ and $Q[i-1, i]$ of width at most ε_0 can exist is if $\|p_{j+1} - p_j\| = 2\varepsilon_0$ and both p_j and p_{j+1} are matched to $x = B(p_j, \varepsilon_0) \cap B(p_{j+1}, \varepsilon_0) \cap Q[i-1, i]$. Therefore the width of such a Fréchet matching is exactly ε_0 and $\varepsilon^* = \varepsilon_0$.

It remains to show that if the algorithm passes step (2) it returns ε^* at the end of step (4). Since $\varepsilon_0 \in C$ and $\varepsilon^* \leq \varepsilon_0$, the binary search will return ε^* if $\varepsilon^* = \varepsilon_0$. So assume now that $\varepsilon^* < \varepsilon_0$. Since the algorithm passes step (2), it follows from Lemma 1 that the matching induced by the γ_i is indeed a piecewise orthogonal matching of width less than ε_0. From Observation 6 follows that all critical values for $P[\gamma^*_{i-1}, \gamma^*_i]$ and $Q[i-1, i]$ must be contained in the set C_i of all critical values for $P[\alpha_{i-1}, \gamma_i]$ and $Q[i-1, i]$. Thus, $\varepsilon^* \in C = \cup_{i=2}^m C_i$, and the binary search in step (4) returns ε^*. □

Computing The Critical Values: A piecewise orthogonal matching of width ε^* between P and Q is comprised of orthogonal matchings between $P[\gamma^*_{i-1}, \gamma^*_i]$ and $Q[i-1, i]$ for all $i = 2, \ldots, m$. The piecewise orthogonal matching may map vertices from P to $Q[i-1, i]$ either by an orthogonal projection or by mapping to the endpoints q_{i-1}, q_i. And vertices q_i may be mapped by on orthogonal projection to $P[\alpha_i, \gamma_i]$. These mappings define point-to-point distances that are candidates for ε^*, and thus critical values between $P[\gamma^*_{i-1}, \gamma^*_i]$ and $Q[i-1, i]$ that we need to optimize over. But since ε^* is not known beforehand, we compute the superset C_i of critical values between $P[\alpha_{i-1}, \gamma_i]$ and $Q[i-1, i]$ as follows: Let $\mathcal{H}_1$ be the hyperplane perpendicular to $Q[i-1, i]$ and tangent to $B(q_{i-1}, \varepsilon_0)$ that intersects $Q[i-1, i]$. Similarly, define $\mathcal{H}_2$ with respect to $B(q_i, \varepsilon_0)$. For each $p_j \in P[\alpha_{i-1}, \gamma_i]$: (1) If p_j lies between $\mathcal{H}_1$ and $\mathcal{H}_2$, then any orthogonal matching of width ε^* maps p_j to its orthogonal projection on $Q[i-1, i]$. We therefore add the distance $\|p_j, Q[i-1, i]\|$ to C_i. (2) If p_j lies on the q_{i-1}-side of $\mathcal{H}_1$, then an orthogonal matching of width ε^* can map p_j either to q_{i-1} or to its orthogonal projection on $Q[i-1, i]$. In this case we store both $\|p_j, Q[i-1, i]\|$ and $\|p_j - q_{i-1}\|$ in C_i. Similarly, if p_j lies on the q_i-side of $\mathcal{H}_2$ then we store $\|p_j, Q[i-1, i]\|$ and $\|p_j - q_i\|$ in C_i. Finally, for each edge e in $P[\alpha_i, \gamma_i]$: (3) we store $\|q_i, e\|$. See Fig. 9 for more illustration. We have the following theorem:

Theorem 7 (Optimization). *Let $P : [1, n] \to \mathbb{R}^d$ and $Q : [1, m] \to \mathbb{R}^d$ be two polygonal curves. If $\delta_F(P, Q) < \min\{l_P/2, l_Q/(1 + \sqrt{d})\}$, then $\delta_F(P, Q)$ can be computed in $O((n + m) \log(n + m))$ time.*

Proof. By Lemma 5 we know that the optimization algorithm returns $\delta_F(P, Q)$

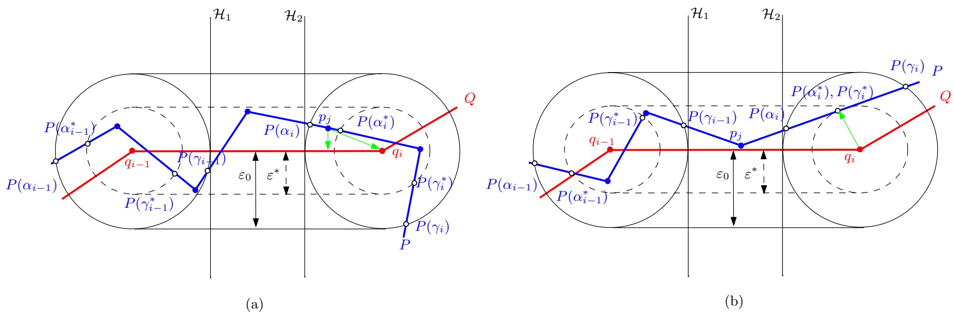


Fig. 9. Shown are two examples of orthogonal matchings between $P[\gamma^*_{i-1}, \gamma^*_i]$ and $Q[i-1, i]$, and the associated critical values (point-to-point distances defined by the matching). The cylinders $C(Q[i-1, i], \varepsilon_0)$ and $C(Q[i-1, i], \varepsilon^*)$ are shown, where $\varepsilon_0 \geq \varepsilon^*$. (a) p_j falls into case (2), when the orthogonal matching maps p_j either to q_i (if p_j lies inside $B(q_i, \varepsilon^*)$) or orthogonally to $Q[i-1, i]$ (if p_j lies outside $B(q_i, \varepsilon^*)$). (b) If an edge of P is tangent to $B(q_i, \varepsilon^*)$, then case (3) occurs. Here, the orthogonal matching has to map q_i to an edge e in $P[\alpha_i, \gamma_i]$.

correctly if $\delta_F(P, Q)$ is strictly less than $\min\{l_P/2, l_Q/(1+\sqrt{d})\}$. It only remains to prove the runtime of the algorithm. First we show that the number of critical values is linear. For each segment $Q[i-1, i]$, there are three cases for critical values contained in C_i: (1) There are at most $\lceil \alpha_i - \gamma_{i-1} \rceil + 1$ values if vertex p_j lies between $\mathcal{H}_1$ and $\mathcal{H}_2$. This is an upper bound for the number of vertices in $P[\gamma_{i-1}, \alpha_i]$. (2) There are at most $2(\lceil \gamma_{i-1} - \alpha_{i-1} \rceil + 1)$ values if vertex p_j lies on the q_{i-1}-side of $\mathcal{H}_1$, and similarly there are at most $2(\lceil \gamma_i - \alpha_i \rceil + 1)$ values if p_j lies on the q_i-side of $\mathcal{H}_2$. (3) There are at most $\lceil \gamma_i - \alpha_i \rceil$ critical values for each edge e in $P[\alpha_i, \gamma_i]$. Overall, the total is: $|C_i| \leq (\gamma_i - \alpha_i) + (\gamma_{i-1} - \alpha_{i-1}) + (\gamma_i - \alpha_{i-1}) + 11 < 2(\gamma_i - \alpha_{i-1}) + 11 = 2(\gamma_i - \gamma_{i-1}) + 2(\gamma_{i-1} - \alpha_{i-1}) + 11$. The latter inequality follows because $\alpha_{i-1} \leq \gamma_{i-1} \leq \alpha_i \leq \gamma_i$, see Fig. 9. Note that $\sum_{i=2}^{m} (\gamma_i - \gamma_{i-1}) = \gamma_m - \gamma_1 = n - 1$, and $\sum_{i=2}^{m} (\gamma_{i-1} - \alpha_{i-1}) < n$, therefore, $|C| = \sum_{i=2}^{m} |C_i| < 2(n-1) + 2n + 11(m-1) = O(n+m)$.

The optimization algorithm first runs Algorithm 1 in $O(n+m)$ time, then computes C in $O(n+m)$ time, and finally sorts C in $O((n+m)\log(n+m))$ time and performs binary search on C using the decision algorithm in $O((n+m)\log(n+m))$ time. Therefore, the total runtime is $O((n+m)\log(n+m))$. □

4.2. *Approximation Algorithm*

In this section we present a $\sqrt{d}$-approximation algorithm running in linear time. As a counterpart to the notion of longest ε-prefix we now introduce the notion of minimum prefix, which is the longest prefix of P with minimum Fréchet distance to a line segment e.

Definition 3 (Minimum Prefix). Let $P : [1, n] \to \mathbb{R}^d$ be a polygonal curve and $e : [1, 2] \to \mathbb{R}^d$ be a segment. Define $\gamma' = \max \operatorname{argmin}_{1 \leq t \leq n} \delta_F(P[1, t], e)$. We call $P[1, \gamma']$ the *minimum prefix of P with respect to e.*

Note that in the definition above, $P(\gamma')$ necessarily lies on the boundary of $B(e_2, \varepsilon')$, where $\varepsilon' = \min_{1 \leq t \leq n} \delta_F(P[1,t], e)$. The approximation algorithm is presented in Algorithm 2. First, for an initial threshold $\varepsilon_0 = \min\{l_P/(2\sqrt{d}), l_Q/(2d)\}$, it runs DecisionAlgorithm(P, Q, ε_0). The algorithm only continues if "Yes" gets returned. This ensures that P and Q have long edges, with $l_P \geq 2\sqrt{d}\varepsilon > 2\varepsilon$ and $l_Q \geq 2d\varepsilon > (1+\sqrt{d})\varepsilon$. Then, similar to the decision algorithm, the approximation algorithm greedily searches for longest ε-prefixes with respect to each segment of Q. However, it updates the current value of ε in each step, by computing the minimum prefix and its associated Fréchet distance to the portion of Q considered so far.

Algorithm 2: Approximate $\delta_F(P, Q)$

ApproximationAlgorithm$(P[1,n], Q[1,m])$
    $\varepsilon_0 \leftarrow \min\{l_P/2\sqrt{d}, l_Q/2d\}$
    if DecisionAlgorithm$(P, Q, \varepsilon_0) =$ *"No"* **then return** *"I don't know"*
    $(\gamma_2, \varepsilon_2) \leftarrow$MinimumPrefix$(P[1,n], Q[1,2])$
    $\varepsilon \leftarrow \varepsilon_2$
    $s \leftarrow \gamma_2$
    for $i \leftarrow 3$ **to** m **do**
        $(\gamma_i, \varepsilon_i) \leftarrow$MinimumPrefix$(P[s,n], Q[i-1,i])$
        $\varepsilon \leftarrow \max\{\varepsilon, \varepsilon_i\}$
        $s \leftarrow \gamma_i$
    if $\gamma_m = n$ **then**
        return ε
    else
        $\varepsilon \leftarrow \max\{\varepsilon, \delta_F(P[\gamma_m, n], q_m)\}$
        return ε

Now we are ready to prove the correctness of Algorithm 2:

Lemma 6 (The Approximation). *Let $P = P[1,n]$ and $Q = Q[1,m]$ be two polygonal curves and let $\varepsilon^* = \delta_F(P,Q)$. If $\varepsilon^* \leq \min\{l_P/2\sqrt{d}, l_Q/2d\}$ then* ApproximationAlgorithm*(P, Q) returns a value between ε^* and $\sqrt{d}\varepsilon^*$. Otherwise it returns "I don't know".*

Proof. From Algorithm 2 we have that $\varepsilon_i = \delta_F(P[\gamma_{i-1}, \gamma_i], Q[i-1,i])$. We prove by induction on i that $\varepsilon_i \leq \sqrt{d}\varepsilon^*$. For $i = 2$, ε_2 is being minimized and obviously $\varepsilon_2 \leq \varepsilon^* < \sqrt{d}\varepsilon^*$. For any $i > 2$, there are two possible cases: either $\varepsilon_i \leq \varepsilon^*$ or $\varepsilon_i > \varepsilon^*$. In the former case, trivially $\varepsilon_i < \sqrt{d}\varepsilon^*$. In the remainder of the proof we consider the latter case that is $\varepsilon_i > \varepsilon^*$. We know from Theorem 1 that all $\gamma_i^* =$ LongestEpsilonPrefix$(P[\gamma_{i-1}^*, n], Q[i-1,i], \varepsilon^*)$ for all $i = 1, 2, \ldots, m$ exist. And by inductive hypothesis we know that $\max\{\varepsilon_2, \ldots, \varepsilon_{i-1}\} \leq \sqrt{d}\varepsilon^*$.

We also know from line 8 of Algorithm 2 that $\varepsilon_i = \delta_F(P[\gamma_{i-1}, \gamma_i], Q[i-1, i])$ and $P[\gamma_{i-1}, \gamma_i]$ is the minimum prefix with respect to $Q[i-1, i]$. For the sake of contradiction we assume $\varepsilon_i > \sqrt{d}\varepsilon^* > \varepsilon^*$. We now distinguish two cases:

(a) If $\gamma_{i-1} < \gamma^*_{i-1}$, then by Lemma 2 we have $\delta_F(P[\gamma_{i-1}, \gamma^*_{i-1}], q_{i-1}) \leq \sqrt{d}\varepsilon^*$. Also $\delta_F(P[\gamma^*_{i-1}, \gamma^*_i], Q[i-1, i]) \leq \varepsilon^*$, hence $\delta_F(P[\gamma_{i-1}, \gamma^*_i], Q[i-1, i]) \leq \sqrt{d}\varepsilon^* < \varepsilon_i$. This contradicts the fact that $P[\gamma_{i-1}, \gamma_i]$ is the minimum prefix of $P[\gamma_{i-1}, n]$ with respect to $Q[i-1, i]$, see Fig. 10(a).

(b) Now for the case that $\gamma^*_{i-1} < \gamma_{i-1}$, consider the matching (σ, θ) realizing $\delta_F(P, Q) = \varepsilon^*$. There exists some $t \in [0, 1]$ such that $\gamma_{i-1} = \sigma(t)$. We can see that $Q(\theta(t)) \in Q[i-1, i]$ as follows: We know that $B(q_{i-1}, \varepsilon_{i-1}) \cap B(q_i, \varepsilon^*) = \emptyset$ since $\|q_{i-1} - q_i\| \geq 2d\varepsilon^* > (1 + \sqrt{d})\varepsilon^*$ and $\varepsilon_{i-1} \leq \sqrt{d}\varepsilon^*$. This implies $\gamma_{i-1} < \gamma^*_i$ and therefore $\gamma^*_{i-1} < \gamma_{i-1} < \gamma^*_i$, and correspondingly $i - 1 \leq \theta(t) \leq i$. By inductive hypothesis we know that $\varepsilon_{i-1} = \|q_{i-1} - P(\gamma_{i-1})\| \leq \sqrt{d}\varepsilon^*$, thus $Q[i-1, \theta(t)] \subseteq B(P(\gamma_{i-1}), \sqrt{d}\varepsilon^*)$ which implies $\delta_F(P(\gamma_{i-1}), Q[i-1, \theta(t)]) \leq \sqrt{d}\varepsilon^*$. Combining this with $\delta_F(P[\gamma_{i-1}, \gamma^*_i], Q[\theta(t), i]) \leq \varepsilon^*$ from the optimal matching yields $\delta_F(P[\gamma_{i-1}, \gamma^*_i], Q[i-1, i]) \leq \sqrt{d}\varepsilon^* < \varepsilon_i$. This contradicts that $P[\gamma_{i-1}, \gamma_i]$ is the minimum prefix of $P[\gamma_{i-1}, n]$ with respect to $Q[i-1, i]$, see Fig. 10(b).

In the end, if $\gamma_m < n = \gamma^*_m$, then Lemma 2 again implies $\delta_F(P[\gamma_m, n], q_m) \leq \sqrt{d}\varepsilon^*$ as claimed. The algorithm returns $\max\{\varepsilon_2, \ldots, \varepsilon_m\}$. Since there has to be some $\varepsilon_j > \varepsilon^*$, and we proved by induction that all $\varepsilon_i \leq \sqrt{d}\varepsilon^*$, the algorithm returns a value between ε^* and $\sqrt{d}\varepsilon^*$. □

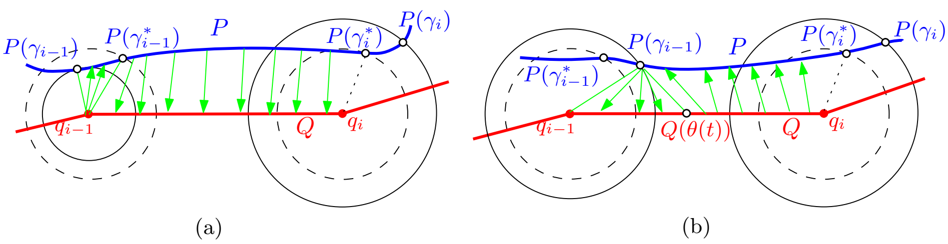


Fig. 10. Illustration for the proof of Lemma 6 when $\varepsilon_i > \varepsilon^*$. (a) $\gamma_{i-1} < \gamma^*_{i-1}$ (b) $\gamma_{i-1} > \gamma^*_{i-1}$.

The MinimumPrefix Procedure: Given a polygonal curve $P : [1, n] \to \mathbb{R}^d$ and a segment $e : [1, 2] \to \mathbb{R}^d$, we implement MinimumPrefix(P, e), as described in Algorithm 3, as follows: For every $i = 1, \ldots, n-1$, let c_i be the distance associated with a minimum prefix ending on the segment $P[i, i+1]$. Formally, $c_i = \min_{t \in [i, i+1]} \delta_F(P[1, t], e)$. Algorithm 3 computes all the c_i in a dynamic programming fashion. The minimum of the c_i is the desired ε, and the LongestEpsilonPrefix computes the corresponding γ.

Before we can prove the correctness of Algorithm 3, we need the following technical lemma that states when ε is increased, the longest ε-prefix can only get longer.

Algorithm 3: Compute $\textsc{MinimumPrefix}(P[1,n], e[1,2])$

$\textsc{MinimumPrefix}(P[1,n], e[1,2])$
- $c \leftarrow ||p_1 - e_1||$
- $\varepsilon' \leftarrow \min\{l_P/2, \|e\|/2\sqrt{d}\}$
- $\gamma' \leftarrow \textsc{LongestEpsilonPrefix}(P[1,n], e[1,2], \varepsilon')$
- **for** $i \leftarrow 1$ **to** $\lfloor \gamma' \rfloor$ **do**
  - $c_i \leftarrow \max\{c, ||e_2, P[i, i+1]||\}$
  - $c \leftarrow \max\{c, ||p_{i+1}, e[1,2]||\}$
- $\varepsilon = \min_{1 \le i \le \lfloor \gamma' \rfloor} c_i$
- **return** $(\varepsilon,\ \textsc{LongestEpsilonPrefix}(P[1,n], e[1,2], \varepsilon))$

Lemma 7 (Prefix monotonicity). *Let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves and $\varepsilon > \varepsilon' > 0$. Let $\gamma_1 = \gamma'_1 = 1$ and for all $i = 2, \ldots, m$ let $\gamma_i = \textsc{LongestEpsilonPrefix}(P[\gamma_{i-1}, n], Q[i-1, i], \varepsilon)$ and $\gamma'_i = \textsc{LongestEpsilonPrefix}(P[\gamma'_{i-1}, n], Q[i-1, i], \varepsilon')$. Then $\gamma_i < \gamma'_i$ for all $i = 2, \ldots, m$.*

Proof. The proof is by the induction. For $i = 2$, we know that $\delta_F(P[1, \gamma'_2], Q[1,2]) \le \varepsilon' < \varepsilon$. Let x be a parameter such that $P(x)$ is the first intersection point between $P[\gamma'_2, n]$ and the boundary of $B(q_2, \varepsilon)$, thus $\gamma'_2 < x$. Now observe that $\delta_F(P[\gamma'_2, x], q_2) \le \varepsilon$. Combining $\delta_F(P[1, \gamma'_2], Q[1,2]) < \varepsilon$ and $\delta_F(P[\gamma'_2, x], q_2) \le \varepsilon$ yields $\delta_F(P[1, x], Q[1,2]) \le \varepsilon$. Since γ_2 is the longest ε-prefix with respect to $Q[1,2]$, we have $\gamma'_2 < x \le \gamma_2$, and therefore $\gamma'_2 < \gamma_2$. Now for $i > 2$, by the inductive hypothesis we have $\gamma'_{i-1} < \gamma_{i-1}$. It remains to show $\gamma'_i < \gamma_i$. Consider a matching (σ, θ) realizing $\delta_F(P[\gamma'_{i-1}, \gamma'_i], Q[i-1, i]) \le \varepsilon'$. Let t be the value such that $\sigma(t) = \gamma_{i-1}$. Now we construct a new matching for $P[\gamma_{i-1}, x]$, where x is defined as in the inductive base, but with respect to $B(q_i, \varepsilon)$. We know that $\delta_F(P(\gamma_{i-1}), Q[i-1, \theta(t)]) \le \varepsilon$. Also we have $\delta_F(P[\gamma_{i-1}, \gamma'_i], Q[\theta(t), i]) \le \varepsilon' < \varepsilon$ by (σ, θ). Observe that $\delta_F(P[\gamma'_i, x], q_i) \le \varepsilon$. Thus, $\delta_F(P[\gamma_{i-1}, x], Q[i-1, i]) \le \varepsilon$ and using a similar argument as in the inductive base we have $\gamma'_i < x \le \gamma_i$, therefore $\gamma'_i < \gamma_i$. □

Now we are ready to prove the correctness of Algorithm 3:

Lemma 8 (Correctness). *Let $e : [1,2] \to \mathbb{R}^d$ be a line segment and let $P : [1, n'] \to \mathbb{R}^d$ be a polygonal curve monotone with respect to the line supporting e. The distance returned by $\textsc{MinimumPrefix}(P, e)$ is $\min_{1 \le t \le n'} \delta_F(P[1,t], e)$.*

Proof. According to the algorithm:

$$c_i = \max\{\|p_1 - e_1\|, \max_{1 \le j \le i-1} \|p_{j+1}, e\|, \|e_2, P[i, i+1]\|\}.$$

Since $e[1,2]$ is a segment and $P[1,n']$ is monotone with respect to the line supporting e, it follows from Lemma 1 that for any $i \le t \le i+1$ there exists an orthogonal matching such that:

$$\delta_F(P[1,t],e) = \max\{\|p_1 - e_1\|, \max_{1\le j\le i-1} \|p_{j+1}, e\|, \|P(t) - e_2\|\}$$

By taking the minimum on both sides, we get $\min_{i\le t\le i+1} \delta_F(P[1,t],e) = \max\{\|p_1 - e_1\|, \max_{1\le j\le i-1} \|p_{j+1}, e\|, \min_{i\le t\le i+1} \|P(t) - e_2\|\} = c_i$. It suffices to run the for-loop until $n' = \lfloor \gamma' \rfloor$, since by the assumption we only compute the minimum ε-prefix $P[1,\gamma]$ if its distance is at most ε' (line 3 of Algorithm 3), and by Lemma 7 it follows $\gamma < \gamma'$. Therefore, $\varepsilon = \min_{i=1}^{\lfloor\gamma'\rfloor} c_i = \min_{i=1}^{n'} c_i = \min_{1\le t\le n'} \delta_F(P[1,t],e)$. □

Theorem 8 (Runtime). *Let $P : [1,n] \to \mathbb{R}^d$ and $Q : [1,m] \to \mathbb{R}^d$ be two polygonal curves. If $\delta_F(P,Q) \le \min\{l_P/(2\sqrt{d}), l_Q/(2d)\}$, then Algorithm 2 approximates $\delta_F(P,Q)$ in $O(n+m)$ time within an approximation factor of $\sqrt{d}$.*

Proof. Let $\varepsilon^* = \delta_F(P,Q)$. The algorithm only proceeds past line 3 if $\varepsilon^* \le \varepsilon_0 = \min\{l_P/2\sqrt{d}, l_Q/2d\}$ and DecisionAlgorithm(P,Q,ε_0) returns "Yes". Now, let $\varepsilon' = \sqrt{d}\varepsilon_0$, $\gamma'_1 = 1$, and for all $i = 2,\ldots,m$ let $\gamma'_i =$ LongestEpsilonPrefix$(P[\gamma'_{i-1},n], Q[i-1,i], \varepsilon')$. Note that by definition of ε', both curves have long edges, i.e., $l_P \ge 2\sqrt{d}\varepsilon_0 > 2\varepsilon'$ and $l_Q \ge 2d\varepsilon_0 = 2\sqrt{d}\varepsilon' > (1+\sqrt{d})\varepsilon'$. From the proof of Lemma 6 we know that $\varepsilon_i \le \sqrt{d}\varepsilon^* \le \sqrt{d}\varepsilon_0 = \varepsilon'$ and since $\|q_{i-1} - q_i\| > 2\sqrt{d}\varepsilon'$, we have that $B(q_{i-1},\varepsilon') \cap B(q_i,\varepsilon_i) = \emptyset$. Therefore, $\gamma'_{i-1} < \gamma_i$. Lemma 7 implies that $\gamma_i \le \gamma'_i$ due to $\varepsilon_i \le \varepsilon'$, therefore $\gamma_{i-1} < \gamma'_{i-1} < \gamma_i < \gamma'_i$ for all $i = 2,\ldots,m$.

The for-loop in Algorithm 2 has $m-2$ iterations. In iteration i, the algorithm calls MinimumPrefix$(P[\gamma_{i-1},\gamma'_i], Q[i-1,i])$ in line 8. The for-loop in Algorithm 3 has $\lceil \gamma'_i - \gamma_{i-1} \rceil + 1$ iterations, where $\lceil \gamma'_i - \gamma_{i-1} \rceil + 1$ is the upper bound for the number of vertices in $P[\gamma_{i-1},\gamma'_i]$. Therefore, the runtime of Algorithm 2 is: $\sum_{i=2}^m (\lceil \gamma'_i - \gamma_{i-1} \rceil + 1) \le \sum_{i=2}^m (\gamma'_i - \gamma_{i-1} + 2) = \sum_{i=2}^m (\gamma'_i - \gamma_i) + \sum_{i=2}^m (\gamma_i - \gamma_{i-1}) + 2(m-1)$. Since $\gamma'_{i-1} \le \gamma_i$, we have $\sum_{i=2}^m (\gamma'_i - \gamma_i) \le \sum_{i=2}^m (\gamma'_i - \gamma'_{i-1})$. Thus, $\sum_{i=2}^m (\gamma'_i - \gamma_i) + \sum_{i=2}^m (\gamma_i - \gamma_{i-1}) + 2(m-1) \le \sum_{i=2}^m (\gamma'_i - \gamma'_{i-1}) + \sum_{i=2}^m (\gamma_i - \gamma_{i-1}) + 2(m-1) = \gamma'_m - \gamma'_1 + \gamma_m - \gamma_1 + 2(m-1) = 2(n-1) + 2(n-1) + 2(m-1) = O(n+m)$. □

5. Data Structure For Longest ε-Prefix Queries

In this section, we consider query variants of the setting in Section 3 for curves in the plane. We wish to solve the following problem: Preprocess a polygonal curve $P : [1,n] \to \mathbb{R}^2$ into a data structure such that for any polygonal query curve $Q : [1,m] \to \mathbb{R}^2$ and a positive $\varepsilon < \min\{l_P/2, l_Q/(1+\sqrt{2})\}$ one can efficiently decide whether $\delta_F(P,Q) \le \varepsilon$. Note that throughout this section we assume, as before, that P and Q have long edges, i.e., $l_P > 2\varepsilon$ and $l_Q > (1+\sqrt{2})\varepsilon$. Our query algorithm is identical to Algorithm 1. However, the key idea for speeding up the query algorithm is to efficiently compute LongestEpsilonPrefix($P[1, n], Q[1, 2], \varepsilon$) for a given query segment $Q[1, 2]$ in sublinear time. Our algorithm to compute the longest ε-prefix with respect to $Q[1, 2]$ is shown in Algorithm 4. According to Lemma 1 if $\delta(P[1, \gamma], Q[1, 2]) \leq \varepsilon$, then $P[1, \gamma]$ is $(Q[1, 2], \varepsilon)$-monotone. This is equivalent to computing the largest parameter $1 < t \leq n$ such that the following conditions hold: (1) $p_1 \in B(q_1, \varepsilon)$ and $P(t) \in B(q_2, \varepsilon)$, (2) $P[1, t] \subseteq C(Q[1, 2], \varepsilon)$, and (3) $P[1, t]$ is monotone with respect to line supporting $Q[1, 2]$. Note that the smallest value t that violates either of the conditions above is a potential γ.

Algorithm 4: Compute LongestEpsilonPrefix($P[1, n], Q[1, 2], \varepsilon$)

LongestEpsilonPrefix($P[1, n], Q[1, 2], \varepsilon$)
- **if** $p_1 \notin B(q_1, \varepsilon)$ **then return** *'null'*
- $\lambda \leftarrow$ LongestMonotonePrefix($P[1, n], Q[1, 2]$)
- $\alpha \leftarrow$ FirstIntersection($P[1, \lambda], B(q_2, \varepsilon)$)
- **if** $\alpha = null$ **then return** *'null'*
- $\beta \leftarrow$ LastIntersection($P[1, \lambda], B(q_2, \varepsilon)$)
- $r \leftarrow$ CylinderIntersection($P[1, \lambda], C(Q[1, 2], \varepsilon)$)
- **if** $r = null$ **then return** $\min(\lambda, \beta)$
- **if** $r < \alpha$ ***or*** $\lambda < \alpha$ **then return** *'null'*
- **if** $\alpha < r < \beta$ ***or*** $\alpha < \lambda < \beta$ **then return** $\min(r, \lambda)$
- **if** $r > \beta$ ***and*** $\lambda > \beta$ **then return** β

Here, LongestMonotonePrefix returns λ, where $P(\lambda)$ is the endpoint of the longest subcurve of $P[1, n]$ that starts in p_1 and is monotone with respect to the line supporting $Q[1, 2]$. FirstIntersection returns α, where $P(\alpha)$ is the first intersection point between $P[1, \lambda]$ and $B(q_2, \varepsilon)$. Similarly, LastIntersection returns β, where $P(\beta)$ is the last intersection point. CylinderIntersection finds r where $P(r)$ is the first point along P that intersects the boundary of $C(Q[1, 2], \varepsilon)$.

Computing LongestMonotonePrefix**:** We store all the edges of P in the leaves of a binary tree T ordered with respect to their indices. We call the subset of edges stored in the leaves of the subtree rooted at a node v the *canonical subset* of v. A set of nodes $v_1, \cdots, v_k$ in the subtree of v is called a set of *canonical nodes* of v if their leaves sets are disjoint and the union of their leaves sets is the leaves of the subtree of v. For each edge in P we consider its direction vector. Each internal node v stores the pair of the minimum/maximum angles between the direction vector and x-axis among all associated direction vectors stored in its canonical subset. Once given a query angle Φ and a starting point p_1, we retrieve $O(\log n)$ many leftmost (starting with p_1) canonical nodes of T whose leaves spans all edges in P that satisfy the monotonicity condition, i.e., condition (3) as mentioned earlier, with respect to Φ. This can simply be done by recursively searching children of a node v violating the monotonicity condition with respect to Φ. Once satisfying the condition, we already have $O(\log n)$ internal nodes to report their leaves as $P[1, \lambda]$. Searching children and reporting nodes take $O(\log n)$ time altogether using $O(n)$ space and $O(n \log n)$ preprocessing time.

Computing FIRSTINTERSECTION **and** LASTINTERSECTION**:** Let $\mathcal{H}$ be the hyperplane intersecting $Q[1, 2]$ that is perpendicular to $Q[1, 2]$ and is tangent to $B(q_2, \varepsilon)$. Let $\mathcal{H}'$ be the other hyperplane perpendicular to $Q[1, 2]$ and tangent to $B(q_2, \varepsilon)$. Since $P[1, \lambda]$ is monotone with respect to the line supporting $Q[1, 2]$, we know that λ must lie on the q_2-side of $\mathcal{H}$. And $P(\alpha) \in P[1, \lambda]$ must be located on the first edge intersecting $\mathcal{H}_2$. We start from p_1 and perform an exponential search on the edges of $P[1, \lambda]$ to find the first edge that intersects $\mathcal{H}$. Once the edge is found, we can find $P(\alpha)$ in constant time since each edge of P is longer than 2ε which is the diameter of $B(q_2, \varepsilon)$. Using the same method we can find $P(\beta) \in P[1, \lambda]$, if we consider $\mathcal{H}'$ instead of $\mathcal{H}$. If λ is on the q_2-side of $\mathcal{H}'$, we perform the exponential search on $P[1, \lambda]$ to find $P(\beta)$. If λ is on the q_1-side of $\mathcal{H}'$ then there is no $P(\beta) \in P[1, \lambda]$ and the algorithm does not require it. The whole process takes $O(\log n)$ time.

Computing CYLINDERINTERSECTION**:** Similar to Gudmundsson and Smid,[18] we construct a balanced binary search tree storing the points $p_1, p_2, \dots, p_n$ in its leaves (sorted by their indices). At each node of this tree, we store the convex hull of all points stored in its subtree. Given a query range $P[1, \lambda]$, we can retrieve $O(\log n)$ many canonical nodes of the tree containing convex hulls whose leaves span the whole range. For each convex hull we only need to compute extreme points with respect to the direction vector of the edge $Q[1, 2]$. If all extreme points lie inside $C(Q[1, 2], \varepsilon)$, then $r = null$, otherwise we consider the first extreme point $P(x)$ of some convex hull which lies outside $C(Q[1, 2], \varepsilon)$. Note that $P[1, x]$ crosses one of the two boundaries of $C(Q[1, 2], \varepsilon)$. Performing exponential search on $P[1, x]$ will find the first point that lies outside the respective boundary of $C(Q[1, 2], \varepsilon)$ for which $P(x)$ is obtained. This structure needs $O(n \log n)$ space and preproccessing time and answers queries in $O(\log^2 n)$ time. Plugging Algorithm 4 into the decision algorithm (Algorithm 1), we obtain the following theorem:

Theorem 9 (General Curves). *Let $P : [1, n] \to \mathbb{R}^2$ be a polygonal curve. A data structure of $O(n \log n)$ size can be built in $O(n \log n)$ time such that for any query curve $Q : [1, m] \to \mathbb{R}^2$ and a positive constant $\varepsilon < \min(l_P/2, l_Q/(1 + \sqrt{2}))$, it can be decided in $O(m \log^2 n)$ time whether $\delta_F(P, Q) \leq \varepsilon$.*

Proof. The correctness of the query algorithm follows from Theorem 1. As we mentioned, the space and preprocessing time of the whole data structure is $O(n \log n)$. Using Algorithm 4, the longest ε-prefix can be computed in $O(\log^2 n)$ time per segment, and hence the query algorithm runs in $O(m \log^2 n)$ time. □

When P is an x-monotone curve, we can handle queries in a slightly faster query time and also smaller space. In this case, we assume that ε is given at the preprocessing stage. The x-monotonicity of P allows us to use a different data structure for supporting the CylinderIntersection procedure, since the query time and space of this structure dominates the cost of our entire data structure.

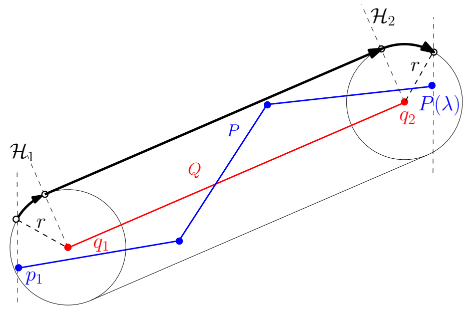


Fig. 11. Illustrating how combining the straight and circular line ray shooting queries can find the first point along $P[1, \lambda]$ that leaves $C(Q[1, 2], \varepsilon)$.

We implement the CylinderIntersection procedure by performing two types of ray shooting queries, straight and circular, along the boundary of $C(Q[1, 2], \varepsilon)$. It is easy to see that it suffices to perform at most two straight ray shooting queries and four circular ray shooting queries since P is x-monotone. See Fig. 11 for an illustration of the queries for the top part of the boundary of $C(Q[1, 2], \varepsilon)$.

For straight ray shooting queries we use the data structure by Hershberger and Suri.[19] Given a simple polygon, their structure returns the first point on the boundary of the polygon that is hit by a query ray ρ. It can be built in $O(n \log n)$ time using $O(n)$ space and answer queries in $O(\log n)$ time. However, to be able to use this structure we need to reduce our problem to ray shooting in a simple polygon. Let P_H be the (unbounded) polygon bounded from below by P, from the left by a vertical ray from p_1 to ∞, and from the right by a vertical ray from p_n to ∞. Similarly let P_L be the (unbounded) polygon bounded from above by P, from the left by a vertical ray from p_1 to $-\infty$, and from the right by a vertical ray from p_n to $-\infty$. We build one data structure for P_L and one for P_H. For circular ray shooting queries we use the data structure by Cheong et al.[20] Consider a simple polygon $\mathcal{P}$ with size n in the plane and let $r > 0$. For any circular query ray ρ with center o, radius r, and start point s', one can report in $O(\log n)$ query time the first point on the boundary of $\mathcal{P}$ which is hit by ρ. Combining these structures gives us the first point along $P[1, \lambda]$ that leaves the cylinder, which completes the implementation of CylinderIntersection. We have the following theorem:

Theorem 10 (x-Monotone Preprocessed Curve). *Let $\varepsilon > 0$ and let $P : [1, n] \to \mathbb{R}^2$ be an x-monotone polygonal curve in $\mathbb{R}^2$ such that $l_P > 2\varepsilon$. A lin-*

ear size data structure can be built in $O(n \log n)$ time such that for any polygonal query curve $Q : [1, m] \to \mathbb{R}^2$ with $l_Q > (1 + \sqrt{2})\varepsilon$, one can decide in $O(m \log n)$ time whether $\delta_F(P, Q) \leq \varepsilon$.

6. Discussion and Future Work

In this paper we provided a linear time decision algorithm, an $O((n+m)\log(n+m))$ time optimization algorithm, a linear time $\sqrt{d}$-approximation algorithm and a data structure with $O(m \log^2 n)$ query time for the Fréchet distance between curves that have long edges. Our algorithms are simple greedy algorithms that run in any constant dimension. In Section 3.4 we gave a critical example that justifies our assumptions on the edge lengths.

We proposed several greedy algorithms. Our assumption on the edge lengths allowed us to obtain a linear time constant-factor approximation algorithm for the (continuous) Fréchet distance. On the other hand, Bringmann and Mulzer[12] presented a greedy linear time exponential approximation algorithm for general curves under the discrete Fréchet distance. An interesting future research direction would be to develop a trade-off between the lengths of edges and the runtime, and in general prove hardness in terms of the edge lengths.

Acknowledgements. We thank the anonymous reviewers for helping to improve the presentation of this paper. We particularly thank one anonymous reviewer for insightful comments that helped us to improve the algorithm in Section 3.

References

1. J. Gudmundsson, M. Mirzanezhad, A. Mohades and C. Wenk, Fast Fréchet distance between curves with long edges, *arXiv e-prints* .
2. J. Gudmundsson, M. Mirzanezhad, A. Mohades and C. Wenk, Fast Fréchet distance between curves with long edges, in *Proceedings of the 3rd International Workshop on Interactive and Spatial Computing (IWISC)* (2018), pp. 52–58.
3. A. Efrat, L. Guibas, S. Har-Peled, J. Mitchell and T. Murali, New similarity measures between polylines with applications to morphing and polygon sweeping, *Discrete & Computational Geometry* **28** (2002) 535.
4. J. Gudmundsson, P. Laube and T. Wolle, Movement patterns in spatio-temporal data, in *Encyclopedia of GIS*, eds. S. Shekhar and H. Xiong (Springer-Verlag, 2007).
5. R. Sriraghavendra, K. Karthik and C. Bhattacharyya, Fréchet distance based approach for searching online handwritten documents, in *Proc. 9th International Conference on Document Analysis and Recognition (ICDAR)* (2007), pp. 461–465.
6. M. Jiang, Y. Xu and B. Zhu, Protein structure-structure alignment with discrete Fréchet distance, *Journal of Bioinformatics and Computational Biology* **6** (2008) 51.
7. H. Alt and M. Godau, Computing the Fréchet distance between two polygonal curves, *International Journal of Computational Geometry and Applications* **5** (1995) 75.
8. K. Buchin, M. Buchin, C. Knauer, G. Rote and C. Wenk., How difficult is it to walk the dog?, in *23rd European Workshop on Computational Geometry* (2007), pp. 170–173.
9. K. Bringmann, Why walking the dog takes time: Fréchet distance has no strongly sub-

quadratic algorithms unless SETH fails, in *Proceedings of the 55th IEEE Symposium on Foundations of Computer Science (FOCS)* (2014), pp. 661–670.
10. P. Agarwal, R. Avraham, H. Kaplan and M. Sharir, Computing the discrete Fréchet distance in subquadratic time, *SIAM Journal on Computing* **43** (2014) 429.
11. K. Buchin, M. Buchin, W. Meulemans and W. Mulzer, Four Soviets walk the dog - with an application to Alt's conjecture, *Discrete & Computational Geometry* **58** (2017) 180.
12. K. Bringmann and W. Mulzer, Approximability of the discrete Fréchet distance, *Journal of Computational Geometry* **7** (2016) 46.
13. H. Alt, C. Knauer and C. Wenk, Comparison of distance measures for planar curves, *Algorithmica* **38** (2004) 45.
14. B. Aronov, S. Har-Peled, C. Knauer, Y. Wang and C. Wenk., Fréchet distance for curves, revisited., in *14th Annual European Symposium on Algorithms* (2006), pp. 52–63.
15. A. Driemel, S. Har-Peled and C. Wenk, Approximating the Fréchet distance for realistic curves in near linear time, *Discrete & Computational Geometry* **48** (2012) 94.
16. North America Migration Flyways, `https://www.nps.gov/pais/learn/nature/birds.htm`, accessed: 2016-09-21.
17. J. Gudmundsson and M. Smid, Fast algorithms for approximate Fréchet matching queries in geometric trees, *Computational Geometry – Theory and Applications* **48** (2015) 479.
18. J. Gudmundsson and M. Smid, Fréchet queries in geometric trees, in *21st European Symposium on Algorithms (ESA)* (2013), pp. 565–576.
19. J. Hershberger and S. Suri, A pedestrian approach to ray shooting: Shoot a ray, take a walk, *Journal of Algorithms* **18** (1995) 403.
20. S.-W. Cheng, O. Cheong, H. Everett and R. van Oostrum, Hierarchical decompositions and circular ray shooting in simple polygons, *Discrete & Computational Geometry* **32** (2004) 401.