Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds
(extended abstract)

Karl Bringmann[*] Marvin Künnemann[†]

July 7, 2014

Abstract

The Fréchet distance is a well-studied and very popular measure of similarity of two curves. The best known algorithms have quadratic time complexity, which has recently been shown to be optimal assuming the Strong Exponential Time Hypothesis (SETH) [Bringmann FOCS'14].

To overcome the worst-case quadratic time barrier, restricted classes of curves have been studied that attempt to capture realistic input curves. The most popular such class are c-packed curves, for which the Fréchet distance has a $(1+\varepsilon)$-approximation in time $\tilde{\mathcal{O}}(cn/\varepsilon)$ [Driemel et al. DCG'12]. In dimension $d \geqslant 5$ this cannot be improved to $\mathcal{O}((cn/\sqrt{\varepsilon})^{1-\delta})$ for any $\delta > 0$ unless SETH fails [Bringmann FOCS'14].

In this paper, exploiting properties that prevent stronger lower bounds, we present an improved algorithm with runtime $\tilde{\mathcal{O}}(cn/\sqrt{\varepsilon})$. This is optimal in high dimensions apart from lower order factors unless SETH fails. Our main new ingredients are as follows: For filling the classical free-space diagram we project short subcurves onto a line, which yields one-dimensional separated curves with roughly the same pairwise distances between vertices. Then we tackle this special case in near-linear time by carefully extending a greedy algorithm for the Fréchet distance of one-dimensional separated curves.

[*] Max Planck Institute for Informatics, Campus E1 4, 66123 Saarbrücken, Germany; `kbringma@mpi-inf.mpg.de`.

Karl Bringmann is a recipient of the *Google Europe Fellowship in Randomized Algorithms*, and this research is supported in part by this Google Fellowship.

[†] Max Planck Institute for Informatics, Campus E1 4, 66123 Saarbrücken, Germany; `marvin@mpi-inf.mpg.de`. Saarbrücken Graduate School of Computer Science, Germany.

1 Introduction

The Fréchet distance is a very popular measure of similarity of two given curves and has two classic variants. Roughly speaking, the *continuous Fréchet distance* of two curves π, σ is the minimal length of a leash required to connect a dog to its owner, as they walk without backtracking along π and σ, respectively. In the *discrete Fréchet distance* we replace the dog and its owner by two frogs – in each time step each frog can jump to the next vertex along its curve or stay where it is.

In a seminal paper in 1991, Alt and Godau introduced the continuous Fréchet distance to computational geometry [4, 19]. For polygonal curves π and σ with n and m vertices[1], respectively, they presented an $\mathcal{O}(nm \log nm)$ algorithm. The discrete Fréchet distance was defined by Eiter and Mannila [18], who presented an $\mathcal{O}(nm)$ algorithm.

Since then, Fréchet distance has become a rich field of research: The literature contains generalizations to surfaces (see, e.g., [3]), approximation algorithms for realistic input curves ([5, 6, 17]), the geodesic and homotopic Fréchet distance (see, e.g., [12, 15]), and many more variants (see, e.g., [9, 16, 22, 24]). As a natural measure for curve similarity [2], the Fréchet distance has found applications in various areas such as signature verification (see, e.g., [25]), map-matching tracking data (see, e.g., [7]), and moving objects analysis (see, e.g., [10]).

Apart from log-factor improvements [1, 11] the quadratic complexity of the classic algorithms for the continuous and discrete Fréchet distance are still the state of the art. In fact, the first author recently showed a conditional lower bound: Assuming the Strong Exponential Time Hypothesis (SETH) there is no algorithm for the (continuous or discrete) Fréchet distance in time $\mathcal{O}((nm)^{1-\delta})$ for any $\delta > 0$, so apart from lower order terms of the form $n^{o(1)}$ the classic algorithms are optimal [8].

In attempts to obtain faster algorithms for realistic inputs, various restricted classes of curves have been considered, such as backbone curves [6], κ-bounded and κ-straight curves [5], and ϕ-low density curves [17]. The most popular model of realistic inputs are *c-packed curves*. A curve π is c-packed if for any point $z \in \mathbb{R}^d$ and any radius $r > 0$ the total length of π inside the ball $B(z, r)$ is at most cr, where $B(z, r)$ is the ball of radius r around z. This model has been used for several generalizations of the Fréchet distance, such as map matching [14], the mean curve problem [21], a variant of the Fréchet distance allowing shortcuts [16], and Fréchet matching queries in trees [20]. Driemel et al. [17] introduced c-packed curves and presented a $(1+\varepsilon)$-approximation for the continuous Fréchet distance in time $\mathcal{O}(cn/\varepsilon + cn \log n)$, which works in any $\mathbb{R}^d$, $d \geqslant 2$. Assuming SETH, the following lower bounds have been shown for c-packed curves: (1) For sufficiently small constant $\varepsilon > 0$ there is no $(1 + \varepsilon)$-approximation in time $\mathcal{O}((cn)^{1-\delta})$ for any $\delta > 0$ [8]. Thus, for constant ε the algorithm by Driemel et al. is optimal apart from lower order terms of the form $n^{o(1)}$. (2) In any dimension $d \geqslant 5$ and for varying $\varepsilon > 0$ there is no $(1 + \varepsilon)$-approximation in time $\mathcal{O}((cn/\sqrt{\varepsilon})^{1-\delta})$ for any $\delta > 0$ [8]. Note that this does not match the runtime of the algorithm by Driemel et al. for any $\varepsilon = n^{-b}$ and constant $b > 0$.

In this paper we improve upon the algorithm by Driemel et al. [17] by presenting an algorithm that matches the conditional lower bound of [8].

Theorem 1.1. *For any $0 < \varepsilon \leqslant 1$ we can compute a $(1 + \varepsilon)$-approximation on c-packed curves for the continuous and discrete Fréchet distance in time $\tilde{\mathcal{O}}(cn/\sqrt{\varepsilon})$.*

Specifically, our runtime is $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log(1/\varepsilon) + cn \log n)$ for the discrete variant and $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log^2(1/\varepsilon) + cn \log n)$ for the continuous variant.

We want to highlight that in general dimensions (specifically, $d \geqslant 5$) this runtime is *optimal* (apart from lower order terms of the form $n^{o(1)}$ unless SETH fails [8]). Moreover, we obtained

[1] We always assume that $m \leqslant n$.

our new algorithm by investigating why the conditional lower bound [8] cannot be improved and exploiting the discovered properties. Thus, the above theorem is the outcome of a synergetic effect of algorithms and lower bounds[2].

We remark that the same algorithm also yields improved runtime guarantees for other models of realistic input curves, like κ-bounded and κ-straight curves, where we are also able to essentially replace ε by $\sqrt{\varepsilon}$ in the runtime bound. In contrast to c-packed curves, it is not clear how far these bounds are from being optimal. We omit these results in this extended abstract.

Outline We give an improved algorithm that approximately decides whether the Fréchet distance of two given curves π, σ is at most δ. Using a construction of [16] to search over possible values of δ, this yields an improved approximation algorithm. We partition our curves into subcurves, each of which is either a *long segment*, i.e., a single segment of length at least $\Lambda = \Theta(\sqrt{\varepsilon}\delta)$, or a *piece*, i.e., a subcurve staying in the ball of radius Λ around its initial vertex. Now we run the usual algorithm that explores the reachable free-space (see Section 2 for definitions), however, we treat regions spanned by a piece π' of π and a piece σ' of σ in a special way. Typically, if π', σ' consist of n', m' segments then their free-space would be resolved in time $\mathcal{O}(n'm')$. Our overall speedup comes from reducing this runtime to $\tilde{\mathcal{O}}(n' + m')$, which is our first main contribution. To this end, we consider the line through the initial vertices of the pieces π', σ', and project π', σ' onto this line to obtain curves $\hat{\pi}, \hat{\sigma}$. Since π', σ' are *pieces*, i.e., they stay within distance $\Lambda = \Theta(\sqrt{\varepsilon}\delta)$ of their initial vertices, this projection does not change distances from π to σ significantly (it follows from the Pythagorean theorem that any distance of approximately δ is changed, by the projection, by less than $\varepsilon\delta$). Thus, we can replace π', σ' by $\hat{\pi}, \hat{\sigma}$ without introducing too much error. Note that $\hat{\pi}, \hat{\sigma}$ are *one-dimensional* curves; without loss of generality we can assume that they lie on $\mathbb{R}$. Moreover, we show how to ensure that $\hat{\pi}, \hat{\sigma}$ are *separated*, i.e., all vertices of $\hat{\pi}$ lie above 0 and all vertices of $\hat{\sigma}$ lie below 0. Hence, we reduced our problem to resolving the free-space region of one-dimensional separated curves.

It is known[3] that the Fréchet distance of one-dimensional separated curves can be computed in near-linear time, essentially since we can walk along π and σ with *greedy steps* to either find a feasible traversal or bottleneck subcurves. However, we face the additional difficulty that we have to resolve the *free-space region* of one-dimensional separated curves, i.e., given entry points on $\hat{\pi}$ and $\hat{\sigma}$, compute all exits on $\hat{\pi}$ and $\hat{\sigma}$. Our second main contribution is that we present an extension of the known result to handle this much more complex problem.

Organization We start with basic definitions and techniques borrowed from [16] in Section 2. In Section 3 we present our approximate decision procedure which reduces the problem to one-dimensional separated curves. We solve the latter in Section 4. In the whole paper, we focus on the continuous Fréchet distance. It is straightforward to obtain a similar algorithm for the discrete variant, in fact, then Section 4.1 becomes obsolete, which is why we save a factor of $\log 1/\varepsilon$ in the running time. All omitted proofs and illustrations are included in the full version in the appendix.

2 Preliminaries

For $z \in R^d$, $r > 0$ we let $B(z, r)$ be the ball of radius r around z. For $i, j \in \mathbb{N}$, $i \leqslant j$, we let $[i..j] := \{i, i{+}1, \ldots, j\}$, which is not to be confused with the real interval $[i, j] = \{x \in \mathbb{R} \mid i \leqslant x \leqslant j\}$.

[2] This yields one more reason why conditional lower bounds such as [8] should be studied, as they can show tractable cases and suggest properties that make these cases tractable.

[3] We thank Wolfgang Mulzer for pointing us to this result by Matias Korman and Sergio Cabello (personal communication). To the best of our knowledge this result is not published.

Throughout the paper we fix the dimension $d \geqslant 2$. A (polygonal) curve π is defined by its vertices $(\pi_1, \dots, \pi_n)$ with $\pi_p \in \mathbb{R}^d$, $p \in [1..n]$. We let $|\pi| = n$ be the number of vertices of π and $\|\pi\|$ be its total length $\sum_{i=1}^{n-1} \|p_i - p_{i+1}\|$. We write $\pi_{p..b}$ for the subcurve $(\pi_p, \pi_{p+1}, \dots, \pi_b)$. Similarly, for an interval $I = [p..b]$ we write $\pi_I = \pi_{p..b}$. We can also view π as a continuous function $\pi \colon [1, n] \to \mathbb{R}^d$ with $\pi_{p+\lambda} = (1-\lambda)\pi_p + \lambda\pi_{p+1}$ for $p \in [1..n-1]$ and $\lambda \in [0, 1]$. For the second curve $\sigma = (\sigma_1, \dots, \sigma_m)$ we will use indices of the form $\sigma_{q..d}$ for the reader's convenience.

Variants of the Fréchet distance Let Φ_n be the set of all continuous and non-decreasing functions ϕ from $[0, 1]$ onto $[1, n]$. The *continuous Fréchet distance* between two curves π, σ with n and m vertices, respectively, is defined as

$$d_{\mathrm{F}}(\pi, \sigma) := \inf_{\substack{\phi_1 \in \Phi_n \\ \phi_2 \in \Phi_m}} \max\nolimits_{t \in [0,1]} \|\pi_{\phi_1(t)} - \sigma_{\phi_2(t)}\|,$$

where $\|.\|$ denotes the Euclidean distance. We call $\phi := (\phi_1, \phi_2)$ a (continuous) *traversal* of (π, σ), and say that it has *width* $\max_{t \in [0,1]} \|\pi_{\phi_1(t)} - \sigma_{\phi_2(t)}\|$.

In the discrete case, we let Δ_n be the set of all non-decreasing functions ϕ from $[0, 1]$ onto $[1..n]$. We obtain the *discrete Fréchet distance* $d_{\mathrm{dF}}(\pi, \sigma)$ by replacing Φ_n and Φ_m by Δ_n and Δ_m. We obtain an analogous notion of a (discrete) *traversal* and its *width.* Note that any $\phi \in \Delta_n$ is a staircase function attaining all values in $[1..n]$. Hence, $(\phi_1(t), \phi_2(t))$ changes only at finitely many points in time t. At any such *time step*, we jump to the next vertex in π or σ or both.

Free-space diagram The discrete *free-space* of curves π, σ is defined as $\mathcal{D}^d_{\leqslant\delta}(\pi, \sigma) := \{(p, q) \in [1..n] \times [1..m] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$. Note that any discrete traversal of π, σ of width at most δ corresponds to a monotone sequence of points in the free-space where at each point in time we increase p or q or both. Because of this property, the free-space is a standard concept used in many algorithms for the Fréchet distance.

The continuous free-space is defined as $\mathcal{D}_{\leqslant\delta}(\pi, \sigma) := \{(p, q) \in [1, n] \times [1, m] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$. Again, a monotone path from $(1, 1)$ to (n, m) in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$ corresponds to a traversal of width at most δ. It is well-known [4, 19] that each *free-space cell* $C_{i,j} := \{(p, q) \in [i, i+1] \times [j, j+1] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$ (for $i \in [1..n-1], j \in [1..m-1]$) is convex, specifically it is the intersection of an ellipsoid with $[i, i+1] \times [j, j+1]$. In particular, the intersection of the free-space with any interval $[i, i+1] \times \{j\}$ (or $\{i\} \times [j, j+1]$) is an interval $I^h_{i,j}$ (or $I^v_{i,j}$), and for any such interval the subset that is reachable by a monotone path from $(1, 1)$ is an interval $R^h_{i,j}$ (or $R^v_{i,j}$). Moreover, in constant time one can solve the following *free-space cell problem*: Given intervals $R^h_{i,j} \subseteq [i, i+1] \times \{j\}, R^v_{i,j} \subseteq \{i\} \times [j, j+1]$, determine the intervals $R^h_{i,j+1} \subseteq [i, i+1] \times \{j+1\}, R^v_{i+1,j} \subseteq \{i+1\} \times [j, j+1]$ consisting of all points that are reachable from a point in $R^h_{i,j} \cup R^v_{i,j}$ by a monotone path within the free-space cell $C_{i,j}$. Solving this problem for all cells from lower left to upper right we determine whether (n, m) is reachable from $(1, 1)$ by a monotone path and thus decide whether the Fréchet distance is at most δ.

From approximate deciders to approximation algorithms An *approximate decider* is an algorithm that, given curves π, σ and $\delta > 0, 0 < \varepsilon \leqslant 1$, returns one of the outputs (1) $d_{\mathrm{F}}(\pi, \sigma) > \delta$ or (2) $d_{\mathrm{F}}(\pi, \sigma) \leqslant (1 + \varepsilon)\delta$. In any case, the returned answer has to be correct. In particular, if $\delta < d_{\mathrm{F}}(\pi, \sigma) \leqslant (1 + \varepsilon)\delta$ the algorithm may return either of the two outputs.

Let $D(\pi, \sigma, \delta, \varepsilon)$ be the runtime of an approximate decider and set $D(\pi, \sigma, \varepsilon) := \max_{\delta > 0} D(\pi, \sigma, \delta, \varepsilon)$. We assume polynomial dependence on ε, in particular, that there are constants $0 < c_1 < c_2 < 1$ such that for any $1 < \varepsilon \leqslant 1$ we have $c_1 D(\pi, \sigma, \varepsilon/2) \leqslant D(\pi, \sigma, \varepsilon) \leqslant c_2 D(\pi, \sigma, \varepsilon/2)$.

Driemel et al. [17] gave a construction of a $(1+\varepsilon)$-approximation for the Fréchet distance given an approximate decider. (This follows from [17, Theorem 3.15] after replacing their concrete approximate decider with runtime "$\mathcal{O}(N(\varepsilon,\pi,\sigma))$" by any approximate decider with runtime $D(\pi,\sigma,\varepsilon)$.)

Lemma 2.1. *Given an approximate decider with runtime $D(\pi,\sigma,\varepsilon)$ we can construct a $(1+\varepsilon)$-approximation for the Fréchet distance with runtime $\mathcal{O}\big(D(\pi,\sigma,\varepsilon)+D(\pi,\sigma,1)\log n\big)$.*

3 The approximate decider

Let π,σ be curves for which we want to (approximately) decide whether $d_{\mathrm{F}}(\pi,\sigma)>\delta$ or $d_{\mathrm{F}}(\pi,\sigma)\leqslant(1+\varepsilon)\delta$. We modify the curve π by introducing new vertices as follows. Start with the initial vertex π_1 as current vertex. If the segment following the current vertex has length at least $\Lambda=\Lambda_{\varepsilon,\delta}:=\min\{\frac{1}{2}\sqrt{\varepsilon},\frac{1}{4}\}\cdot\delta$ then mark this segment as *long* and set the next vertex as the current vertex. Otherwise follow π from the current vertex π_x to the first point π_y such that $\|\pi_x-\pi_y\|=\Lambda$ (or until we reach the last vertex of π). If π_y is not a vertex, but lies on some segment of π, then introduce a new vertex at π_y. Mark $\pi_{x..y}$ as a *piece* of π and set π_y as current vertex. Repeat until π is completely traversed. Since this procedure introduces at most $|\pi|$ new vertices and does not change the shape of π, with slight abuse of notation we call the resulting curve again π and set $n:=|\pi|$. This partitions π into subcurves $\pi^1,\dots,\pi^k$, with $\pi^s=\pi_{p_s..b_s}$, where every part π^s is either

- a *long segment*: $b_s=p_s+1$ and $\|\pi_{p_s}-\pi_{b_s}\|\geqslant\Lambda$, or
- a *piece*[4]: $\|\pi_{p_s}-\pi_{b_s}\|=\Lambda$ and $\|\pi_{p_s}-\pi_x\|<\Lambda$ for all $x\in[p_s,b_s)$.

Similarly, we introduce new vertices on σ and partition it into subcurves $\sigma^1,\dots,\sigma^\ell$, with $\sigma^t=\sigma_{q_t..d_t}$, each of which is a long segment or a piece. Let $m:=|\sigma|$.

We do not want to resolve each free-space cell on its own, as in the standard decision algorithm for the Fréchet distance. Instead, for any pair of pieces we want to consider the free-space region spanned by the two pieces at once. This is made formal by the following subproblem.

Problem 3.1 (Free-space region problem). Given $\delta>0$, $0<\varepsilon\leqslant1$, curves π,σ with n and m vertices, and *entry intervals* $\tilde{R}^h_{i,1}\subseteq[i,i+1]\times\{1\}$ for $i\in[1..n)$ and $\tilde{R}^v_{1,j}\subseteq\{1\}\times[j,j+1]$ for $j\in[1..m)$, compute *exit intervals* $\tilde{R}^h_{i,m}\subseteq[i,i+1]\times\{m\}$ for $i\in[1..n)$ and $\tilde{R}^v_{n,j}\subseteq\{n\}\times[j,j+1]$ for $j\in[1..m)$ such that (1) the exit intervals contain all points reachable from the entry intervals by a monotone path in $\mathcal{D}_{\leqslant\delta}(\pi,\sigma)$ and (2) all points in the exit intervals are reachable from the entry intervals by a monotone path in $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi,\sigma)$.

The standard solution to the free-space region problem would split it up into $n\cdot m$ free-space cells and resolve each cell in constant time, resulting in an $\mathcal{O}(n\cdot m)$ algorithm (this solves the problem even exactly, i.e., for $\varepsilon=0$). Restricted to pieces, we will show the following improvement, which will yield the desired overall speedup of a factor of $\sqrt{\varepsilon}$.

Lemma 3.2. *If π and σ are pieces then the free-space region problem can be solved in time $\mathcal{O}((n+m)\log^2 1/\varepsilon)$.*

We will prove this lemma in Sections 3.1 and 4.

Algorithm 3.3. Using an algorithm for the free-space region problem on pieces as in Lemma 3.2, we obtain an approximate decider for the Fréchet distance a follows. We create a directed graph which has a node $v_{s,t}$ for every region $[p_s,b_s]\times[q_t,d_t]$ spanned by pieces π^s and σ^t, and a node $u_{i,j}$ for every remaining region $[i,i+1]\times[j,j+1]$ (which is not contained in any region spanned by

[4] Note that the last piece actually might have distance $\|\pi_{p_s}-\pi_{b_s}\|$ less than Λ, however, for simplicity we assume equality for all pieces (in fact, a special handling of the last piece would only be necessary in Lemma 3.6).

two pieces), $i \in [1..n)$, $j \in [1..m)$. We add edges between two nodes whenever their regions touch (i.e., have a common interval I on their boundary), and direct this edge from the region that is to the left or below I to the other one. With each node $u_{i,j}$ we store the entry intervals $\tilde{R}^h_{i,j}$ and $\tilde{R}^v_{i,j}$, and with each node $v_{s,t}$ we store the entry intervals $\tilde{R}^h_{i,q_t} \subseteq [i, i+1] \times \{q_t\}$ for $i \in [p_s..b_s)$ and $\tilde{R}^v_{p_s,j} \subseteq \{p_s\} \times [j, j+1]$ for $j \in [q_t..d_t)$. After correctly initializing the outer reachability intervals $\tilde{R}^h_{i,1}$ and $\tilde{R}^v_{1,j}$, we follow any topological ordering of this graph. For any node $u_{i,j}$, we resolve its region by solving the corresponding free-space *cell* problem in constant time. For any node $v_{s,t}$, we solve the corresponding free-space *region* problem on $\pi' = \pi^s, \sigma' = \sigma^t$ (and $\delta' = \delta, \varepsilon' = \varepsilon$) using Lemma 3.2. Finally, we return $d_F(\pi, \sigma) \leqslant (1+\varepsilon)\delta$ if $(n, m) \in \tilde{R}^h_{n-1,m}$ and $d_F(\pi, \sigma) > \delta$ otherwise.

Lemma 3.4. *Algorithm 3.3 is a correct approximate decider.*

In the above algorithm we can ignore *unreachable* nodes, i.e., nodes where all stored entry intervals would be empty. To this end, we fix a topological ordering by mapping a node corresponding to a region $[x_1, x_2] \times [y_1, y_2]$ to $x_2 + y_2$ and sorting by this value ascendingly. This yields $n + m$ layers of nodes, where the order within each layer is arbitrary. For each layer we build a dictionary data structure (a hash table), in which we store only the reachable nodes of this layer. This allows to quickly enumerate all reachable nodes of a layer. The total overhead for managing the $n + m$ dictionaries is $\mathcal{O}(n + m)$.

Let us analyze the runtime of the obtained approximate decider. Let S be the set of non-empty free-space cells $C_{i,j}$ of $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi, \sigma)$ such that i or j is not contained in a piece. Moreover, let T be the set of all pairs (s, t) such that π^s, σ^t are pieces with initial vertices within distance $(1+\varepsilon)\delta + 2\Lambda$. Define $N(\pi, \sigma, \delta, \varepsilon) := |S| + \sum_{(s,t)\in T}(|\pi^s| + |\sigma^t|)$ and set $N(\pi, \sigma, \varepsilon) := \max_{\delta>0} N(\pi, \sigma, \delta, \varepsilon)$. Since the algorithm considers only *reachable* cells and any reachable cell is also non-empty, the cost over all free-space cell problems solved by our approximate decider is bounded by $\mathcal{O}(|S|)$. Since every reachable (thus non-empty) region spanned by two pieces has initial points within distance $(1+\varepsilon)\delta + 2\Lambda$, the second term bounds the cost over all free-space region problems on pieces (apart from the $\log^2 1/\varepsilon$ factor). Hence, we obtain the following.

Lemma 3.5. *The approximate decider has runtime $D(\pi, \sigma, \varepsilon) = \mathcal{O}(N(\pi, \sigma, \varepsilon) \cdot \log^2 1/\varepsilon)$.*

Using similar arguments as [16, Lemma 4.4], we can analyze the free-space complexity of c-packed curves to obtain the following result. Plugging this into Lemmas 3.5 and 2.1, we obtain an approximation algorithm for the Fréchet distance with running time $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log^2 1/\varepsilon + cn \log n) = \tilde{\mathcal{O}}(\frac{cn}{\sqrt{\varepsilon}})$, as desired.

Lemma 3.6. *Let π, σ be c-packed curves with n vertices in total. Then $N(\pi, \sigma, \varepsilon) = \mathcal{O}(cn/\sqrt{\varepsilon})$.*

3.1 Solving the free-space region problem on pieces

It remains to prove Lemma 3.2. Let $(\pi, \sigma, \delta, \varepsilon)$ be an instance of the free-space region problem, where $n := |\pi|$, $m := |\sigma|$, with $\|\pi_1 - \pi_x\|, \|\sigma_1 - \sigma_y\| \leqslant \Lambda_{\varepsilon,\delta} = \Lambda$ for any $x \in [1, n]$, $y \in [1, m]$ (and entry intervals $\tilde{R}^h_{i,1} \subseteq [i, i+1] \times \{1\}$ for $i \in [1..n)$ and $\tilde{R}^v_{1,j} \subseteq \{1\} \times [j, j+1]$ for $j \in [1..m)$). We reduce this instance to the free-space region problem on *one-dimensional separated* curves, i.e., curves $\hat{\pi}, \hat{\sigma}$ in $\mathbb{R}$ such that all vertices of $\hat{\pi}$ lie above 0 and all vertices of $\hat{\sigma}$ lie below 0.

Since π and σ stay within distance Λ of their initial vertices, if their initial vertices are within distance $\|\pi_1 - \sigma_1\| \leqslant \delta - 2\Lambda$ then all pairs of points in π, σ are within distance δ. In this case, we find a translation of π making $\|\pi_1 - \sigma_1\| = \delta - 2\Lambda$ and all pairwise distances are still at most δ. This ensures that the curves π, σ are contained in disjoint balls of radius $\Lambda \leqslant \frac{1}{4}\delta$ centered at their initial vertices.

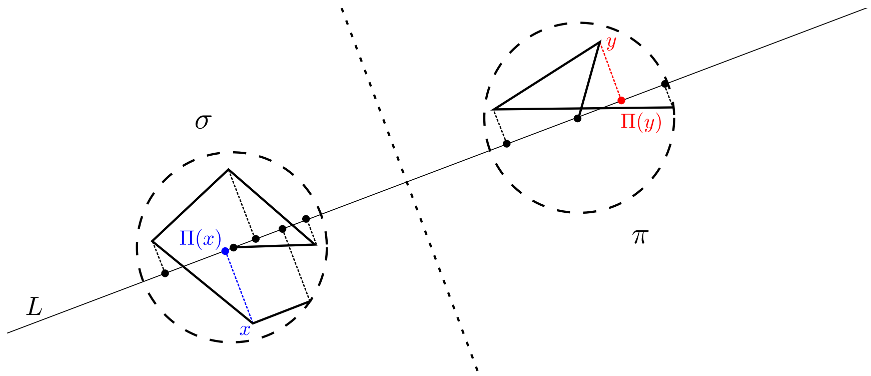


Figure 1: Projection of the pieces π, σ onto the line L through their initial vertices. This yields one-dimensional separated curves $\hat{\pi}, \hat{\sigma}$.

Consider the line L through the initial vertices π_1 and σ_1. Denote by $\Pi\colon \mathbb{R}^d \to L$ the projection onto L. Now, instead of the pieces π, σ we consider their projections $\hat{\pi} := \Pi(\pi) = (\Pi(\pi_1), \dots, \Pi(\pi_n))$ and $\hat{\sigma} := \Pi(\sigma) = (\Pi(\sigma_1), \dots, \Pi(\sigma_m))$, see Figure 1. Note that after rotation and translation we can assume that $\hat{\pi}$ and $\hat{\sigma}$ lie on $\mathbb{R} \subset \mathbb{R}^d$ and $\hat{\pi}$ and $\hat{\sigma}$ are separated by $0 \in \mathbb{R}$ (since π and σ are contained in disjoined balls centered on L). Now we solve the free-space region problem on $\hat{\pi}$, $\hat{\sigma}$, $\hat{\delta} := \delta$, and $\hat{\varepsilon} := \frac{1}{2}\varepsilon$ (with the same entry intervals $\tilde{R}^h_{i,j}, \tilde{R}^v_{i,j}$)). The Pythagorean theorem shows that the projection preserves pairwise distances sufficiently well.

Lemma 3.7. *Any solution to the the free-space region problem on $(\hat{\pi}, \hat{\sigma}, \hat{\delta}, \hat{\varepsilon})$ solves the free-space region problem on $(\pi, \sigma, \delta, \varepsilon)$.*

We will show the following lemma in Section 4, concluding the proof of Lemma 3.2.

Lemma 3.8. *The free-space region problem on one-dimensional separated curves can be solved in time $\mathcal{O}((n+m)\log^2 1/\varepsilon)$.*

4 On one-dimensional separated curves

In this section, we show how to solve the free-space region problem on one-dimensional separated curves in time $\mathcal{O}((n+m)\log^2 1/\varepsilon)$, i.e., we prove Lemma 3.8. First we reduce it to the following.

Problem 4.1 (Reduced free-space problem). Given one-dimensional separated curves π, σ with n, m vertices and all vertices being multiples of $\frac{1}{3}\varepsilon\delta$, and given an entry set $E \subseteq [1..n]$, compute the exit set $F^\pi \subseteq [1..n]$ consisting of all points f such that $d_{\mathrm{dF}}(\pi_{e..f}, \sigma) \leqslant \delta$ for some $e \in E$ and the exit set $F^\sigma \subseteq [1..m]$ consisting of all points f such that $d_{\mathrm{dF}}(\pi_{e..n}, \sigma_{1..f}) \leqslant \delta$ for some $e \in E$.

Lemma 4.2. *The reduced free-space problem can be solved in time $\mathcal{O}((n+m)\log 1/\varepsilon)$.*

As a second step, we prove the above lemma. We first consider the special case of $E = \{1\}$ and the problem of deciding whether $n \in F^\pi$, i.e., the lower left corner $(1,1)$ of the free-space is the only entry point and we want to determine whether the upper right corner (n,m) is an exit. This is equivalent to deciding whether the discrete Fréchet distance of π, σ is at most δ. We present a greedy algorithm for this special case in Section 4.2. To extend this to the reduced free-space problem, we prove useful structural properties of one-dimensional separated curves in Section 4.3. With these, we first solve the problem of determining the exit set F^π assuming $E = \{1\}$ in Section 4.4.1. This can be extended to compute, for general $E \subseteq [1..n]$, F^π (Section 4.4.2) and F^σ (omitted in this extended abstract; this case is more complex than F^π but solved by similar ideas).

4.1 Reduction from the continuous to the discrete case

Essentially we use the following lemma to reduce the *continuous* free-space region problem on one-dimensional separated curves to the *discrete* reduced free-space problem.

Lemma 4.3. *Let π, σ be one-dimensional separated curves with subcurves $\pi_{p..b}, \sigma_{q..d}$. Then we have $d_{\mathrm{F}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{dF}}(\pi_{p..b}, \sigma_{q..d})$. In particular, assume that we subdivide any segments of π, σ by adding new vertices, which yields new curves π', σ' with subcurves $\pi'_{p'..b'}, \sigma'_{q'..d'}$ that are subdivisions of $\pi_{p..b}, \sigma_{q..d}$. Then we have $d_{\mathrm{dF}}(\pi'_{p'..b'}, \sigma'_{q'..d'}) = d_{\mathrm{dF}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{F}}(\pi_{p..b}, \sigma_{q..d})$.*

The above lemma allows the following trick. Consider any finite sets $E \subseteq [1, n]$ and $F \subseteq [1, n]$. Add π_x as a vertex to π for any $x \in E \cup F$, with slight abuse of notation we say that π now has vertices at π_i, $i \in [1..n]$, and π_x, $x \in E \cup F$. Mark the vertices π_x, $x \in E$, as entries. Now solve the reduced free-space problem instance (π, σ, E). This yields the set F^π of all values $f \in F$ such that there is an $e \in E$ with $d_{\mathrm{dF}}(\pi_{e..f}, \sigma) \leqslant \delta$, which by Lemma 4.3 is equivalent to $d_{\mathrm{F}}(\pi_{e..f}, \sigma) \leqslant \delta$. Thus, we computed all exit points in F given entry points in E, with respect to the continuous Fréchet distance. This is already near to a solution of the free-space region problem, however, we have to cope with entry and exit *intervals*.

For the full reduction we need two more arguments. First, we can replace all non-empty input intervals $\tilde{R}^h_{i,1}$ by the leftmost point $(y_i, 1)$ in $\tilde{R}^h_{i,1} \cap \mathcal{D}_{\leqslant\delta}(\pi, \sigma)$, specifically, we show that any traversal starting in a point in $\tilde{R}^h_{i,1}$ can be transformed into a traversal starting in $(y_i, 1)$. Thus, we add π_{y_i} as a vertex and mark it as an entry to obtain a finite and small set of entry points. Second, for any segment $\pi_{i..i+1}$ we call a point $f \in [i, i+1]$ *reachable* if there is an $e \in E$ with $d_{\mathrm{F}}(\pi_{e..f}, \sigma) \leqslant \delta$. We show that if f is reachable then essentially all points $f' \in [i, i+1]$ with $\pi_{f'} \leqslant \pi_f$ are also reachable. Thus, the set of reachable points is an interval with one trivial endpoint, and we only need to search for the other endpoint of the interval, which can be done by binary search. Moreover, we can parallelize all these binary searches, as solving one reduced free-space problem can answer for every segment of π whether a particular point on this segment is reachable (after adding this point as a vertex). To make these binary searches finite, we round all vertices of π and σ to multiples of $\gamma := \frac{1}{3}\varepsilon\delta$ and only search for exit points that are multiples of γ. This is allowed since the free-space region problem only asks for an approximate answer. A similar procedure yields the exits on σ reachable from entries on π, and determining the exits reachable from entries on σ is a symmetric problem. Since for the binary searches we reduce to $\mathcal{O}(\log 1/\varepsilon)$ instances of the reduced free-space problem, Lemma 3.8 follows from Lemma 4.2.

4.2 Greedy Decider for the Fréchet Distance of One-Dimensional Separated Curves

In the remainder of the paper all indices of curves will be integral. Let $\pi = (\pi_1, \ldots, \pi_n)$ and $\sigma = (\sigma_1, \ldots, \sigma_m)$ be two separated polygonal curves in $\mathbb{R}$, i.e., $\pi_i \geqslant 0 \geqslant \sigma_j$. For indices $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$, define $\mathrm{vis}_\sigma(i, j) := \{k \mid k \geqslant j \text{ and } \sigma_k \geqslant \pi_i - \delta\}$ as the index set of vertices on σ that are later in sequence than σ_j and are still in distance δ to π_i (i.e, *seen* by π_i) and, likewise, $\mathrm{vis}_\pi(i, j) := \{k \mid k \geqslant i \text{ and } \pi_k \leqslant \sigma_j + \delta\}$. Hence, the set of points that we may reach on σ by starting in (π_i, σ_j) and staying in π_i can be defined as the longest contiguous subsequence $[j+1..j+k]$ such that $[j+1..j+k] \subseteq \mathrm{vis}_\sigma(i, j)$. Let $\mathrm{reach}_\sigma(i, j) := [j+1..j+k]$ denote this subsequence and let $\mathrm{reach}_\pi(i, j)$ be defined symmetrically. Note that $\pi_i \leqslant \pi_{i'}$ implies that $\mathrm{vis}_\sigma(i, j) \supseteq \mathrm{vis}_\sigma(i', j)$, however the converse does not necessarily hold. Also, $\mathrm{vis}_\sigma(i, j) \not\supseteq \mathrm{vis}_\sigma(i', j)$ implies that $\mathrm{vis}_\sigma(i, j) \subsetneq \mathrm{vis}_\sigma(i', j)$ and $\pi_i > \pi_{i'}$.

The visibility sets established above enable us to define a greedy algorithm for the Fréchet distance of π and σ. Let $1 \leqslant p \leqslant n$ and $1 \leqslant q \leqslant m$ be arbitrary indices on σ and π. We say that p'

is a *greedy step on π from (p,q)*, written $p' \leftarrow \textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$, if $p' \in \mathrm{reach}_\pi(p,q)$ and $\mathrm{vis}_\sigma(i,q) \subseteq \mathrm{vis}_\sigma(p',q)$ holds for all $p \leqslant i \leqslant p'$. Symmetrically, $q' \in \mathrm{reach}_\sigma(p,q)$ is a *greedy step on σ from (p,q)*, if $\mathrm{vis}_\pi(p,i) \subseteq \mathrm{vis}_\pi(p,q')$ for all $q \leqslant i \leqslant q'$. In pseudo code, $\textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$ denotes a function that returns an arbitrary greedy step p' on π from (p,q) if such an index exists and returns an error otherwise (symmetrically for σ).

Consider the following greedy algorithm:

Algorithm 1 Greedy algorithm for the Fréchet distance of separated curves $\pi_{1..n}$ and $\sigma_{1..m}$ in $\mathbb{R}$

```
$p \leftarrow 1, q \leftarrow 1$
**repeat**
    **if** $p' \leftarrow \textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$ **then** $p \leftarrow p'$
    **if** $q' \leftarrow \textsc{GreedyStep}_\sigma(\pi_{p..n}, \sigma_{q..m})$ **then** $q \leftarrow q'$
**until** no greedy step was found in the last iteration
**if** $p = n$ and $q = m$ **then return** $d_{\mathrm{dF}}(\pi, \sigma) \leqslant \delta$
**else return** $d_{\mathrm{dF}}(\pi, \sigma) > \delta$
```

Theorem 4.4. *Let π and σ be separated curves in $\mathbb{R}$ and $\delta > 0$. Algorithm 1 decides whether $d_{\mathrm{dF}}(\pi, \sigma) \leqslant \delta$ in time $\mathcal{O}((n+m)\log(nm))$.*

Correctness Note that Algorithm 1 considers potentially only very few points of the curve explicitly during its execution. Call the indices (p,q) of point pairs considered in some iteration of the algorithm (for any choice of greedy steps, if more than one exists) *greedy (point) pairs* and all points contained in some such pair *greedy points (of π and σ)*. The following useful monotonicity property holds: If some greedy point on π sees a point on σ that is yet to be traversed, all following greedy points on π will see it *until it is traversed*.

Lemma 4.5. *Let $(p_1, q_1), \dots, (p_i, q_i)$ be the greedy point pairs considered in the iterations $1, \dots, i$. It holds that*

1. $\mathrm{vis}_\sigma(\ell, q_i) \subseteq \mathrm{vis}_\sigma(p_i, q_i)$ *for all* $1 \leqslant \ell \leqslant p_i$, *and*
2. $\mathrm{vis}_\pi(p_i, \ell) \subseteq \mathrm{vis}_\pi(p_i, q_i)$ *for all* $1 \leqslant \ell \leqslant q_i$.

We exploit this monotonicity to prove that if Algorithm 1 finds a greedy point pair that allows no further greedy steps, then no feasible traversal of π and σ exists. We derive an even stronger statement using the following notion: For a greedy point pair (p,q), define $\textsc{stop}_\pi(\pi_{p..n}, \sigma_{q..m}) := \max(\mathrm{reach}_\pi(p,q) \cup \{p\}) + 1$ as the index of the first point after π_p on π which is not seen by σ_q, or $n+1$ if no such index exists. Let $\textsc{stop}_\sigma$ be defined symmetrically.

Lemma 4.6 (Correctness of Algorithm 1)**.** *Let (p,q) be a greedy point of π and σ, $p_{\mathrm{stop}} := \textsc{stop}_\pi(\pi_{p..n}, \sigma_{q..m})$ and $q_{\mathrm{stop}} := \textsc{stop}_\sigma(\pi_{p..n}, \sigma_{q..m})$. If on both curves, no greedy step from (p,q) exists, then $d_{\mathrm{dF}}(\pi, \sigma) > \delta$.*

In this case, it holds in particular that if $q_{\mathrm{stop}} < m$, then for all $1 \leqslant p' \leqslant n$, we have that $d_{\mathrm{dF}}(\pi_{1..p'}, \sigma_{1..q_{\mathrm{stop}}}) > \delta$ and if $p_{\mathrm{stop}} < n$, then $d_{\mathrm{dF}}(\pi_{1..p_{\mathrm{stop}}}, \sigma_{1..q'}) > \delta$ for all $1 \leqslant q' \leqslant m$.

Note that the correctness of Algorithm 1 follows immediately: If the algorithm is stuck, then $d_{\mathrm{dF}}(\pi, \sigma) > \delta$. Otherwise, it finds a feasible traversal.

Implementation In this extended abstract, we omit the discussion of how to implement greedy steps efficiently by reducing them to range searching on the point sets $P = \{(i, \pi_i) \mid i \in [1..n]\}$ and $Q = \{(i, \sigma_i) \mid i \in [1..m]\}$, yielding the following lemma.

Lemma 4.7. *For one-dimensional separated curves $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$, $\textsc{MaxGreedyStep}_\pi$, $\textsc{MinGreedyStep}_\pi$ and $\textsc{stop}_\pi$ can be implemented to run in time $\mathcal{O}(\log nm)$ after $O((n+m)\log nm)$ preprocessing. If π, σ are input curves of the reduced free-space problem then these procedures can be implemented in time $\mathcal{O}(\log 1/\varepsilon)$ after $O((n+m)\log 1/\varepsilon)$ preprocessing.*

4.3 Composition of one-dimensional curves

In this subsection, we highlight two composition properties of feasible traversals of one-dimensional curves that enable us to tackle the reduced free-space problem. The first result surprisingly allows to argue about the union of two subcurves.

Lemma 4.8. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be one-dimensional separated curves and let $I, J \subseteq [1..n]$ be intervals with $I \cap J \neq \emptyset$. If $d_{\mathrm{dF}}(\pi_I, \sigma) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_J, \sigma) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{I\cup J}, \sigma) \leqslant \delta$.*

The second result formalizes situations in which a traversal ϕ of subcurves has to cross a traversal ψ of other subcurves, in which case we can stick together parts of both to obtain new traversals.

Lemma 4.9. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be one-dimensional curves and consider intervals $I = [a_I, b_I] \subseteq [1..n]$ and $J \subseteq [1..m]$. If $d_{\mathrm{dF}}(\pi, \sigma_J) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_I, \sigma) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{1..b_I}, \sigma) \leqslant \delta$.*

4.4 Solving the Reduced Free-space Problem

4.4.1 Single Entry

Given the separated curves $\pi = (\pi_1, \ldots, \pi_n)$ and $\sigma = (\sigma_1, \ldots, \sigma_m)$ and entry set $E = \{1\}$, we show how to compute F^σ. The following recursive algorithm does the job.

Algorithm 2 Special Case: Single entry

```
function Find-σ-exits($\pi_{p..b}, \sigma_{q..d}$)
    if $q = d$ then
        if $\textsc{stop}_\pi(\pi_{p..b}, \sigma_q) = b + 1$ then return $\{q\}$    ▷ End of $\pi$ reachable while staying in $\sigma_q$
        else return $\emptyset$
    if $p' \leftarrow \textsc{MaxGreedyStep}_\pi(\pi_{p..b}, \sigma_{q..d})$ then
        return Find-σ-exits($\pi_{p'..b}, \sigma_{q..d}$))
    else if $q' \leftarrow \textsc{GreedyStep}_\sigma(\pi_{p..b}, \sigma_{q..d})$ then
        return Find-σ-exits($\pi_{p..b}, \sigma_{q..q'-1}$) $\cup$ Find-σ-exits($\pi_{p..b}, \sigma_{q'..d}$)
    else
        return Find-σ-exits($\pi_{p..b}, \sigma_{q..d-1}$)    ▷ No greedy step possible
```

Lemma 4.10. *Algorithm 2 correctly identifies F^σ given the single entry $E = \{1\}$.*

Proof. Clearly, if Find-σ-exits(π, σ) finds and returns an exit e on σ, then it is contained in F^σ, since the algorithm uses only feasible (greedy) steps. Conversely, we show that for all $I = [p..b]$ and $J = [q..d]$, where (p, q) is a greedy point pair of π and σ, and all $e \in J$ with $d_{\mathrm{dF}}(\pi_I, \sigma_{J\cap[1..e]}) \leqslant \delta$, we have $e \in$ Find-σ-Exits(π_I, σ_J), i.e. we find all exits.

Consider some call of Find-σ-Exits(π_I, σ_J) for which the precondition is fulfilled. If J consists only of a single point, then $J = \{e\}$, and a feasible traversal of π_I and σ_J exists if and only if σ_e sees all points on π_I. Let $I = [p..b]$, then this happens if and only if $\textsc{stop}_\pi(\pi_I, \sigma_e) = b + 1$, hence the base case is treated correctly.

Assume that $I = [p..b]$ and a maximal greedy step p' on π exists. Note that this step is a greedy step also with respect to $\sigma_{J\cap[1..e]}$. Hence by Lemma 4.6, if there is a traversal of $\pi_{p..b}$ and $\sigma_{J\cap[1..e]}$, then a traversal of $\pi_{[p'..b]}$ and $\sigma_{J\cap[1..e]}$ also exists.

Consider the case in which $J = [q..d]$ and a greedy step q' in σ exists. If $e < q'$, then $e \in [q..q'-1]$ and $J \cap [1..e] = [q..q'-1] \cap [1..e]$. Hence, e is found in the recursive call with $J' = [q..q'-1]$. If $e \geqslant q'$, then this step is also a greedy step with respect to the curves π_I and $\sigma_{J\cap[1..e]}$. Again, by Lemma 4.6, the existence of a feasible traversal of π_I and σ_J implies that also a feasible traversal of π_I and $\sigma_{J\cap[q'..e]}$ exists.

It remains to regard the case in which no greedy step exists. By Lemma 4.6, there is no feasible traversal of $\pi_{1..n}$ and $\sigma_{1..d}$. This implies $e \neq d$ and all exits are found in the recursive call with $J' = [q, d-1]$. □

Lemma 4.11. *FIND-σ-EXITS$(\pi_{p..b}, \sigma_{q..d})$ runs in time $\mathcal{O}((d-q+1)\cdot\log 1/\varepsilon)$.*

4.4.2 Entries on π, Exits on π

In this section, we tackle the task of determining F^π given a set of entries E on π. It is essential to avoid computing the exits by iterating over every single entry. We show how to divide π into disjoint subcurves that can be solved by a single call to FIND-π-EXITS each.

Assume we want to traverse $\pi_{p..b}$ and $\sigma_{q..d}$ starting in π_p and σ_q. Let $u(p) := \max\{p' \in [p,b] \mid \exists q \leqslant q' \leqslant d : d_{\mathrm{dF}}(\pi_{p..p'}, \sigma_{q..q'}) \leqslant \delta\}$ be the last point on π that is reachable while traversing an arbitrary subcurve of $\sigma_{q..d}$ that starts in σ_q. This point fulfills the following properties.

Lemma 4.12. *It holds that*

1. *If there are $p \leqslant e \leqslant e' \leqslant u(p)$ with $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma_{q..d}) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{p..e'}, \sigma_{q..d}) \leqslant \delta$.*
2. *For all $p \leqslant e \leqslant u(p) < e'$, we have that $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma_{q..d}) > \delta$.*

The above lemma implies that we can ignore all entries in $[p..u(p)]$ except for p and that all exits reachable from p are contained in the interval $[p..u(p)]$. This gives rise to the following algorithm.

Algorithm 3 Given entry points E on π, compute all exits on π.

```
function π-EXITS-FROM-π$(\pi, \sigma, E)$
    $S \leftarrow \emptyset$
    while $E \neq \emptyset$ do
        $\hat{p} \leftarrow$ pop minimal index from $E$
        $p \leftarrow \hat{p}$, $q \leftarrow 1$
        repeat
            if $q' \leftarrow$ MAXGREEDYSTEP$_\sigma(\pi_{p..n}, \sigma_{q..m})$ then $q \leftarrow q'$
            if $p' \leftarrow$ GREEDYSTEP$_\pi(\pi_{p..n}, \sigma_{q..m})$ then $p \leftarrow p'$
        until no greedy step was found in the last iteration
        $\overline{p} \leftarrow$ STOP$_\pi(\pi_{p..n}, \sigma_{q..m}) - 1$        $\triangleright$ determines the maximal reachable point $u(\hat{p})$
        $S \leftarrow S \cup$ FIND-π-EXITS$(\pi_{\hat{p}..\overline{p}}, \sigma)$
        $E \leftarrow E \cap [\overline{p}+1, n]$        $\triangleright$ drops all entries in $[\hat{p}, u(\hat{p})]$
    return $S$
```

In the full version of this paper we prove correctness of the above algorithm as follows.

Lemma 4.13. *Algorithm 3 correctly computes F^π in time $\mathcal{O}((n+m)\log 1/\varepsilon)$.*

References

[1] P. Agarwal, R. B. Avraham, H. Kaplan, and M. Sharir. Computing the discrete Fréchet distance in subquadratic time. In *Proc. 24th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA'13)*, pages 156–167, 2013.

[2] H. Alt. The computational geometry of comparing shapes. In *Efficient Algorithms*, volume 5760 of *LNCS*, pages 235–248. Springer, 2009.

[3] H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.

[4] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.

[5] H. Alt, C. Knauer, and C. Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38 (1):45–58, 2004.

[6] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. In *Proc. 14th Annu. European Symp. Algorithms (ESA'06)*, pages 52–63. Springer, 2006.

[7] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proc. 31st International Conf. Very Large Data Bases (VLDB'05)*, pages 853–864, 2005.

[8] K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annu. IEEE Sympos. Foundations of Computer Science (FOCS'14)*. IEEE, 2014. To appear, preprint at arXiv:1404.1448.

[9] K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA'09)*, pages 645–654. SIAM, 2009.

[10] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Internat. J. Comput. Geom. Appl.*, 21(3):253–282, 2011.

[11] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog - with an application to Alt's conjecture. In *Proc. 25th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA'14)*, pages 1399–1413, 2014.

[12] E. W. Chambers, É. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010.

[13] B. Chazelle and L. J. Guibas. Fractional cascading: I. a data structuring technique. *Algorithmica*, 1 (2):133–162, 1986.

[14] D. Chen, A. Driemel, L. J. Guibas, A. Nguyen, and C. Wenk. Approximate map matching with respect to the Fréchet distance. In *Proc. 13th Workshop on Algorithm Engineering and Experiments (ALENEX'11)*, pages 75–83. SIAM, 2011.

[15] A. F. Cook and C. Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms*, 7(1):193–204, 2010.

[16] A. Driemel and S. Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

[17] A. Driemel, S. Har-Peled, and C. Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012.

[18] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

[19] M. Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In *Proc. 8th Sympos. Theoret. Aspects Comput. Sci. (STACS'91)*, volume 480 of *LNCS*, pages 127–136. Springer, 1991.

[20] J. Gudmundsson and M. Smid. Fréchet queries in geometric trees. In H. L. Bodlaender and G. F. Italiano, editors, *Algorithms - ESA 2013*, volume 8125 of *LNCS*, pages 565–576. Springer, 2013.

[21] S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. In *Proc. 27th Annu. Symp. Comp. Geometry (SoCG'11)*, pages 448–457. ACM, 2011.

[22] P. Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proc.*

18th Annu. Symp. Comp. Geometry (SoCG'02), pages 102–106. ACM, 2002.

[23] G. S. Lueker. A data structure for orthogonal range queries. In *Proc. 19th Annu. Sympos. Foundations of Computer Science*, SFCS '78, pages 28–34. IEEE, 1978.

[24] A. Maheshwari, J.-R. Sack, K. Shahbaz, and H. Zarrabi-Zadeh. Fréchet distance with speed limits. *Computational Geometry*, 44(2):110–120, 2011.

[25] M. E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proc. 7th Intl. Conf. Comp. Vision*, volume 1, pages 108–115. IEEE, 1999.

[26] D. E. Willard. *Predicate-oriented database search algorithms*. PhD thesis, Aiken Comput. Lab, Harvard Univ., Cambridge, MA, 1978. Report TR-20-78.

Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds
(full version)

Karl Bringmann[*] Marvin Künnemann[†]

July 7, 2014

Abstract

The Fréchet distance is a well-studied and very popular measure of similarity of two curves. The best known algorithms have quadratic time complexity, which has recently been shown to be optimal assuming the Strong Exponential Time Hypothesis (SETH) [Bringmann FOCS'14].

To overcome the worst-case quadratic time barrier, restricted classes of curves have been studied that attempt to capture realistic input curves. The most popular such class are c-packed curves, for which the Fréchet distance has a $(1+\varepsilon)$-approximation in time $\tilde{\mathcal{O}}(cn/\varepsilon)$ [Driemel et al. DCG'12]. In dimension $d \geqslant 5$ this cannot be improved to $\mathcal{O}((cn/\sqrt{\varepsilon})^{1-\delta})$ for any $\delta > 0$ unless SETH fails [Bringmann FOCS'14].

In this paper, exploiting properties that prevent stronger lower bounds, we present an improved algorithm with runtime $\tilde{\mathcal{O}}(cn/\sqrt{\varepsilon})$. This is optimal in high dimensions apart from lower order factors unless SETH fails. Our main new ingredients are as follows: For filling the classical free-space diagram we project short subcurves onto a line, which yields one-dimensional separated curves with roughly the same pairwise distances between vertices. Then we tackle this special case in near-linear time by carefully extending a greedy algorithm for the Fréchet distance of one-dimensional separated curves.

[*] Max Planck Institute for Informatics, Campus E1 4, 66123 Saarbrücken, Germany; kbringma@mpi-inf.mpg.de.
Karl Bringmann is a recipient of the *Google Europe Fellowship in Randomized Algorithms*, and this research is supported in part by this Google Fellowship.

[†] Max Planck Institute for Informatics, Campus E1 4, 66123 Saarbrücken, Germany; marvin@mpi-inf.mpg.de. Saarbrücken Graduate School of Computer Science, Germany

1 Introduction

The Fréchet distance is a very popular measure of similarity of two given curves and has two classic variants. Roughly speaking, the *continuous Fréchet distance* of two curves π, σ is the minimal length of a leash required to connect a dog to its owner, as they walk without backtracking along π and σ, respectively. In the *discrete Fréchet distance* we replace the dog and its owner by two frogs – in each time step each frog can jump to the next vertex along its curve or stay where it is.

In a seminal paper in 1991, Alt and Godau introduced the continuous Fréchet distance to computational geometry [4, 19]. For polygonal curves π and σ with n and m vertices[1], respectively, they presented an $\mathcal{O}(nm \log nm)$ algorithm. The discrete Fréchet distance was defined by Eiter and Mannila [18], who presented an $\mathcal{O}(nm)$ algorithm.

Since then, Fréchet distance has become a rich field of research: The literature contains generalizations to surfaces (see, e.g., [3]), approximation algorithms for realistic input curves ([5, 6, 17]), the geodesic and homotopic Fréchet distance (see, e.g., [12, 15]), and many more variants (see, e.g., [9, 16, 22, 24]). As a natural measure for curve similarity [2], the Fréchet distance has found applications in various areas such as signature verification (see, e.g., [25]), map-matching tracking data (see, e.g., [7]), and moving objects analysis (see, e.g., [10]).

Apart from log-factor improvements [1, 11] the quadratic complexity of the classic algorithms for the continuous and discrete Fréchet distance are still the state of the art. In fact, the first author recently showed a conditional lower bound: Assuming the Strong Exponential Time Hypothesis (SETH) there is no algorithm for the (continuous or discrete) Fréchet distance in time $\mathcal{O}((nm)^{1-\delta})$ for any $\delta > 0$, so apart from lower order terms of the form $n^{o(1)}$ the classic algorithms are optimal [8].

In attempts to obtain faster algorithms for realistic inputs, various restricted classes of curves have been considered, such as backbone curves [6], κ-bounded and κ-straight curves [5], and ϕ-low density curves [17]. The most popular model of realistic inputs are *c-packed curves*. A curve π is c-packed if for any point $z \in \mathbb{R}^d$ and any radius $r > 0$ the total length of π inside the ball $B(z, r)$ is at most cr, where $B(z, r)$ is the ball of radius r around z. This model has been used for several generalizations of the Fréchet distance, such as map matching [14], the mean curve problem [21], a variant of the Fréchet distance allowing shortcuts [16], and Fréchet matching queries in trees [20]. Driemel et al. [17] introduced c-packed curves and presented a $(1+\varepsilon)$-approximation for the continuous Fréchet distance in time $\mathcal{O}(cn/\varepsilon + cn \log n)$, which works in any $\mathbb{R}^d$, $d \geqslant 2$. Assuming SETH, the following lower bounds have been shown for c-packed curves: (1) For sufficiently small constant $\varepsilon > 0$ there is no $(1+\varepsilon)$-approximation in time $\mathcal{O}((cn)^{1-\delta})$ for any $\delta > 0$ [8]. Thus, for constant ε the algorithm by Driemel et al. is optimal apart from lower order terms of the form $n^{o(1)}$. (2) In any dimension $d \geqslant 5$ and for varying $\varepsilon > 0$ there is no $(1+\varepsilon)$-approximation in time $\mathcal{O}((cn/\sqrt{\varepsilon})^{1-\delta})$ for any $\delta > 0$ [8]. Note that this does not match the runtime of the algorithm by Driemel et al. for any $\varepsilon = n^{-b}$ and constant $b > 0$.

In this paper we improve upon the algorithm by Driemel et al. [17] by presenting an algorithm that matches the conditional lower bound of [8].

Theorem 1.1. *For any $0 < \varepsilon \leqslant 1$ we can compute a $(1+\varepsilon)$-approximation on c-packed curves for the continuous and discrete Fréchet distance in time $\tilde{\mathcal{O}}(cn/\sqrt{\varepsilon})$.*

Specifically, our runtime is $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log(1/\varepsilon) + cn \log n)$ for the discrete variant and $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log^2(1/\varepsilon) + cn \log n)$ for the continuous variant.

We want to highlight that in general dimensions (specifically, $d \geqslant 5$) this runtime is *optimal* (apart from lower order terms of the form $n^{o(1)}$ unless SETH fails [8]). Moreover, we obtained

[1] We always assume that $m \leqslant n$.

our new algorithm by investigating why the conditional lower bound [8] cannot be improved and exploiting the discovered properties. Thus, the above theorem is the outcome of a synergetic effect of algorithms and lower bounds[2].

We remark that the same algorithm also yields improved runtime guarantees for other models of realistic input curves, like κ-bounded and κ-straight curves, where we are also able to essentially replace ε by $\sqrt{\varepsilon}$ in the runtime bound. In contrast to c-packed curves, it is not clear how far these bounds are from being optimal. See Section 3.2 for details.

Outline We give an improved algorithm that approximately decides whether the Fréchet distance of two given curves π, σ is at most δ. Using a construction of [16] to search over possible values of δ, this yields an improved approximation algorithm. We partition our curves into subcurves, each of which is either a *long segment*, i.e., a single segment of length at least $\Lambda = \Theta(\sqrt{\varepsilon}\delta)$, or a *piece*, i.e., a subcurve staying in the ball of radius Λ around its initial vertex. Now we run the usual algorithm that explores the reachable free-space (see Section 2 for definitions), however, we treat regions spanned by a piece π' of π and a piece σ' of σ in a special way. Typically, if π', σ' consist of n', m' segments then their free-space would be resolved in time $\mathcal{O}(n'm')$. Our overall speedup comes from reducing this runtime to $\tilde{\mathcal{O}}(n' + m')$, which is our first main contribution. To this end, we consider the line through the initial vertices of the pieces π', σ', and project π', σ' onto this line to obtain curves $\hat{\pi}, \hat{\sigma}$. Since π', σ' are *pieces*, i.e., they stay within distance $\Lambda = \Theta(\sqrt{\varepsilon}\delta)$ of their initial vertices, this projection does not change distances from π to σ significantly (it follows from the Pythagorean theorem that any distance of approximately δ is changed, by the projection, by less than $\varepsilon\delta$). Thus, we can replace π', σ' by $\hat{\pi}, \hat{\sigma}$ without introducing too much error. Note that $\hat{\pi}, \hat{\sigma}$ are *one-dimensional* curves; without loss of generality we can assume that they lie on $\mathbb{R}$. Moreover, we show how to ensure that $\hat{\pi}, \hat{\sigma}$ are *separated*, i.e., all vertices of $\hat{\pi}$ lie above 0 and all vertices of $\hat{\sigma}$ lie below 0. Hence, we reduced our problem to resolving the free-space region of one-dimensional separated curves.

It is known[3] that the Fréchet distance of one-dimensional separated curves can be computed in near-linear time, essentially since we can walk along π and σ with *greedy steps* to either find a feasible traversal or bottleneck subcurves. However, we face the additional difficulty that we have to resolve the *free-space region* of one-dimensional separated curves, i.e., given entry points on $\hat{\pi}$ and $\hat{\sigma}$, compute all exits on $\hat{\pi}$ and $\hat{\sigma}$. Our second main contribution is that we present an extension of the known result to handle this much more complex problem.

Organization We start with basic definitions and techniques borrowed from [16] in Section 2. In Section 3 we present our approximate decision procedure which reduces the problem to one-dimensional separated curves. We solve the latter in Section 4. In the whole paper, we focus on the continuous Fréchet distance. It is straightforward to obtain a similar algorithm for the discrete variant, in fact, then Section 4.1 becomes obsolete, which is why we save a factor of $\log 1/\varepsilon$ in the running time.

[2] This yields one more reason why conditional lower bounds such as [8] should be studied, as they can show tractable cases and suggest properties that make these cases tractable.

[3] We thank Wolfgang Mulzer for pointing us to this result by Matias Korman and Sergio Cabello (personal communication). To the best of our knowledge this result is not published.

2 Preliminaries

For $z \in R^d$, $r > 0$ we let $B(z, r)$ be the ball of radius r around z. For $i, j \in \mathbb{N}$, $i \leqslant j$, we let $[i..j] := \{i, i+1, \ldots, j\}$, which is not to be confused with the real interval $[i, j] = \{x \in \mathbb{R} \mid i \leqslant x \leqslant j\}$. Throughout the paper we fix the dimension $d \geqslant 2$. A (polygonal) curve π is defined by its vertices $(\pi_1, \ldots, \pi_n)$ with $\pi_p \in \mathbb{R}^d$, $p \in [1..n]$. We let $|\pi| = n$ be the number of vertices of π and $\|\pi\|$ be its total length $\sum_{i=1}^{n-1} \|p_i - p_{i+1}\|$. We write $\pi_{p..b}$ for the subcurve $(\pi_p, \pi_{p+1}, \ldots, \pi_b)$. Similarly, for an interval $I = [p..b]$ we write $\pi_I = \pi_{p..b}$. We can also view π as a continuous function $\pi \colon [1, n] \to \mathbb{R}^d$ with $\pi_{p+\lambda} = (1-\lambda)\pi_p + \lambda\pi_{p+1}$ for $p \in [1..n-1]$ and $\lambda \in [0, 1]$. For the second curve $\sigma = (\sigma_1, \ldots, \sigma_m)$ we will use indices of the form $\sigma_{q..d}$ for the reader's convenience.

Variants of the Fréchet distance Let Φ_n be the set of all continuous and non-decreasing functions ϕ from $[0, 1]$ onto $[1, n]$. The *continuous Fréchet distance* between two curves π, σ with n and m vertices, respectively, is defined as

$$d_{\mathrm{F}}(\pi, \sigma) := \inf_{\substack{\phi_1 \in \Phi_n \\ \phi_2 \in \Phi_m}} \max\nolimits_{t \in [0,1]} \|\pi_{\phi_1(t)} - \sigma_{\phi_2(t)}\|,$$

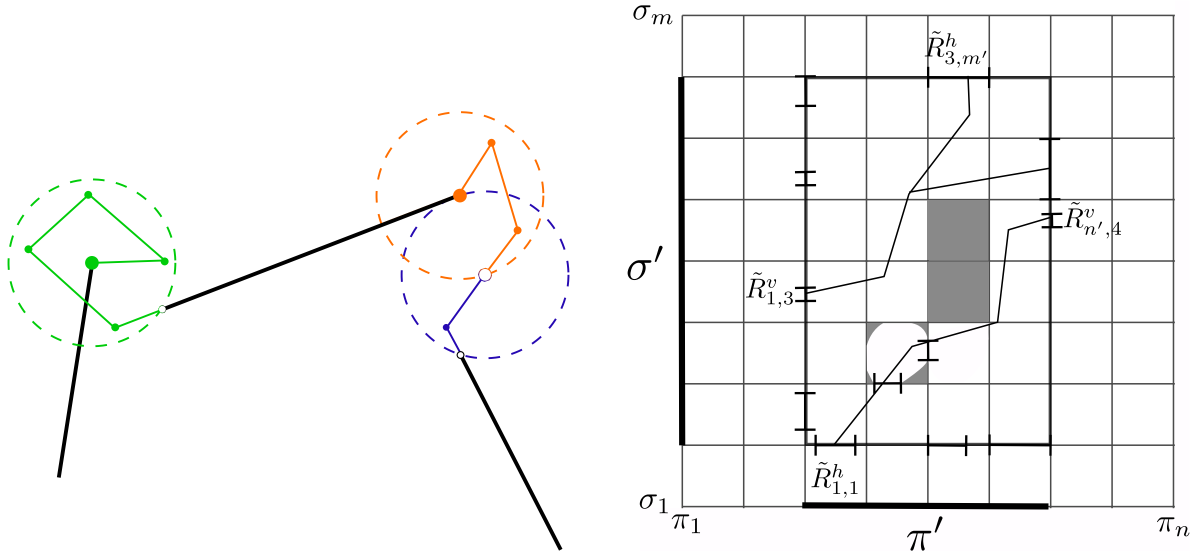
where $\|.\|$ denotes the Euclidean distance. We call $\phi := (\phi_1, \phi_2)$ a (continuous) *traversal* of (π, σ), and say that it has *width* $\max_{t \in [0,1]} \|\pi_{\phi_1(t)} - \sigma_{\phi_2(t)}\|$.

In the discrete case, we let Δ_n be the set of all non-decreasing functions ϕ from $[0, 1]$ onto $[1..n]$. We obtain the *discrete Fréchet distance* $d_{\mathrm{dF}}(\pi, \sigma)$ by replacing Φ_n and Φ_m by Δ_n and Δ_m. We obtain an analogous notion of a (discrete) *traversal* and its *width*. Note that any $\phi \in \Delta_n$ is a staircase function attaining all values in $[1..n]$. Hence, $(\phi_1(t), \phi_2(t))$ changes only at finitely many points in time t. At any such *time step*, we jump to the next vertex in π or σ or both.

Free-space diagram The discrete *free-space* of curves π, σ is defined as $\mathcal{D}^d_{\leqslant\delta}(\pi, \sigma) := \{(p, q) \in [1..n] \times [1..m] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$. Note that any discrete traversal of π, σ of width at most δ corresponds to a monotone sequence of points in the free-space where at each point in time we increase p or q or both. Because of this property, the free-space is a standard concept used in many algorithms for the Fréchet distance.

The continuous free-space is defined as $\mathcal{D}_{\leqslant\delta}(\pi, \sigma) := \{(p, q) \in [1, n] \times [1, m] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$. Again, a monotone path from $(1, 1)$ to (n, m) in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$ corresponds to a traversal of width at most δ. It is well-known [4, 19] that each *free-space cell* $C_{i,j} := \{(p, q) \in [i, i+1] \times [j, j+1] \mid \|\pi_p - \sigma_q\| \leqslant \delta\}$ (for $i \in [1..n-1], j \in [1..m-1]$) is convex, specifically it is the intersection of an ellipsoid with $[i, i+1] \times [j, j+1]$. In particular, the intersection of the free-space with any interval $[i, i+1] \times \{j\}$ (or $\{i\} \times [j, j+1]$) is an interval $I^h_{i,j}$ (or $I^v_{i,j}$), and for any such interval the subset that is reachable by a monotone path from $(1, 1)$ is an interval $R^h_{i,j}$ (or $R^v_{i,j}$). Moreover, in constant time one can solve the following *free-space cell problem*: Given intervals $R^h_{i,j} \subseteq [i, i+1] \times \{j\}, R^v_{i,j} \subseteq \{i\} \times [j, j+1]$, determine the intervals $R^h_{i,j+1} \subseteq [i, i+1] \times \{j+1\}, R^v_{i+1,j} \subseteq \{i+1\} \times [j, j+1]$ consisting of all points that are reachable from a point in $R^h_{i,j} \cup R^v_{i,j}$ by a monotone path within the free-space cell $C_{i,j}$. Solving this problem for all cells from lower left to upper right we determine whether (n, m) is reachable from $(1, 1)$ by a monotone path and thus decide whether the Fréchet distance is at most δ.

From approximate deciders to approximation algorithms An *approximate decider* is an algorithm that, given curves π, σ and $\delta > 0, 0 < \varepsilon \leqslant 1$, returns one of the outputs (1) $d_{\mathrm{F}}(\pi, \sigma) > \delta$



(a) This figure illustrates our partitioning of a curve into *pieces* (contained in dashed circles) and *long segments* (bold edges).

(b) The free-space problem for pieces π' and σ' in the free-space diagram of π and σ. Given entry intervals on the lower and left boundary of the region, compute exit intervals on the upper and right boundary.

Figure 1: Definition and treatment of pieces.

or (2) $d_F(\pi, \sigma) \leqslant (1+\varepsilon)\delta$. In any case, the returned answer has to be correct. In particular, if $\delta < d_F(\pi, \sigma) \leqslant (1+\varepsilon)\delta$ the algorithm may return either of the two outputs.

Let $D(\pi, \sigma, \delta, \varepsilon)$ be the runtime of an approximate decider and set $D(\pi, \sigma, \varepsilon) := \max_{\delta>0} D(\pi, \sigma, \delta, \varepsilon)$. We assume polynomial dependence on ε, in particular, that there are constants $0 < c_1 < c_2 < 1$ such that for any $1 < \varepsilon \leqslant 1$ we have $c_1 D(\pi, \sigma, \varepsilon/2) \leqslant D(\pi, \sigma, \varepsilon) \leqslant c_2 D(\pi, \sigma, \varepsilon/2)$. Driemel et al. [17] gave a construction of a $(1+\varepsilon)$-approximation for the Fréchet distance given an approximate decider. (This follows from [17, Theorem 3.15] after replacing their concrete approximate decider with runtime "$\mathcal{O}(N(\varepsilon, \pi, \sigma))$" by any approximate decider with runtime $D(\pi, \sigma, \varepsilon)$.)

Lemma 2.1. *Given an approximate decider with runtime $D(\pi, \sigma, \varepsilon)$ we can construct a $(1+\varepsilon)$-approximation for the Fréchet distance with runtime $\mathcal{O}\big(D(\pi, \sigma, \varepsilon) + D(\pi, \sigma, 1) \log n\big)$.*

3 The approximate decider

Let π, σ be curves for which we want to (approximately) decide whether $d_F(\pi, \sigma) > \delta$ or $d_F(\pi, \sigma) \leqslant (1+\varepsilon)\delta$. We modify the curve π by introducing new vertices as follows. Start with the initial vertex π_1 as current vertex. If the segment following the current vertex has length at least $\Lambda = \Lambda_{\varepsilon,\delta} := \min\{\frac{1}{2}\sqrt{\varepsilon}, \frac{1}{4}\} \cdot \delta$ then mark this segment as *long* and set the next vertex as the current vertex. Otherwise follow π from the current vertex π_x to the first point π_y such that $\|\pi_x - \pi_y\| = \Lambda$ (or until we reach the last vertex of π). If π_y is not a vertex, but lies on some segment of π, then introduce a new vertex at π_y. Mark $\pi_{x..y}$ as a *piece* of π and set π_y as current vertex. Repeat until π is completely traversed. Since this procedure introduces at most $|\pi|$ new vertices and does not change the shape of π, with slight abuse of notation we call the resulting curve again π and set $n := |\pi|$. This partitions π into subcurves $\pi^1, \ldots, \pi^k$, with $\pi^s = \pi_{p_s..b_s}$, where every part π^s is either (see also Figure 1a)

- a *long segment*: $b_s = p_s + 1$ and $\|\pi_{p_s} - \pi_{b_s}\| \geqslant \Lambda$, or
- a *piece*: $\|\pi_{p_s} - \pi_{b_s}\| = \Lambda$ and $\|\pi_{p_s} - \pi_x\| < \Lambda$ for all $x \in [p_s, b_s)$.

Note that the last piece actually might have distance $\|\pi_{p_s} - \pi_{b_s}\|$ less than Λ, however, for simplicity we assume equality for all pieces (in fact, a special handling of the last piece would only be necessary in Lemma 3.6). Similarly, we introduce new vertices on σ and partition it into subcurves $\sigma^1, \ldots, \sigma^\ell$, with $\sigma^t = \sigma_{q_t..d_t}$, each of which is a long segment or a piece. Let $m := |\sigma|$.

We do not want to resolve each free-space cell on its own, as in the standard decision algorithm for the Fréchet distance. Instead, for any pair of pieces we want to consider the free-space region spanned by the two pieces at once, see Figure 1b. This is made formal by the following subproblem.

Problem 3.1 (Free-space region problem). Given $\delta > 0$, $0 < \varepsilon \leqslant 1$, curves π, σ with n and m vertices, and *entry intervals* $\tilde{R}^h_{i,1} \subseteq [i, i+1] \times \{1\}$ for $i \in [1..n)$ and $\tilde{R}^v_{1,j} \subseteq \{1\} \times [j, j+1]$ for $j \in [1..m)$, compute *exit intervals* $\tilde{R}^h_{i,m} \subseteq [i, i+1] \times \{m\}$ for $i \in [1..n)$ and $\tilde{R}^v_{n,j} \subseteq \{n\} \times [j, j+1]$ for $j \in [1..m)$ such that (1) the exit intervals contain all points reachable from the entry intervals by a monotone path in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$ and (2) all points in the exit intervals are reachable from the entry intervals by a monotone path in $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi, \sigma)$.

To stress that we work with approximations, we denote reachable intervals by $\tilde{R}$ instead of R in the remainder of the paper.

The standard solution to the free-space region problem would split it up into $n \cdot m$ free-space cells and resolve each cell in constant time, resulting in an $\mathcal{O}(n \cdot m)$ algorithm (this solves the problem even exactly, i.e., for $\varepsilon = 0$). Restricted to pieces, we will show the following improvement, which will yield the desired overall speedup of a factor of $\sqrt{\varepsilon}$.

Lemma 3.2. *If π and σ are pieces then the free-space region problem can be solved in time $\mathcal{O}((n + m)\log^2 1/\varepsilon)$.*

We will prove this lemma in Sections 3.3 and 4.

Algorithm 3.3. Using an algorithm for the free-space region problem on pieces as in Lemma 3.2, we obtain an approximate decider for the Fréchet distance a follows. We create a directed graph which has a node $v_{s,t}$ for every region $[p_s, b_s] \times [q_t, d_t]$ spanned by pieces π^s and σ^t, and a node $u_{i,j}$ for every remaining region $[i, i+1] \times [j, j+1]$ (which is not contained in any region spanned by two pieces), $i \in [1..n)$, $j \in [1..m)$. We add edges between two nodes whenever their regions touch (i.e., have a common interval I on their boundary), and direct this edge from the region that is to the left or below I to the other one. With each node $u_{i,j}$ we store the entry intervals $\tilde{R}^h_{i,j}$ and $\tilde{R}^v_{i,j}$, and with each node $v_{s,t}$ we store the entry intervals $\tilde{R}^h_{i,q_t} \subseteq [i, i+1] \times \{q_t\}$ for $i \in [p_s..b_s)$ and $\tilde{R}^v_{p_s,j} \subseteq \{p_s\} \times [j, j+1]$ for $j \in [q_t..d_t)$. After correctly initializing the outer reachability intervals $\tilde{R}^h_{i,1}$ and $\tilde{R}^v_{1,j}$, we follow any topological ordering of this graph. For any node $u_{i,j}$, we resolve its region by solving the corresponding free-space *cell* problem in constant time. For any node $v_{s,t}$, we solve the corresponding free-space *region* problem on $\pi' = \pi^s, \sigma' = \sigma^t$ (and $\delta' = \delta, \varepsilon' = \varepsilon$) using Lemma 3.2. Finally, we return $d_{\mathrm{F}}(\pi, \sigma) \leqslant (1 + \varepsilon)\delta$ if $(n, m) \in \tilde{R}^h_{n-1,m}$ and $d_{\mathrm{F}}(\pi, \sigma) > \delta$ otherwise.

Lemma 3.4. *Algorithm 3.3 is a correct approximate decider.*

Proof. Observe that if $(n, m) \in \tilde{R}^h_{n-1,m}$ then there exists a monotone path from $(1, 1)$ to (n, m) in $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi, \sigma)$, which implies $d_{\mathrm{F}}(\pi, \sigma) \leqslant (1 + \varepsilon)\delta$. If $d_{\mathrm{F}}(\pi, \sigma) \leqslant \delta$ then there is a monotone path from $(1, 1)$ to (n, m) in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$, implying $(n, m) \in \tilde{R}^h_{n-1,m}$. □

In the above algorithm we can ignore *unreachable* nodes, i.e., nodes where all stored entry intervals would be empty. To this end, we fix a topological ordering by mapping a node corresponding to a region $[x_1, x_2] \times [y_1, y_2]$ to $x_2 + y_2$ and sorting by this value ascendingly. This yields $n+m$ layers of nodes, where the order within each layer is arbitrary. For each layer we build a dictionary data structure (a hash table), in which we store only the reachable nodes of this layer. This allows to quickly enumerate all reachable nodes of a layer. The total overhead for managing the $n+m$ dictionaries is $\mathcal{O}(n+m)$.

Let us analyze the runtime of the obtained approximate decider. Let S be the set of non-empty free-space cells $C_{i,j}$ of $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi,\sigma)$ such that i or j is not contained in a piece. Moreover, let T be the set of all pairs (s,t) such that π^s, σ^t are pieces with initial vertices within distance $(1+\varepsilon)\delta + 2\Lambda$. Define $N(\pi,\sigma,\delta,\varepsilon) := |S| + \sum_{(s,t)\in T}(|\pi^s| + |\sigma^t|)$ and set $N(\pi,\sigma,\varepsilon) := \max_{\delta>0} N(\pi,\sigma,\delta,\varepsilon)$. Since the algorithm considers only *reachable* cells and any reachable cell is also non-empty, the cost over all free-space cell problems solved by our approximate decider is bounded by $\mathcal{O}(|S|)$. Since every reachable (thus non-empty) region spanned by two pieces has initial points within distance $(1+\varepsilon)\delta + 2\Lambda$, the second term bounds the cost over all free-space region problems on pieces (apart from the $\log^2 1/\varepsilon$ factor). Hence, we obtain the following.

Lemma 3.5. *The approximate decider has runtime* $D(\pi,\sigma,\varepsilon) = \mathcal{O}(N(\pi,\sigma,\varepsilon)\cdot\log^2 1/\varepsilon)$.

3.1 The free-space complexity of c-packed curves

Recall that a curve π is c-packed if for any point $z \in \mathbb{R}^d$ and any radius $r > 0$ the total length of π inside the ball $B(z,r)$ is at most cr.

Lemma 3.6. *Let π, σ be c-packed curves with n vertices in total and $\varepsilon > 0$. Then $N(\pi,\sigma,\varepsilon) = \mathcal{O}(cn/\sqrt{\varepsilon})$.*

Proof. Our proof uses a similar argument as [16, Lemma 4.4]. Let $\delta > 0$ be arbitrary. First consider the set S of non-empty free-space cells $C_{i,j}$ of $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi,\sigma)$ such that i or j is not contained in a piece. Then one of the segments $\pi_{i..i+1}$ and $\sigma_{j..j+1}$ is long, i.e., of length at least $\Lambda = \min\{\frac{1}{2}\sqrt{\varepsilon}, \frac{1}{4}\}\cdot\delta$. We charge the cell $C_{i,j}$ to the shorter of the two segments. Let us analyze how often any segment $v = \pi_{i..i+1}$ can be charged. Consider the ball B of radius $r := \frac{1}{2}\|v\| + (1+\varepsilon)\delta + \max\{\|v\|,\Lambda\}$ centered at the midpoint of v. Every segment $u = \sigma_{j..j+1}$ with $(i,j) \in S$, which charges v, is of length at least $\mu := \max\{\|v\|,\Lambda\}$ (since it is longer than v and a long segment) and contributes at least μ to the total length of σ in B. Since σ is c-packed, the number of such charges is at most

$$\frac{\|\sigma \cap B\|}{\mu} \leqslant \frac{cr}{\mu} \leqslant \frac{c(\frac{1}{2}\|v\| + (1+\varepsilon)\delta + \max\{\|v\|,\Lambda\})}{\max\{\|v\|,\Lambda\}} \leqslant \tfrac{3}{2}c + \frac{c(1+\varepsilon)\delta}{\min\{\frac{1}{2}\sqrt{\varepsilon},\frac{1}{4}\}\cdot\delta} = \mathcal{O}\Big(\frac{c}{\sqrt{\varepsilon}}\Big).$$

Thus, the contribution of $|S|$ to the free-space complexity $N(\pi,\sigma,\varepsilon)$ is $\mathcal{O}(cn/\sqrt{\varepsilon})$.

Let T be the set of all pairs (s,t) such that π^s, σ^t are pieces of π, σ with initial vertices within distance $(1+\varepsilon)\delta + 2\Lambda$, and consider $\Sigma := \sum_{(s,t)\in T}(|\pi^s| + |\sigma^t|)$. We distribute Σ over the segments of π, σ by charging 1 to every segment of π^s and σ^t for any pair $(s,t) \in T$. Let us analyze how often any segment v of a piece π^s can be charged. Consider the ball B' of radius $r' := (1+\varepsilon)\delta + 3\Lambda$ around the initital vertex π_{p_s} of π^s. Since $\|\sigma^t\| \geqslant \Lambda$, for any $(s,t) \in T$ the piece σ^t contributes at least Λ to the total length of σ in B'. Since σ is c-packed, the number of such charges to v is at most

$$\frac{\|\sigma \cap B'\|}{\Lambda} \leqslant \frac{cr'}{\Lambda} = \frac{c(1+\varepsilon+\frac{3}{2}\sqrt{\varepsilon})}{\min\{\frac{1}{2}\sqrt{\varepsilon},\frac{1}{4}\}} = \mathcal{O}\Big(\frac{c}{\sqrt{\varepsilon}}\Big).$$

Hence, the contribution of Σ to the free-space complexity $N(\pi, \sigma, \varepsilon)$ is also at most $\mathcal{O}(cn/\sqrt{\varepsilon})$, which finishes the proof. $\square$

Combining Lemmas 3.6, 3.5, and 2.1, we obtain an approximation algorithm for the Fréchet distance with running time $\mathcal{O}(\frac{cn}{\sqrt{\varepsilon}} \log^2 1/\varepsilon + cn \log n) = \tilde{\mathcal{O}}(\frac{cn}{\sqrt{\varepsilon}})$, as desired.

3.2 The free-space complexity of κ-bounded and κ-straight curves

Definition 3.7. *Let $\kappa \geqslant 1$ be a given parameter. A curve π is* κ-straight *if for any $p, b \in [1, |\pi|]$ we have $\|\pi_{p..b}\| \leqslant \kappa \|\pi_p - \pi_b\|$. A curve π is* κ-bounded *if for all p, b the subcurve $\pi_{p..b}$ is contained in $B(\pi_p, r) \cup B(\pi_b, r)$, where $r = \frac{\kappa}{2}\|\pi_p - \pi_b\|$.*

The following lemma from [16] allows us to transfer our speedup for c-packed curves directly to κ-straight curves.

Lemma 3.8. *A κ-straight curve is 2κ-packed.*

In the remainder of this section we consider κ-bounded curves, closely following [16, Sect. 4.2].

Lemma 3.9. *Let $\delta > 0$, $0 < \varepsilon \leqslant 1$, $\lambda > 0$, and let π be a κ-bounded curve with disjoint subcurves $\pi^1, \ldots, \pi^k$, where $\pi^s = \pi_{p_s..b_s}$ and $\|\pi_{p_s} - \pi_{b_s}\| \geqslant \lambda$ for all s. Then for any $z \in \mathbb{R}^d$, $r > 0$ the number of subcurves π^s intersecting $B(z, r)$ is bounded by $\mathcal{O}(\kappa^d(1 + r/\lambda)^d)$.*

Proof. Let $\pi^{s_1}, \ldots, \pi^{s_\ell}$ be the subcurves that intersect the ball $B = B(z, r)$. Let $X = \{s_1, s_3, \ldots, \}$ be the odd indices among the intersecting subcurves. For all $s \in X$ pick any point π_{x_s} in $\pi^s \cap B$. Between any points $\pi_{x_s}, \pi_{x_{s'}}$ there must lie an even subcurve $\pi^{s_{2i}}$. As the endpoints of this even subcurve have distance at least λ, we have $\|\pi_{x_s} - \pi_{x_{s'}}\| \geqslant \lambda/(\kappa + 1)$. Otherwise the even part would not fit into $B(\pi_{x_s}, r) \cup B(\pi_{x_{s'}}, r)$ which has diameter $(\kappa + 1)\|\pi_{x_s} - \pi_{x_{s'}}\|$. Hence, the balls $B(\pi_{x_s}, \lambda/2(\kappa + 1))$ are disjoint and contained in $B(z, r + \lambda)$. A standard packing argument now shows that $\ell \leqslant 2 \cdot (r + \lambda)^d/(\lambda/2(\kappa + 1))^d = \mathcal{O}(\kappa^d(1 + r/\lambda)^d)$. $\square$

Lemma 3.10. *For any κ-bounded curves π, σ with n vertices in total, $0 < \varepsilon \leqslant 1$, we have $N(\pi, \sigma, \varepsilon) = \mathcal{O}((\kappa/\sqrt{\varepsilon})^d n)$.*

Proof. Let $\delta > 0$ and consider the partitionings into long segments and pieces $\pi^1, \ldots, \pi^k$, $\sigma^1, \ldots, \sigma^\ell$ computed by our algorithm. Then $\sigma^t = \sigma_{q_t..d_t}$ satisfies $\|\sigma_{q_t} - \sigma_{d_t}\| \geqslant \Lambda = \min\{\frac{1}{2}\sqrt{\varepsilon}, \frac{1}{4}\} \cdot \delta$ for all t. We use the same charging scheme as in Lemma 3.6. Consider any segment v of a piece π^s. The segment v can be charged by a part σ^t which is either a long segment or a piece. In both cases, σ^t intersects the ball B centered at the midpoint of $\|v\|$ with radius $r := (1+\varepsilon)\delta + 2\Lambda$. By Lemma 3.9 with $\lambda := \Lambda$, the number of such charges is bounded by $\mathcal{O}((\kappa/\sqrt{\varepsilon})^d)$.

Now consider any long segment v of π. The segment v can be charged by segments of σ which are longer than v. Any such charging gives rise to a long segment σ^t intersecting the ball B centered at the midpoint of v of radius $r := (1+\varepsilon)\delta + \frac{1}{2}\|v\|$. By Lemma 3.9 with $\lambda := \|v\|$, the number of such charges is bounded by $\mathcal{O}(\kappa^d(\frac{3}{2} + (1 + \varepsilon)\delta/\|v\|)^d) = \mathcal{O}((\kappa/\sqrt{\varepsilon})^d)$, since $\|v\| \geqslant \Lambda = \min\{\frac{1}{2}\sqrt{\varepsilon}, \frac{1}{4}\} \cdot \delta$.

Hence, every segment of π is charged $\mathcal{O}((\kappa/\sqrt{\varepsilon})^d)$ times; a symmetric statement holds for σ. $\square$

Plugging the above lemma into Lemma 2.1 we obtain the following result. The best previously known runtime was $\mathcal{O}((\kappa/\varepsilon)^d n + \kappa^d n \log n)$ [16].

Theorem 3.11. *For any $0 < \varepsilon \leqslant 1$ there is a $(1 + \varepsilon)$-approximation for the continuous and discrete Fréchet distance on κ-bounded curves with n vertices in total in time $\mathcal{O}((\kappa/\sqrt{\varepsilon})^d n \log^2 1/\varepsilon + \kappa^d n \log n) = \tilde{\mathcal{O}}((\kappa/\sqrt{\varepsilon})^d n)$.*

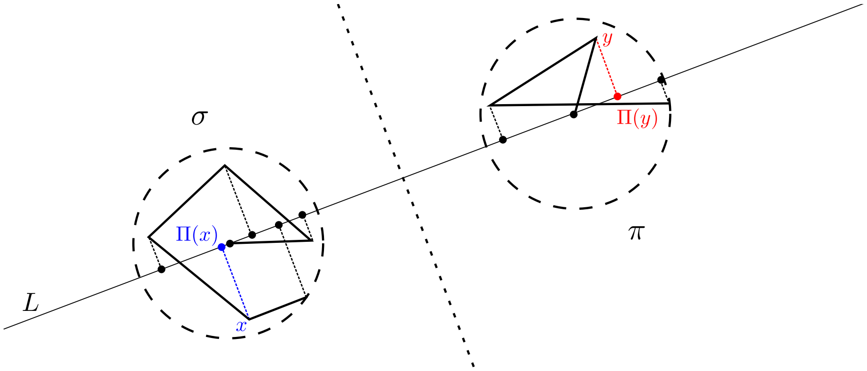


Figure 2: Projection of the pieces π, σ onto the line L through their initial vertices. This yields one-dimensional separated curves $\hat{\pi}, \hat{\sigma}$.

3.3 Solving the free-space region problem on pieces

It remains to prove Lemma 3.2. Let $(\pi, \sigma, \delta, \varepsilon)$ be an instance of the free-space region problem, where $n := |\pi|$, $m := |\sigma|$, with $\|\pi_1 - \pi_x\|, \|\sigma_1 - \sigma_y\| \leqslant \Lambda_{\varepsilon,\delta} = \Lambda$ for any $x \in [1, n]$, $y \in [1, m]$ (and entry intervals $\tilde{R}^h_{i,1} \subseteq [i, i+1] \times \{1\}$ for $i \in [1..n)$ and $\tilde{R}^v_{1,j} \subseteq \{1\} \times [j, j+1]$ for $j \in [1..m)$). We reduce this instance to the free-space region problem on *one-dimensional separated* curves, i.e., curves $\hat{\pi}, \hat{\sigma}$ in $\mathbb{R}$ such that all vertices of $\hat{\pi}$ lie above 0 and all vertices of $\hat{\sigma}$ lie below 0.

Since π and σ stay within distance Λ of their initial vertices, if their initial vertices are within distance $\|\pi_1 - \sigma_1\| \leqslant \delta - 2\Lambda$ then all pairs of points in π, σ are within distance δ. In this case, we find a translation of π making $\|\pi_1 - \sigma_1\| = \delta - 2\Lambda$ and all pairwise distances are still at most δ. This ensures that the curves π, σ are contained in disjoint balls of radius $\Lambda \leqslant \frac{1}{4}\delta$ centered at their initial vertices.

Consider the line L through the initial vertices π_1 and σ_1. Denote by $\Pi \colon \mathbb{R}^d \to L$ the projection onto L. Now, instead of the pieces π, σ we consider their projections $\hat{\pi} := \Pi(\pi) = (\Pi(\pi_1), \ldots, \Pi(\pi_n))$ and $\hat{\sigma} := \Pi(\sigma) = (\Pi(\sigma_1), \ldots, \Pi(\sigma_m))$, see Figure 2. Note that after rotation and translation we can assume that $\hat{\pi}$ and $\hat{\sigma}$ lie on $\mathbb{R} \subset \mathbb{R}^d$ and $\hat{\pi}$ and $\hat{\sigma}$ are separated by $0 \in \mathbb{R}$ (since π and σ are contained in disjoined balls centered on L). Now we solve the free-space region problem on $\hat{\pi}$, $\hat{\sigma}$, $\hat{\delta} := \delta$, and $\hat{\varepsilon} := \frac{1}{2}\varepsilon$ (with the same entry intervals $\tilde{R}^h_{i,j}, \tilde{R}^v_{i,j}$)).

Lemma 3.12. *Any solution to the the free-space region problem on $(\hat{\pi}, \hat{\sigma}, \hat{\delta}, \hat{\varepsilon})$ solves the free-space region problem on $(\pi, \sigma, \delta, \varepsilon)$.*

Proof. Let x, y be vertices of π, σ, respectively. Clearly, $\|\Pi(x) - \Pi(y)\| \leqslant \|x - y\|$. Hence, any monotone path in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$ yields a monotone path in $\mathcal{D}_{\leqslant\delta}(\hat{\pi}, \hat{\sigma}) = \mathcal{D}_{\leqslant\hat{\delta}}(\hat{\pi}, \hat{\sigma})$, so it will be found.

Note that x and y have distance at most Λ to L. Since $\Pi(x) - \Pi(y)$ and $x - \Pi(x) - (y - \Pi(y))$ are orthogonal, we can use the Pythagorean theorem to obtain

$$\|x - y\| = \sqrt{\|\Pi(x) - \Pi(y)\|^2 + \|x - \Pi(x) - (y - \Pi(y))\|^2} \leqslant \sqrt{\|\Pi(x) - \Pi(y)\|^2 + (2\Lambda)^2}.$$

Hence, any monotone path in $\mathcal{D}_{\leqslant(1+\hat{\varepsilon})\hat{\delta}}(\hat{\pi}, \hat{\sigma})$ yields a monotone path in $\mathcal{D}_{\leqslant\alpha}(\pi, \sigma)$ with $\alpha \leqslant \sqrt{(1+\hat{\varepsilon})^2\hat{\delta}^2 + (2\Lambda)^2}$. Plugging in $\hat{\delta} = \delta$, $\hat{\varepsilon} = \frac{1}{2}\varepsilon$, and $\Lambda = \min\{\frac{1}{2}\sqrt{\varepsilon}, \frac{1}{4}\} \cdot \delta$ we obtain $\alpha \leqslant \sqrt{(1+\frac{1}{2}\varepsilon)^2 + \varepsilon} \cdot \delta \leqslant (1+\varepsilon)\,\delta$. Thus, the desired guarantees for the free-space region problem are satisfied. □

We will show the following lemma in Section 4, concluding the proof of Lemma 3.2.

Lemma 3.13. *The free-space region problem on one-dimensional separated curves can be solved in time $\mathcal{O}((n+m)\log^2 1/\varepsilon)$.*

4 On one-dimensional separated curves

In this section, we show how to solve the free-space region problem on one-dimensional separated curves in time $\mathcal{O}((n+m)\log^2 1/\varepsilon)$, i.e., we prove Lemma 3.13.

First, in Section 4.1, we show how to reduce this problem to a *discrete* version, meaning that we can eliminate the continuous Fréchet distance and only consider the much simpler discrete Fréchet distance (for general curves such a reduction is not known to exist, but we only need it for one-dimensional separated curves). Moreover, we simplify our curves further by rounding the vertices. This yields a reduction to the following subproblem. Note that we no longer ask for an approximation algorithm.

Problem 4.1 (Reduced free-space problem). Given one-dimensional separated curves π, σ with n, m vertices and all vertices being multiples of $\frac{1}{3}\varepsilon\delta$, and given an entry set $E \subseteq [1..n]$, compute the exit set $F^\pi \subseteq [1..n]$ consisting of all points f such that $d_{\mathrm{dF}}(\pi_{e..f}, \sigma) \leqslant \delta$ for some $e \in E$ and the exit set $F^\sigma \subseteq [1..m]$ consisting of all points f such that $d_{\mathrm{dF}}(\pi_{e..n}, \sigma_{1..f}) \leqslant \delta$ for some $e \in E$.

Lemma 4.2. *The reduced free-space problem can be solved in time $\mathcal{O}((n+m)\log 1/\varepsilon)$.*

As a second step, we prove the above lemma. We first consider the special case of $E = \{1\}$ and the problem of deciding whether $n \in F^\pi$, i.e., the lower left corner $(1,1)$ of the free-space is the only entry point and we want to determine whether the upper right corner (n, m) is an exit. This is equivalent to deciding whether the discrete Fréchet distance of π, σ is at most δ, which is known to have a near-linear time algorithm as π, σ are one-dimensional and separated (see the footnote in the introduction for details). We present a greedy algorithm for this special case in Section 4.2. To extend this to the reduced free-space problem, we prove useful structural properties of one-dimensional separated curves in Section 4.3. With these, we first solve the problem of determining the exit set F^π assuming $E = \{1\}$ in Section 4.4.1. Then we show for general $E \subseteq [1..n]$ how to compute F^π (Section 4.4.2) and F^σ (Section 4.4.3).

4.1 Reduction from the continuous to the discrete case

Essentially we use the following lemma to reduce the *continuous* free-space region problem on one-dimensional separated curves to the *discrete* reduced free-space problem.

Lemma 4.3. *Let π, σ be one-dimensional separated curves with subcurves $\pi_{p..b}, \sigma_{q..d}$. Then we have $d_{\mathrm{F}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{dF}}(\pi_{p..b}, \sigma_{q..d})$. In particular, assume that we subdivide any segments of π, σ by adding new vertices, which yields new curves π', σ' with subcurves $\pi'_{p'..b'}, \sigma'_{q'..d'}$ that are subdivisions of $\pi_{p..b}, \sigma_{q..d}$. Then we have $d_{\mathrm{dF}}(\pi'_{p'..b'}, \sigma'_{q'..d'}) = d_{\mathrm{dF}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{F}}(\pi_{p..b}, \sigma_{q..d})$.*

Proof. It is known that $d_{\mathrm{F}}(\pi, \sigma) \leqslant d_{\mathrm{dF}}(\pi, \sigma)$ holds for all curves π, σ. Thus, we only need to show that any continuous traversal $\phi = (\phi_1, \phi_2)$ of $\pi_{p..b}, \sigma_{q..d}$ can be transformed into a discrete traversal with the same width. We adapt ϕ as follows. For any point in time $t \in [0,1]$, if $\phi_1(t)$ is at a vertex of π we set $\phi'_1(t) := \phi_1(t)$. Otherwise $\phi_1(t)$ is in the interior of a segment $\pi_{i..i+1}$ of π. Let $j \in \{i, i+1\}$ minimize π_j. We set $\phi'_1(t) := j$. Observe that ϕ'_1 indeed is a non-decreasing function from $[0,1]$ onto $[1..n]$. A similar construction, where we round to the value $j \in \{i, i+1\}$ maximizing σ_j, yields ϕ'_2 and we obtain a discrete traversal $\phi' = (\phi'_1, \phi'_2)$. The width of ϕ' is at most the width of ϕ since we rounded in the right way, i.e., we have $\pi(\phi'_1(t)) \leqslant \pi(\phi_1(t))$ and $\sigma(\phi'_2(t)) \geqslant \sigma(\phi_2(t))$ so that $\|\pi(\phi'_1(t)) - \sigma(\phi'_2(t))\| \leqslant \|\pi(\phi_1(t)) - \sigma(\phi_2(t))\|$ for all $t \in [0, 1]$.

Note that the discrete Fréchet distance is in general not preserved under subdivision of segments, but the continuous Fréchet distance is. Thus, the second statement follows from the first one, $d_{\mathrm{dF}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{F}}(\pi_{p..b}, \sigma_{q..d}) = d_{\mathrm{F}}(\pi'_{p'..b'}, \sigma'_{q'..d'}) = d_{\mathrm{dF}}(\pi'_{p'..b'}, \sigma'_{q'..d'})$. □

The above lemma allows the following trick. Consider any finite sets $E \subseteq [1, n]$ and $F \subseteq [1, n]$. Add π_x as a vertex to π for any $x \in E \cup F$, with slight abuse of notation we say that π now has vertices at π_i, $i \in [1..n]$, and π_x, $x \in E \cup F$. Mark the vertices π_x, $x \in E$, as entries. Now solve the reduced free-space problem instance (π, σ, E). This yields the set F^π of all values $f \in F$ such that there is an $e \in E$ with $d_{\mathrm{dF}}(\pi_{e..f}, \sigma) \leqslant \delta$, which by Lemma 4.3 is equivalent to $d_{\mathrm{F}}(\pi_{e..f}, \sigma) \leqslant \delta$. Thus, we computed all exit points in F given entry points in E, with respect to the continuous Fréchet distance. This is already near to a solution of the free-space region problem, however, we have to cope with entry and exit *intervals*.

For the full reduction we need two more arguments. First, we can replace all non-empty input intervals $\tilde{R}^h_{i,1}$ by the leftmost point $(y_i, 1)$ in $\tilde{R}^h_{i,1} \cap \mathcal{D}_{\leqslant\delta}(\pi, \sigma)$, specifically, we show that any traversal starting in a point in $\tilde{R}^h_{i,1}$ can be transformed into a traversal starting in $(y_i, 1)$. Thus, we add π_{y_i} as a vertex and mark it as an entry to obtain a finite and small set of entry points. Second, for any segment $\pi_{i..i+1}$ we call a point $f \in [i, i+1]$ *reachable* if there is an $e \in E$ with $d_{\mathrm{F}}(\pi_{e..f}, \sigma) \leqslant \delta$. We show that if f is reachable then essentially all points $f' \in [i, i+1]$ with $\pi_{f'} \leqslant \pi_f$ are also reachable. Thus, the set of reachable points is an interval with one trivial endpoint, and we only need to search for the other endpoint of the interval, which can be done by binary search. Moreover, we can parallelize all these binary searches, as solving one reduced free-space problem can answer for every segment of π whether a particular point on this segment is reachable (after adding this point as a vertex). To make these binary searches finite, we round all vertices of π and σ to multiples of $\gamma := \frac{1}{3}\varepsilon\delta$ and only search for exit points that are multiples of γ. This is allowed since the free-space region problem only asks for an approximate answer. A similar procedure yields the exits on σ reachable from entries on π, and determining the exits reachable from entries on σ is a symmetric problem. Since for the binary searches we reduce to $\mathcal{O}(\log 1/\varepsilon)$ instances of the reduced free-space problem, Lemma 3.13 follows from Lemma 4.2.

In the following we present the details of this approach. Let π, σ be one-dimensional separated curves, i.e., they are contained in $\mathbb{R}$, all vertices of π lie above 0, and all vertices of σ lie below 0. Let $n = |\pi|$, $m = |\sigma|$, $\delta > 0$ and $0 < \varepsilon \leqslant 1$. Consider entry intervals $\tilde{R}^h_{i,1} \subseteq [i, i+1] \times \{1\}$ for $i \in [1..n)$ and $\tilde{R}^v_{1,j} \subseteq \{1\} \times [j, j+1]$ for $j \in [1..m)$. We reduce this instance of the free-space region problem to $\mathcal{O}(\log 1/\varepsilon)$ instances of the reduced free-space problem.

First we change π, σ as follows. (1) Let $Z \subset \mathbb{R}$ be the set of all integral multiples[4] of $\gamma := \frac{1}{3}\varepsilon\delta$. We round all vertices of π, σ to values in Z, where we round down everything in π and round up in σ, yielding curves π', σ'. (2) Let $I \subseteq [1..n)$ be the set of all i with nonempty $\tilde{R}^h_{i,1} \cap \mathcal{D}_{\leqslant\delta}(\pi', \sigma')$. For any $i \in I$ let $(y_i, 1)$ be the leftmost point in $\tilde{R}^h_{i,1} \cap \mathcal{D}_{\leqslant\delta}(\pi', \sigma')$ and note that π'_{y_i} is also a multiple of γ. Add π'_{y_i} as a vertex to π' and mark it as an entry. With slight abuse of notation, we say that π' now has its vertices at π'_i, $i \in [1..n]$ and π'_{y_i}, $i \in I$. We let $E = \{y_i \mid i \in I\}$ be the indices of the entry vertices. Note that (π', σ', E) can be computed in time $\mathcal{O}(n + m)$.

For every $i \in [1..n)$ consider the multiples of γ on $\pi'_{i..i+1}$, i.e., $S_i := \{x \in [i, i+1] \mid \pi'_x \in Z\}$. Note that S_i forms an arithmetic progression, specifically $S_i = \{i, i + 1/t_i, i + 2/t_i, \ldots, i + 1\}$ for some $t_i \in \mathbb{N}$, since π'_i, π'_{i+1} are in Z and π'_x is a linear function in x. Thus, S_i and subsequences of S_i can be handled efficiently, we omit these details in the following. We want to determine the set

[4] Without loss of generality we assume $1/\varepsilon \in \mathbb{N}$ so that $\delta \in Z$.

F_i of all $f \in S_i$ such that there is an $e \in E$ with $d_{\mathrm{dF}}(\pi'_{e..f}, \sigma') \leqslant \delta$. We first argue that F_i is of an easy form.

Lemma 4.4. *If F_i is non-empty then we have $F_i = [a,b] \cap Z$ for some $a, b \in S_i$ with $\{a,b\} \cap \{i, y_i, i+1\} \neq \emptyset$ (or $\{a,b\} \cap \{i, i+1\} \neq \emptyset$ if y_i does not exist).*

Proof. We show that if any $f \in S_i$ is reachable, i.e., there is an $e \in E$ with $d_{\mathrm{dF}}(\pi'_{e..f}, \sigma') \leqslant \delta$, then any $f' \in S_i$ with $\pi'_{f'} \leqslant \pi'_f$ and $y_i \notin (f', f]$ is also reachable. This proves the claim. Let ϕ be any traversal of $\pi'_{e..f}, \sigma'$ of width at most δ. Note that $e \leqslant f'$, since $y_i \notin (f', f]$ and y_i is the only entry on the segment containing f and f'. If $f' \leqslant f$ then we change ϕ to stop at $\pi'_{f'}$ once it arrives at this point, and we traverse the remaining part of σ staying fixed at $\pi'_{f'}$. Since $\pi'_{f'} \leqslant \pi'_f$ this does not increase the width of the traversal and shows that f' is also reachable. If $f' > f$ then we append a traversal to ϕ that stays fixed at σ'_m but walks in π' from π'_f to $\pi'_{f'}$. Again since $\pi'_{f'} \leqslant \pi'_f$ this does not increase the width of the traversal and shows that f' is also reachable. □

Note that by solving the reduced free-space problem on (π', σ', E) we decide for each $f \in [n] \cup \{y_i \mid i \in I\}$ whether there is an $e \in E$ with $d_{\mathrm{dF}}(\pi'_{e..f}, \sigma') \leqslant \delta$. By the above lemma, this yields one of the endpoints of the interval F_i, say a, and we only have to determine the other endpoint, say b. In the special case $\pi'_i = \pi'_{i+1}$ we even determined both endpoints already, so from now on we can assume $\pi'_i \neq \pi'_{i+1}$ so that $|S_i| < \infty$. We search for the other endpoint of F_i using a binary search over S_i. To test whether any $z \in S_i$ is in F_i, we add π'_z as a vertex of π' and solve the reduced free-space problem on (π', σ', E). If z is in the output set F^π then it is in F_i.

Note that any vertex $\pi'_x > \delta$ on π' does not have any point of σ within distance δ, which is preserved by setting $\pi'_x := 2\delta$. Thus, we can assume that π' takes values in $[0, 2\delta]$, which implies $|S_i| \leqslant \mathcal{O}(1/\varepsilon)$, so that our binary search needs $\mathcal{O}(\log 1/\varepsilon)$ steps. Moreover, note that we can parallelize these binary searches, since we can add a vertex z_i on every subcurve $\pi'_{i..i+1}$, so that one call to the reduced free-space problem determines for every z_i whether it is reachable. Here we use Lemma 4.3, since we need that further subdivision of some segments of π' does not change the discrete Fréchet distance. Note that since we add $\mathcal{O}(n)$ vertices to π' and since we need $\mathcal{O}(\log 1/\varepsilon)$ steps of binary search, Lemma 4.2 implies a total runtime of $\mathcal{O}((n+m)\log^2 1/\varepsilon)$.

We thus computed $F_i = [a,b] \cap Z$ with $a, b \in S_i$. We extend F_i slightly to $F'_i = [a', b'] \cap Z$ by including the neighboring elements of a and b in S_i. Finally, we set $\tilde{R}^h_{i,m}(\pi) := [a', b'] \times \{m\}$. A similar procedure adding entries E^σ on σ' and doing a binary search over exits on π' yields an interval $\tilde{R}^h_{i,m}(\sigma)$ consisting of points $(f, m) \in [i, i+1] \times \{m\}$ such that there is an $e \in E^\sigma$ with $d_{\mathrm{dF}}(\pi'_{1..f}, \sigma'_{e..m}) \leqslant \delta$. We set $\tilde{R}^h_{i,m} := \tilde{R}^h_{i,m}(\pi) \cup \tilde{R}^h_{i,m}(\sigma)$, which will be again an interval (which follows from the proof of Lemma 4.4). A symmetric algorithm determines $\tilde{R}^v_{n,j}$ for $j \in [1..m)$.

We show that we correctly solve the given free-space region problem instance.

Lemma 4.5. *The computed intervals are a valid solution to the given free-space region instance.*

Proof. Let ϕ be any monotone path in $\mathcal{D}_{\leqslant\delta}(\pi, \sigma)$ that starts in a point $(p, 1) \in \tilde{R}^h_{j,1}$ and ends in (b, m), witnessing that $d_{\mathrm{dF}}(\pi_{p..b}, \sigma) \leqslant \delta$. After rounding down π to π' and rounding up σ to σ', ϕ is still a monotone path in $\mathcal{D}_{\leqslant\delta}(\pi', \sigma')$. Moreover, we can prepend a path from $(y_j, 1)$ to $(p, 1)$ to ϕ, since $\tilde{R}^h_{j,1} \cap \mathcal{D}_{\leqslant\delta}(\pi', \sigma')$ is an interval containing $(y_j, 1)$ and $(p, 1)$. Let r be the value of π_b rounded down to a multiple of γ. This value r is attained at some point π_f on the same segment $\pi_{i..i+1}$ as π_b. If $f \leqslant b$ then we change ϕ to stop at π_f whenever it reaches this point. If $f > b$ then we change ϕ by appending a path from (b, m) to (f, m). In any case, this yields a monotone path in $\mathcal{D}_{\leqslant\delta}(\pi', \sigma')$ from $(y_j, 1)$ to (f, m). Since such a continuous traversal is equivalent to a discrete traversal by Lemma 4.3, we have $f \in F_i$. By the construction of F'_i, the point (b, m) will be contained in the output $\tilde{R}^h_{i,m}(\pi)$, so we find the reachable exit (b,m) as desired. A similar argument with entries on σ shows that we satisfy property (1) of the free-space region problem.

Consider any point (f,m) in the output set $\tilde{R}^h_{i,m}(\pi)$. By the construction of F'_i, there is a point b on the same segment as f with $\|\pi'_b - \pi'_f\| \leqslant \gamma$ and there is an entry $e \in E$ with $d_{\mathrm{dF}}(\pi'_{e..b}, \sigma') \leqslant \delta$, witnessed by a traversal ϕ. If $b \leqslant f$ we change ϕ so that it stops at π'_b once it reaches this point. If $b > f$ we change ϕ by appending a path from (b,m) to (f,m). In any case, this shows $d_{\mathrm{dF}}(\pi'_{e..f}, \sigma') \leqslant \delta + \gamma$. Since π', σ' are rounded versions of π, σ where all vertices are moved by less than γ, we obtain $d_{\mathrm{dF}}(\pi_{e..f}, \sigma) \leqslant \delta + 3\gamma = (1+\varepsilon)\delta$. Thus, any point (f,m) in the output set is reachable form the entry sets by a monotone path in $\mathcal{D}_{\leqslant(1+\varepsilon)\delta}(\pi,\sigma)$, which together with a similar argument for entries on σ proves that we satisfy property (2) of the free-space problem. □

4.2 Greedy Decider for the Fréchet Distance of One-Dimensional Separated Curves

In the remainder of the paper all indices of curves will be integral. Let $\pi = (\pi_1, \ldots, \pi_n)$ and $\sigma = (\sigma_1, \ldots, \sigma_m)$ be two separated polygonal curves in $\mathbb{R}$, i.e., $\pi_i \geqslant 0 \geqslant \sigma_j$. For indices $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$, define $\mathrm{vis}_\sigma(i,j) := \{k \mid k \geqslant j \text{ and } \sigma_k \geqslant \pi_i - \delta\}$ as the index set of vertices on σ that are later in sequence than σ_j and are still in distance δ to π_i (i.e, *seen* by π_i) and, likewise, $\mathrm{vis}_\pi(i,j) := \{k \mid k \geqslant i \text{ and } \pi_k \leqslant \sigma_j + \delta\}$. Hence, the set of points that we may reach on σ by starting in (π_i, σ_j) and staying in π_i can be defined as the longest contiguous subsequence $[j+1..j+k]$ such that $[j+1..j+k] \subseteq \mathrm{vis}_\sigma(i,j)$. Let $\mathrm{reach}_\sigma(i,j) := [j+1..j+k]$ denote this subsequence and let $\mathrm{reach}_\pi(i,j)$ be defined symmetrically. Note that $\pi_i \leqslant \pi_{i'}$ implies that $\mathrm{vis}_\sigma(i,j) \supseteq \mathrm{vis}_\sigma(i',j)$, however the converse does not necessarily hold. Also, $\mathrm{vis}_\sigma(i,j) \not\supseteq \mathrm{vis}_\sigma(i',j)$ implies that $\mathrm{vis}_\sigma(i,j) \subsetneq \mathrm{vis}_\sigma(i',j)$ and $\pi_i > \pi_{i'}$.

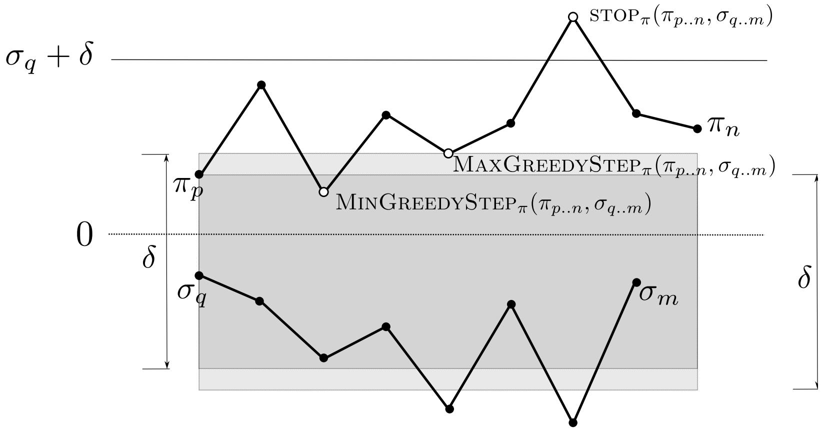
The visibility sets established above enable us to define a greedy algorithm for the Fréchet distance of π and σ. Let $1 \leqslant p \leqslant n$ and $1 \leqslant q \leqslant m$ be arbitrary indices on σ and π. We say that p' is a *greedy step on π from (p,q)*, written $p' \leftarrow \textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$, if $p' \in \mathrm{reach}_\pi(p,q)$ and $\mathrm{vis}_\sigma(i,q) \subseteq \mathrm{vis}_\sigma(p',q)$ holds for all $p \leqslant i \leqslant p'$. Symmetrically, $q' \in \mathrm{reach}_\sigma(p,q)$ is a *greedy step on σ from (p,q)*, if $\mathrm{vis}_\pi(p,i) \subseteq \mathrm{vis}_\pi(p,q')$ for all $q \leqslant i \leqslant q'$. In pseudo code, $\textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$ denotes a function that returns an arbitrary greedy step p' on π from (p,q) if such an index exists and returns an error otherwise (symmetrically for σ). See Figure 3.

Consider the following greedy algorithm:

Algorithm 1 Greedy algorithm for the Fréchet distance of separated curves $\pi_{1..n}$ and $\sigma_{1..m}$ in $\mathbb{R}$

```
$p \leftarrow 1, q \leftarrow 1$
repeat
    if $p' \leftarrow \textsc{GreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$ then
        $p \leftarrow p'$
    if $q' \leftarrow \textsc{GreedyStep}_\sigma(\pi_{p..n}, \sigma_{q..m})$ then
        $q \leftarrow q'$
until no greedy step was found in the last iteration
if $p = n$ and $q = m$ then return $d_{\mathrm{dF}}(\pi,\sigma) \leqslant \delta$
else return $d_{\mathrm{dF}}(\pi,\sigma) > \delta$
```

Theorem 4.6. *Let π and σ be separated curves in $\mathbb{R}$ and $\delta > 0$. Algorithm 1 decides whether $d_{\mathrm{dF}}(\pi,\sigma) \leqslant \delta$ in time $\mathcal{O}((n+m)\log(nm))$.*



$\text{STOP}_\pi(\pi_{p..n}, \sigma_{q..m})$
$\sigma_q + \delta$
π_n
$\text{MaxGreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$
π_p
$\text{MinGreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$
0
δ
σ_q
σ_m
δ

Figure 3: An illustration of greedy steps. For better visibility, the one-dimensional separated curves π, σ are drawn in the plane by mapping π_i to (i, π_i). In particular, the results of $\text{MinGreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$, $\text{MaxGreedyStep}_\pi(\pi_{p..n}, \sigma_{q..m})$, and $\text{STOP}_\pi(\pi_{p..n}, \sigma_{q..m})$ are shown.

We will first prove the correctness of the algorithm in Lemma 4.8 below and postpone the discussion how to implement the algorithm efficiently to Section 4.2.2.

4.2.1 Correctness

Note that Algorithm 1 considers potentially only very few points of the curve explicitly during its execution. Call the indices (p, q) of point pairs considered in some iteration of the algorithm (for any choice of greedy steps, if more than one exists) *greedy (point) pairs* and all points contained in some such pair *greedy points (of π and σ).* The following useful monotonicity property holds: If some greedy point on π sees a point on σ that is yet to be traversed, all following greedy points on π will see it *until it is traversed.*

Lemma 4.7. *Let $(p_1, q_1), \ldots, (p_i, q_i)$ be the greedy point pairs considered in the iterations $1, \ldots, i$. It holds that*

1. $\text{vis}_\sigma(\ell, q_i) \subseteq \text{vis}_\sigma(p_i, q_i)$ *for all* $1 \leqslant \ell \leqslant p_i$*, and*
2. $\text{vis}_\pi(p_i, \ell) \subseteq \text{vis}_\pi(p_i, q_i)$ *for all* $1 \leqslant \ell \leqslant q_i$*.*

Proof. Let $k < i$. We first show that $\text{vis}_\sigma(\ell, q_i) \subseteq \text{vis}_\sigma(p_{k+1}, q_i)$ holds for all $p_k \leqslant \ell < p_{k+1}$. If $p_k = p_{k+1}$, the claim is immediate. Otherwise p_{k+1} is the result of a greedy step on π. By definition of visibility, we have $\text{vis}_\sigma(\ell, q_i) = \text{vis}_\sigma(\ell, q_k) \cap [q_i..m] \subseteq \text{vis}_\sigma(p_{k+1}, q_k) \cap [q_i..m] = \text{vis}_\sigma(p_{k+1}, q_i)$, where the inequality follows from p_{k+1} being a greedy step from (p_k, q_k).

For arbitrary $\ell \leqslant i$, let $k < i$ be such that $p_k \leqslant \ell < p_{k+1}$. Then $\text{vis}_\sigma(\ell, q_i) \subseteq \text{vis}_\sigma(p_{k+1}, q_i) \subseteq \text{vis}_\sigma(p_{k+2}, q_i) \subseteq \cdots \subseteq \text{vis}_\sigma(p_i, q_i)$. The second statement is symmetric. □

We will exploit this monotonicity to prove that if Algorithm 1 finds a greedy point pair that allows no further greedy steps, then no feasible traversal of π and σ exists. We derive an even stronger statement using the following notion: For a greedy point pair (p, q), define $\text{STOP}_\pi(\pi_{p..n}, \sigma_{q..m}) := \max(\text{reach}_\pi(p, q) \cup \{p\}) + 1$ as the index of the first point after π_p on π which is not seen by σ_q, or $n + 1$ if no such index exists. Let STOP_σ be defined symmetrically.

Lemma 4.8 (Correctness of Algorithm 1)**.** *Let (p, q) be a greedy point of π and σ, $p_{\text{stop}} :=$ $\text{STOP}_\pi(\pi_{p..n}, \sigma_{q..m})$ and $q_{\text{stop}} := \text{STOP}_\sigma(\pi_{p..n}, \sigma_{q..m})$. If on both curves, no greedy step from (p, q) exists, then $d_{\text{dF}}(\pi, \sigma) > \delta$.*

In particular, if $q_{\mathrm{stop}} < m$, then for all $1 \leqslant p' \leqslant n$, we have that $d_{\mathrm{dF}}(\pi_{1..p'}, \sigma_{1..q_{\mathrm{stop}}}) > \delta$ and if $p_{\mathrm{stop}} < n$, then $d_{\mathrm{dF}}(\pi_{1..p_{\mathrm{stop}}}, \sigma_{1..q'}) > \delta$ for all $1 \leqslant q' \leqslant m$.

Note that the correctness of Algorithm 1 follows immediately: If the algorithm is stuck, then $d_{\mathrm{dF}}(\pi, \sigma) > \delta$. Otherwise, it finds a feasible traversal.

Proof of Lemma 4.8. Consider the case that no greedy step from (p, q) exists, then the following *stuckness conditions* have to hold:

1. For all $p' \in \mathrm{reach}_\pi(p, q)$, we have $\mathrm{vis}_\sigma(p', q) \subsetneq \mathrm{vis}_\sigma(p, q)$, and
2. for all $q' \in \mathrm{reach}_\sigma(p, q)$, we have $\mathrm{vis}_\pi(p, q') \subsetneq \mathrm{vis}_\pi(p, q)$.

In this case, we can extend the monotonicity property of Lemma 4.7 to include all reachable and the first unreachable point.

Claim 4.9. *If the stuckness conditions hold for (p, q), then we have $\mathrm{vis}_\sigma(i, q) \subseteq \mathrm{vis}_\sigma(p, q)$ for all $1 \leqslant i \leqslant p_{\mathrm{stop}}$. In particular, if π_p does not see σ_ℓ for some $\ell > q$, then no vertex π_i with $1 \leqslant i \leqslant p_{\mathrm{stop}}$ sees σ_ℓ. The symmetric statement holds for σ.*

Proof. By the monotonicity of the previous claim, $\mathrm{vis}_\sigma(i, q) \subseteq \mathrm{vis}_\sigma(p, q)$ holds for all $i \leqslant p$. The first of the stuckness conditions implies $\mathrm{vis}_\sigma(i, q) \subseteq \mathrm{vis}_\sigma(p, q)$ for all $p < i < p_{\mathrm{stop}}$. If $p_{\mathrm{stop}} = n + 1$, this already completes the proof of the claim. Otherwise, note that $\pi_{p_{\mathrm{stop}}} > \pi_p$, since otherwise $p_{\mathrm{stop}} \in \mathrm{reach}_\pi(p, q)$. Hence $\mathrm{vis}_\sigma(p_{\mathrm{stop}}, q) \subseteq \mathrm{vis}_\sigma(p, q)$ holds as well. □

We distinguish the following cases that may occur under the stuckness conditions:

Case 1: $p_{\mathrm{stop}} \leqslant n$ or $q_{\mathrm{stop}} \leqslant m$. Without loss of generality, let $p_{\mathrm{stop}} \leqslant n$ (the other case is symmetric). Assume for contradiction that a feasible traversal ϕ of $\pi_{1..p_{\mathrm{stop}}}$ and $\sigma_{1..q'}$ exists for some $1 \leqslant q' \leqslant m$. In ϕ, at some point in time we have to move in π from $p_{\mathrm{stop}} - 1$ to p_{stop} while moving in $\sigma_{1..q'}$ from some $\sigma_{\ell'}$ to σ_ℓ where $\ell' \in \{\ell - 1, \ell\}$ and σ_ℓ sees $\pi_{p_{\mathrm{stop}}}$. Since σ_q does not see $\pi_{p_{\mathrm{stop}}}$, the previous claim shows that $\ell > q_{\mathrm{stop}}$. If $q_{\mathrm{stop}} = m + 1$ or $q_{\mathrm{stop}} < q'$, this is impossible, yielding a contradiction. Otherwise, to do this transition, in some earlier step we have to move in σ from $q_{\mathrm{stop}} - 1$ to q_{stop} while moving in π from $\pi_{k'}$ to π_k for some $k < p_{\mathrm{stop}}$ and $k' \in \{k - 1, k\}$. However, by definition $q_{\mathrm{stop}} \notin \mathrm{vis}_\sigma(p, q)$, hence Claim 4.9 implies that the transition is illegal, since π_k does not see $\sigma_{q_{\mathrm{stop}}}$. This is a contradiction. By a symmetric argument, it holds that $d_{\mathrm{dF}}(\pi_{1..p'}, \sigma_{1..q_{\mathrm{stop}}}) > \delta$.

Case 2: $p_{\mathrm{stop}} = n + 1$ and $q_{\mathrm{stop}} = m + 1$. In this case, $\mathrm{reach}_\pi(p, q) = [p + 1..n]$ and $\mathrm{reach}_\sigma(p, q) = [q + 1..n]$. By stuckness conditions, there exist an index $p_{\max} > p$ such that no $\sigma_{q'}$ with $q' > q$ sees $\pi_{p_{\max}}$ and an index $q_{\min}$ such that no $\pi_{p'}$ with $p' > p$ sees $\sigma_{q_{\min}}$. Assume for contradiction that a feasible traversal ϕ exists. In ϕ, at some point in time t, we have to cross either (1) from π_p to π_{p+1} while moving in σ from $\sigma_{\ell'}$ to σ_ℓ with $\ell \leqslant q + 1 \leqslant q_{\min}$ and $\ell' \in \{\ell - 1, \ell\}$ or (2) from σ_q to σ_{q+1} while moving from $\pi_{\ell'}$ to π_ℓ with $\ell \leqslant p + 1 \leqslant p_{\max}$ and $\ell' \in \{\ell - 1, \ell\}$. In the first case, $\ell < q_{\min}$ holds, since π_{p+1} does not see $\sigma_{q_{\min}}$. For all consecutive times $t' \geqslant t$, ϕ is in a point $\pi_{p'}$ $(p' \geqslant p + 1)$ that does not see $\sigma_{q_{\min}}$, which still has to be traversed, leading to a contradiction. Symmetrically, in the second case, for all times $t' \geqslant t$, ϕ is in a point $\sigma_{q'}$ $(q' \geqslant q + 1)$ that does not see $\pi_{p_{\max}}$, which still has to be traversed.

This concludes the proof of Lemma 4.8. □

4.2.2 Implementing greedy steps

To prove Theorem 4.6, it remains to show how to implement the algorithm to run in time $\mathcal{O}((n+m)\log(nm))$. We make use of geometric range search queries. The classic technique of fractional cascading [13, 23, 26] provides a data structure D with the following properties: (i) Given n points $\mathcal{P}$ in the plane, $D(\mathcal{P})$ can be constructed in time $\mathcal{O}(n \log n)$ and (ii) given a query rectangle $Q := I_1 \times I_2$ with intervals I_1 and I_2, find and return $q \in Q \cap \mathcal{P}$ with minimal y-coordinate, or report that no such point exists, in time $\mathcal{O}(\log n)$. Here, each interval I_i may be open, half-open or closed.

By invoking the above data structure on $\mathcal{P} := \{(i, \pi_i) \mid i \in [1 \ldots n]\}$ for a given curve $\pi = \pi_{1..n}$ (as well as all three rotations of $\mathcal{P}$ by multiples of 90°), we obtain a datastructure D^π such that:

1. D^π can be constructed in time $\mathcal{O}(n \log n)$,
2. the query $\mathrm{D}^\pi.\textsc{minIndex}([x_1, x_2], [p, b])$ ($\mathrm{D}^\pi.\textsc{maxIndex}([x_1, x_2], [p, b])$) returns the minimum (maximum) index $p \leqslant i \leqslant b$ such that $x_1 \leqslant \pi_i \leqslant x_2$ in time $\mathcal{O}(\log n)$, and
3. the query $\mathrm{D}^\pi.\textsc{minHeight}([x_1, x_2], [p, b])$ ($\mathrm{D}^\pi.\textsc{maxHeight}([x_1, x_2], [p, b])$) returns the minimum (maximum) height $x_1 \leqslant \pi_i \leqslant x_2$ such that $p \leqslant i \leqslant b$ in time $\mathcal{O}(\log n)$.

The queries extend naturally to open and half-open intervals. If no index exists in the queried range, all of these operations return the index ∞. We will use the corresponding data structure D^σ for σ as well.

With these tools, we implement the following basic operations for arbitrary subcurves $\pi' := \pi_{p..b}$ and $\sigma' := \sigma_{q..d}$ of π and σ. See also Figure 3.

1. **Stopping points stop$_\pi(\pi', \sigma')$.** For points p, q, $\textsc{stop}_\pi(\pi', \sigma') := \max(\mathrm{reach}_{\pi'}(p, q) \cup \{p\}) + 1$ returns the index of the first point after π_p on π' which is not seen by σ_q, or $b + 1$ if no such index exists.

Algorithm 2 Finding the stopping point

```
function $\textsc{stop}_\pi(\pi_{p..b}, \sigma_{q..d})$
    $p_{\text{stop}} \leftarrow \mathrm{D}^\pi.\textsc{minIndex}((\sigma_q + \delta, \infty), [p, b])$        ▷ First non-visible point on $\pi$
    if $p_{\text{stop}} < \infty$ then return $p_{\text{stop}}$
    else return $b + 1$
```

2. **Minimal greedy steps MinGreedyStep$_\pi(\pi', \sigma')$.** This function returns the smallest index $p' \in \mathrm{reach}_{\pi'}(p, q)$ such that $\mathrm{vis}_{\sigma'}(p', q) \supseteq \mathrm{vis}_{\sigma'}(p, q)$ or reports that no such index exists.

Algorithm 3 Minimal greedy step

```
function $\textsc{MinGreedyStep}_\pi(\pi_{p..b}, \sigma_{q..d})$
    $q_{\min} \leftarrow \mathrm{D}^\sigma.\textsc{minHeight}([\pi_p - \delta, \infty), [q, d])$        ▷ Lowest still visible point on $\sigma$
    $p_{\text{cand}} \leftarrow \mathrm{D}^\pi.\textsc{minIndex}((-\infty, \sigma_{q_{\min}} + \delta], [p + 1, d])$        ▷ If $p'$ exists, it is $p_{\text{cand}}$
    $p_{\text{stop}} \leftarrow \textsc{stop}_\pi(\pi_{p..b}, \sigma_{q..d})$        ▷ First non-visible point on $\pi$
    if $p_{\text{cand}} < p_{\text{stop}}$ then return $p_{\text{cand}}$
    else return "No greedy step possible."  ▷ $\pi_{p_{\text{cand}}}$ not reachable from $\pi_p$ while staying in $\sigma_q$
```

3. **Maximal greedy steps MaxGreedyStep$_\pi(\pi', \sigma')$.** Let $p' \in \mathrm{reach}_{\pi'}(p, q)$ be such that (i) p' is the largest index maximizing $|\mathrm{vis}_{\sigma'}(z, q)|$ among all $z \in \mathrm{reach}_{\pi'}(p, q)$ and (ii) $\mathrm{vis}_{\sigma'}(p', q) \supseteq$

$\mathrm{vis}_{\sigma'}(p,q)$. If p' exists, $\textsc{MaxGreedyStep}_\pi$ returns this value, otherwise it reports that no such index exists. Note that if p' exists, then by definition there is no greedy step on π starting from (p',q), i.e., this step is a maximal greedy step.

Algorithm 4 Maximal greedy step

```
function $\textsc{MaxGreedyStep}_\pi(\pi_{p..b}, \sigma_{q..d})$
    $q_{\min} \leftarrow \mathrm{D}^\sigma.\textsc{minHeight}([\pi_p - \delta, \infty), [q,d])$    ▷ Lowest still visible point on $\sigma$
    $p_{\mathrm{stop}} \leftarrow \textsc{stop}_\pi(\pi_{p..b}, \sigma_{q..d})$    ▷ First non-visible point on $\pi$
    $p_{\min} \leftarrow \mathrm{D}^\pi.\textsc{minHeight}((-\infty, \sigma_{q_{\min}} + \delta], [p+1, p_{\mathrm{stop}} - 1])$    ▷ Maximizes visibility among reachable points
    if $p_{\min} = \infty$ then
        return "No greedy step possible."    ▷ No reachable point has better visibility than $\pi_p$
    else
        $q_{\min} \leftarrow \mathrm{D}^\sigma.\textsc{minHeight}([\pi_{p_{\min}} - \delta, \infty), [q,d])$    ▷ Lowest point on $\sigma$ still seen by $p_{\min}$
        return $\mathrm{D}^\pi.\textsc{maxIndex}((-\infty, \sigma_{q_{\min}} + \delta], [p_{\min}, p_{\mathrm{stop}} - 1])$
```

4. **Arbitrary greedy steps GreedyStep$_\pi(\pi', \sigma')$.** If, in some situation, it is only required to find an arbitrary index $p' \in \mathrm{reach}_{\pi'}(p,q)$ such that all $p \leqslant i \leqslant p'$ satisfy $\mathrm{vis}_{\sigma'}(i,q) \subseteq \mathrm{vis}_{\sigma'}(p',q)$ or report that no such index exists, we use the function $\textsc{GreedyStep}_\pi(\pi', \sigma')$ to denote that any such function suffices; in particular, $\textsc{MinGreedyStep}_\pi$ or $\textsc{MaxGreedyStep}_\pi$ can be used.

For σ, we define the obvious symmetric operations. Note that in these operations, it is not feasible to traverse all directly feasible points and check whether the visibility criterion is satisfied, since this would not necessarily yield a running time of $\mathcal{O}^*(1)$.

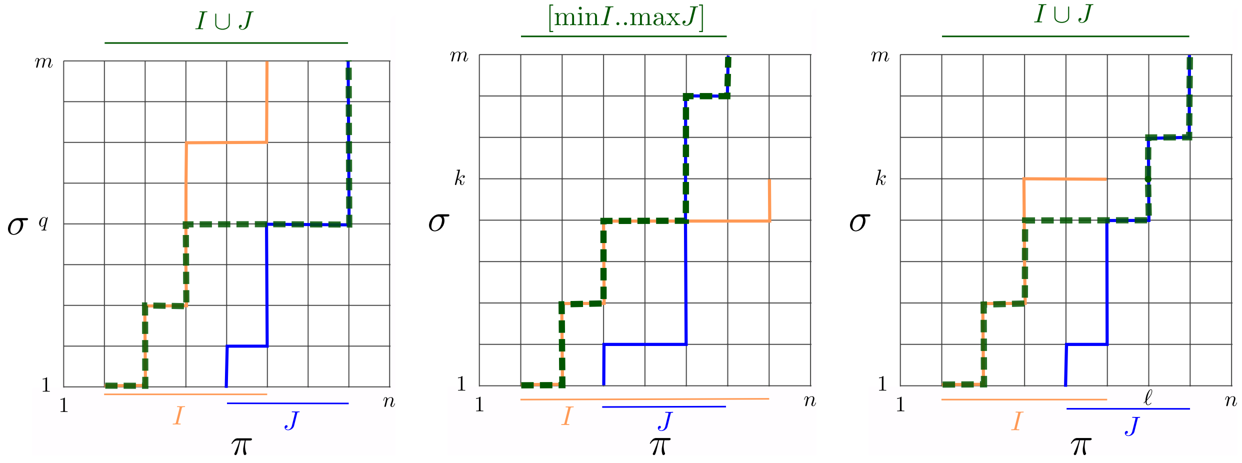
Lemma 4.10. *Using $O((n+m)\log nm)$ preprocessing time,* $\textsc{MaxGreedyStep}_\pi$*,* $\textsc{MinGreedyStep}_\pi$ *and* $\textsc{stop}_\pi$ *can be implemented to run in time $\mathcal{O}(\log nm)$.*

Proof. In time $\mathcal{O}((n+m)\log nm)$, we can build the data structure D^π for π and symmetrically D^σ for σ. Algorithms 2, 3 and 4 implement the greedy steps and $\textsc{stop}_\pi$ using only a constant number of queries to D^π and D^σ, each with running time $\mathcal{O}(\log n)$ or $\mathcal{O}(\log m)$. □

For the reduced free-space problem, these operations can be implemented even faster.

Lemma 4.11. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be input curves of the reduced free-space problem. Using $O((n+m)\log 1/\varepsilon)$ preprocessing time,* $\textsc{MaxGreedyStep}_\pi$*,* $\textsc{MinGreedyStep}_\pi$ *and* $\textsc{stop}_\pi$ *can be implemented to run in time $\mathcal{O}(\log 1/\varepsilon)$.*

Proof. We argue that range searching can be implemented with $\mathcal{O}(\log 1/\varepsilon)$ query time and $\mathcal{O}(n \log 1/\varepsilon)$ preprocessing time. This holds since for the point set $\mathcal{P} = \{(i, \pi_i) \mid i \in [1 \dots n]\}$ (1) the x-values are $1, \dots, n$, so that we can determine the relevant pointers in the first level of the fractional-cascading tree in constant time instead of $\mathcal{O}(\log n)$ and (2) all y-values are multiples of $\frac{1}{3}\varepsilon\delta$ and in $[-2\delta, 2\delta]$, i.e., there are only $\mathcal{O}(1/\varepsilon)$ different y-values. For the latter, note that any point $\pi_p > \delta$ sees no point in σ, and this is preserved by setting π_p to 2δ (and similarly for σ). Using these properties it is straightforward to adapt the fractional-cascading data structure, we omit the details. □



(a) Lemma 4.12 **(b)** Lemma 4.13 **(c)** Lemma 4.14

Figure 4: Composition properties of feasible traversals of one-dimensional separated curves.

4.3 Composition of one-dimensional curves

In this subsection, we collect essential composition properties of feasible traversals of one-dimensional curves that enable us to tackle the reduced free-space problem (see Figure 4 for an illustration of these results). The first tool is a union lemma that states that two intersecting intervals I, J of π that each have a feasible traversal together with σ prove that also $\pi_{I\cup J}$ can be traversed together with σ.

Lemma 4.12. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be one-dimensional separated curves and let $I, J \subseteq [1..n]$ be intervals with $I \cap J \neq \emptyset$. If $d_{\mathrm{dF}}(\pi_I, \sigma) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_J, \sigma) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{I\cup J}, \sigma) \leqslant \delta$.*

Proof. If $I \subseteq J$, the claim is trivial. W.l.o.g, let $I = [a_I..b_I]$ and $J = [a_J..b_J]$, where $a_I \leqslant a_J \leqslant b_I \leqslant b_J$. Let ϕ_I (and ϕ_J) be a feasible traversal of (π_I, σ) (and (π_J, σ), respectively). By reparameterization, we can assume that $\phi_I(t) = (\psi_I(t), f(t))$ and $\phi_J(t) = (\psi_J(t), f(t))$ for suitable (non-decreasing onto) functions $\psi_I, \psi_J : [0,1] \to [1..n]$ and $f : [0,1] \to [1..m]$. One of the following cases occurs.

Case 1: There is some $0 \leqslant t \leqslant 1$ with $\psi_I(t) = \phi_J(t)$. Then we can concatenate $\phi_I(0,t)$ and $\phi_J(t,1)$ to obtain a feasible traversal of $\phi_{I\cup J}$.

Case 2: For all $0 \leqslant t \leqslant 1$, we have $\psi_I(t) < \psi_J(t)$. Let σ_q be the highest point on σ. By $d_{\mathrm{dF}}(\pi_I, \sigma) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_J, \sigma) \leqslant \delta$, the point σ_q sees all points on $\pi_{I\cup J}$. There is some $0 \leqslant t^* \leqslant 1$ with $f(t^*) = q$. We can concatenate $\phi_I(0,t)$ and the traversal of $\pi_{\psi_I(t^*)..\psi_J(t^*)}$ and σ_q to obtain a feasible traversal of $\pi_{a_I..\psi_I(t^*)}$ and $\sigma_{1..f(t^*)}$. Appending $\phi_J(t^*, 1)$ to this traversal yields $d_{\mathrm{dF}}(\pi_{a_I..b_J}, \sigma) \leqslant \delta$. □

The second result formalizes situations in which a traversal ϕ of subcurves has to cross a traversal ψ of other subcurves, yielding the possibility to follow ϕ up to the crossing point and to follow ψ from there on.

Lemma 4.13. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be one-dimensional curves and consider intervals $I = [a_I..b_I]$ and $J = [a_J..b_J]$ with $J \subseteq I \subseteq [1..n]$, and $K = [1..k] \subseteq [1..m]$. If $d_{\mathrm{dF}}(\pi_I, \sigma_K) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_J, \sigma) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{a_I..b_J}, \sigma) \leqslant \delta$.*

Proof. Let ϕ be a feasible traversal of π_I and σ_K and ψ a feasible traversal of π_J and σ. We first show that ϕ and ψ cross, i.e., there are $0 \leqslant t, t' \leqslant 1$ such that $\phi(t) = \psi(t')$. For all $k \in [1..K]$, let $[s_k^\phi..e_k^\phi]$ denote the interval of points that ϕ traverses on π while staying in σ_k. Similarly, $[s_k^\psi..e_k^\psi]$ denotes the interval of points ψ traverses on π while staying in σ_k. Assume for contradiction that $[s_k^\phi..e_k^\phi]$ and $[s_k^\psi..e_k^\psi]$ are disjoint for all $1 \leqslant k \leqslant K$. Then initially, we have $s_1^\phi = a_I \leqslant a_J = s_1^\psi$ and hence $e_1^\phi < s_1^\psi$. This implies $s_2^\phi \leqslant e_1^\phi + 1 \leqslant s_1^\psi \leqslant s_2^\psi$ and inductively we obtain $e_k^\phi < s_k^\psi \leqslant e_k^\psi$ for all $k \in [1..K]$. This contradicts $e_K^\phi = b_I \geqslant b_J \geqslant e_K^\psi$. Hence, for some $1 \leqslant k \leqslant K$, $[s_k^\phi..e_k^\phi]$ and $[s_k^\psi..e_k^\psi]$ intersect, which gives $\phi(t) = (p, k) = \psi(t')$ for any $p \in [s_k^\phi, e_k^\phi] \cap [s_k^\psi, e_k^\psi]$ and the corresponding $0 \leqslant t, t' \leqslant 1$. By concatenating $\phi(0, t)$ with $\psi(t', 1)$, we obtain a feasible traversal of $\pi_{a_I..b_J}$ and σ. □

The last result in our composition toolbox strengthens Lemma 4.12 to the case that the traversal of π_I uses only an initial subcurve $\sigma_{1..k}$ of σ and not the complete curve.

Lemma 4.14. *Let $\pi = \pi_{1..n}$ and $\sigma = \sigma_{1..m}$ be one-dimensional separated curves and consider intervals $I = [a_I..b_I]$ and $J = [a_J..b_J]$ with $1 \leqslant a_I \leqslant a_J \leqslant b_I \leqslant b_J \leqslant n$, and $K = [1..k] \subseteq [1..m]$. If $d_{\mathrm{dF}}(\pi_I, \sigma_K) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_J, \sigma) \leqslant \delta$, then $d_F(\pi_{I \cup J}, \sigma) \leqslant \delta$.*

Proof. Let ϕ be any feasible traversal of π_J and σ. There exists $a_J \leqslant \ell \leqslant b_J$ with $\phi(t) = (\ell, k)$ for some $0 \leqslant t \leqslant 1$. Hence ϕ restricted to $[0, t]$ yields a feasible traversal of $\pi_{a_J..\ell}$ and σ_K, i.e., $d_{\mathrm{dF}}(\pi_{a_J..\ell}, \sigma_K) \leqslant \delta$. Since I and $[a_J..\ell]$ are intersecting, Lemma 4.12 yields that $d_{\mathrm{dF}}(\pi_{a_I..\ell}, \sigma_K) \leqslant \delta$. Let ψ be such a feasible traversal of $\pi_{a_I..\ell}$ and σ_K. Concatenating ψ at $\psi(1) = (\ell, k) = \phi(t)$ with $\phi(t, 1)$, we construct a feasible traversal of $\pi_{a_I..b_J}$ and σ, proving the claim. □

4.4 Solving the Reduced Free-space Problem

In this section, we solve the reduced free-space problems, using the structural properties derived in the previous section and the principles underlying the greedy algorithm of Section 4.2. Recall that the greedy steps implemented as discussed in Section 4.2.2 run in time $\mathcal{O}(\log 1/\varepsilon)$ on the input curves of the reduced free-space problem.

4.4.1 Single Entry

Given the separated curves $\pi = (\pi_1, \ldots, \pi_n)$ and $\sigma = (\sigma_1, \ldots, \sigma_m)$ and entry set $E = \{1\}$, we show how to compute F^σ. We present the following recursive algorithm.

Algorithm 5 Special Case: Single entry

```
function Find-σ-exits($\pi_{p..b}, \sigma_{q..d}$)
    if $q = d$ then
        if stop$_\pi(\pi_{p..b}, \sigma_q) = b + 1$ then
            return $\{q\}$                    ▷ The end of $\pi$ is reachable while staying in $\sigma_q$
        else return $\emptyset$

    if $p' \leftarrow$ MaxGreedyStep$_\pi(\pi_{p..b}, \sigma_{q..d})$ then
        return Find-σ-exits($\pi_{p'..b}, \sigma_{q..d}$))
    else if $q' \leftarrow$ GreedyStep$_\sigma(\pi_{p..b}, \sigma_{q..d})$ then
        return Find-σ-exits($\pi_{p..b}, \sigma_{q..q'-1}$) $\cup$ Find-σ-exits($\pi_{p..b}, \sigma_{q'..d}$)
    else
        return Find-σ-exits($\pi_{p..b}, \sigma_{q..d-1}$)                    ▷ No greedy step possible
```

The following property establishes that a greedy step on a long curve is also a greedy step on a shorter curve. Clearly, the converse does not necessarily hold.

Proposition 4.15. *Let $1 \leqslant p \leqslant P \leqslant n$ and $1 \leqslant q \leqslant Q \leqslant m$. Any greedy step on π from (p, q) to (p', q) with $p' \leqslant P$ is also a greedy step with respect to $\tilde{\pi} := \pi_{p..P}$ and $\tilde{\sigma} := \sigma_{q..Q}$, i.e., if there is some $p' \leqslant P$ with $\text{vis}_\sigma(i, q) \subseteq \text{vis}_\sigma(p', q)$ for all $p \leqslant i \leqslant p'$, then also $\text{vis}_{\tilde{\sigma}}(i, q) \subseteq \text{vis}_{\tilde{\sigma}}(p', q)$.*

Proof. From the definition of vis_σ, we immediately derive $\text{vis}_{\tilde{\sigma}}(i, q) = \text{vis}_\sigma(i, q) \cap [q..Q] \subseteq \text{vis}_\sigma(p', q) \cap [q..Q] = \text{vis}_{\tilde{\sigma}}(p', q)$ for all $p \leqslant i \leqslant p'$. Restricting the length of π also has no influence on the greedy property, except for the trivial requirement that p' still has to be contained in the restricted curve. □

Lemma 4.16. *Algorithm 5 correctly identifies F^σ given the single entry $E = \{1\}$.*

Proof. Clearly, if Find-σ-exits(π, σ) finds and returns an exit e on σ, then it is contained in F^σ, since the algorithm uses only feasible (greedy) steps. Conversely, we show that for all $I = [p..b]$ and $J = [q..d]$, where (p, q) is a greedy point pair of π and σ, and all $e \in J$ with $d_{\text{dF}}(\pi_I, \sigma_{J \cap [1..e]}) \leqslant \delta$, we have $e \in$ Find-σ-Exits(π_I, σ_J), i.e. we find all exits.

Consider some call of Find-σ-Exits(π_I, σ_J) for which the precondition is fulfilled. If J consists only of a single point, then $J = \{e\}$, and a feasible traversal of π_I and σ_J exists if and only if σ_e sees all points on π_I. Let $I = [p..b]$, then this happens if and only if $\text{stop}_\pi(\pi_I, \sigma_e) = b + 1$, hence the base case is treated correctly.

Assume that $I = [p..b]$ and a maximal greedy step p' on π exists. By Property 4.15, this step is a greedy step also with respect to $\sigma_{J \cap [1..e]}$. Hence by Lemma 4.8, if there is a traversal of $\pi_{p..b}$ and $\sigma_{J \cap [1..e]}$, then a traversal of $\pi_{[p'..b]}$ and $\sigma_{J \cap [1..e]}$ also exists.

Consider the case in which $J = [q..d]$ and a greedy step q' in σ exists. If $e < q'$, then $e \in [q..q'-1]$ and $J \cap [1..e] = [q..q' - 1] \cap [1..e]$. Hence, e is found in the recursive call with $J' = [q..q' - 1]$. If $e \geqslant q'$, then by Property 4.15, this step is a greedy step with respect to the curves π_I and $\sigma_{J \cap [1..e]}$. Again, by Lemma 4.8, the existence of a feasible traversal of π_I and σ_J implies that also a feasible traversal of π_I and $\sigma_{J \cap [q'..e]}$ exists.

It remains to regard the case in which no greedy step exists. By Lemma 4.8, there is no feasible traversal of $\pi_{1..n}$ and $\sigma_{1..d}$. This implies $e \neq d$ and all exits are found in the recursive call with $J' = [q, d - 1]$. □

Lemma 4.17. Find-σ-Exits$(\pi_{p..b}, \sigma_{q..d})$ *runs in time $\mathcal{O}((d - q + 1) \cdot \log 1/\varepsilon)$.*

Proof. Since the algorithm's greedy steps on π are maximal, after each greedy step on π, we split σ (by a greedy step on σ) or shorten σ (if no greedy step on σ is found). Thus, it takes at most $\mathcal{O}(\log 1/\varepsilon)$ time until σ is split or shortened. The base case is also handled in time $\mathcal{O}(\log 1/\varepsilon)$. In total, this yields a running time of $\mathcal{O}((d - q + 1) \log 1/\varepsilon)$. □

Note that by swapping the roles of π and σ, Find-σ-Exits can be used to determine F^π given the single entry σ_1 on σ. This is equivalent to having the single entry $E = \{1\}$ on π. Thus, we can also implement the function Find-π-Exits$(\pi_{1..n}, \sigma_{1..m})$ that returns F^π given the single entry $E = \{1\}$ in time $\mathcal{O}(n \log 1/\varepsilon)$.

4.4.2 Entries on π, Exits on π

In this section, we tackle the task of determining F^π given a set of entries E on π. It is essential to avoid computing the exits by iterating over every single entry. We show how to divide π into disjoint subcurves that can be solved by a single call to Find-π-Exits each.

Assume we want to traverse $\pi_{p..b}$ and $\sigma_{q..d}$ starting in π_p and σ_q. Let $u(p) := \max\{p' \in [p,b] \mid \exists q \leqslant q' \leqslant d : d_{\mathrm{dF}}(\pi_{p..p'}, \sigma_{q..q'}) \leqslant \delta\}$ be the last point on π that is reachable while traversing an arbitrary subcurve of $\sigma_{q..d}$ that starts in σ_q. This point fulfills the following properties.

Lemma 4.18. *It holds that*

1. *If there are $p \leqslant e \leqslant e' \leqslant u(p)$ with $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma_{q..d}) \leqslant \delta$, then $d_{\mathrm{dF}}(\pi_{p..e'}, \sigma_{q..d}) \leqslant \delta$.*
2. *For all $p \leqslant e \leqslant u(p) < e'$, we have that $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma_{q..d}) > \delta$.*

Proof. By definition of $u(p)$, there is a $q \leqslant q' \leqslant d$ with $d_{\mathrm{dF}}(\pi_{p..u(p)}, \sigma_{q..q'}) \leqslant \delta$. Since $[e,e'] \subseteq [p,u(p)]$, Lemma 4.13 proves the first statement. For the second statement, assume for contradiction that $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma_{q..d}) \leqslant \delta$. Then, Lemma 4.14 yields that $d_{\mathrm{dF}}(\pi_{p..e'}, \sigma_{q..d}) \leqslant \delta$. This is a contradiction to the choice of $u(p)$, since $e' > u(p)$. □

The above lemma implies that we can ignore all entries in $[p..u(p)]$ except for p and that all exits reachable from p are contained in the interval $[p..u(p)]$. This gives rise to the following algorithm.

Algorithm 6 Given entry points E on π, compute all exits on π.

```
function π-EXITS-FROM-π($\pi, \sigma, E$)
    $S \leftarrow \emptyset$
    while $E \neq \emptyset$ do
        $\hat{p} \leftarrow$ pop minimal index from $E$
        $p \leftarrow \hat{p}$, $q \leftarrow 1$
        repeat
            if $q' \leftarrow$ MAXGREEDYSTEP$_\sigma(\pi_{p..n}, \sigma_{q..m})$ then
                $q \leftarrow q'$
            if $p' \leftarrow$ GREEDYSTEP$_\pi(\pi_{p..n}, \sigma_{q..m})$ then
                $p \leftarrow p'$
        until no greedy step was found in the last iteration
        $\overline{p} \leftarrow$ STOP$_\pi(\pi_{p..n}, \sigma_{q..m}) - 1$        ▷ determines the maximal reachable point $u(\hat{p})$
        $S \leftarrow S \cup$ FIND-π-EXITS$(\pi_{\hat{p}..\overline{p}}, \sigma)$
        $E \leftarrow E \cap [\overline{p}+1, n]$        ▷ drops all entries in $[\hat{p}, u(\hat{p})]$
    return $S$
```

Lemma 4.19. *Algorithm 6 correctly computes F^π.*

Proof. We first argue that for each considered entry $\hat{p}$, the algorithm computes $\overline{p} = u(\hat{p})$. Clearly, $\overline{p} \leqslant u(\hat{p})$, since only feasible steps are used to reach $\overline{p}$. If $\overline{p} = m$, this already implies that also $u(\hat{p}) = m$. Otherwise, let (p,q) be the greedy point pair on the curves $\pi_{\hat{p}...n}$ and σ for which no greedy step has been found. Then by Lemma 4.8, for $p_{\mathrm{stop}} := \mathrm{STOP}_\pi(\pi_{p..n}, \sigma_{q..m})$ and all $1 \leqslant q' \leqslant m$, we have that $d_{\mathrm{dF}}(\pi_{\hat{p}..p_{\mathrm{stop}}}, \sigma_{1..q'}) > \delta$. Hence, $u(\hat{p}) < p_{\mathrm{stop}}$. Finally, note that Algorithm 6 computes $\overline{p} = p_{\mathrm{stop}} - 1$, which proves $\overline{p} = u(\hat{p})$.

It is clear that every found exit is included in F^π. Conversely, let $e' \in F^\pi$ and $1 \leqslant e \leqslant n$ be such that $d_{\mathrm{dF}}(\pi_{e..e'}, \sigma) \leqslant \delta$. For some $\hat{p}$ with $\hat{p} \leqslant e \leqslant u(\hat{p}) = \overline{p}$, we run FIND-π-EXITS$(\pi_{\hat{p}..\overline{p}}, \sigma)$. Hence by Lemma 4.18 (2), $e' \leqslant u(\hat{p})$ and by Lemma 4.18 (1), $d_{\mathrm{dF}}(\pi_{\hat{p}..e'}, \sigma) \leqslant \delta$. Hence, the corresponding call FIND-π-EXITS$(\pi_{\hat{p}..\overline{p}}, \sigma)$ will find e'. □

Lemma 4.20. *Using preprocessing time $\mathcal{O}((n+m)\log 1/\varepsilon)$, Algorithm 6 runs in time $\mathcal{O}(n \log 1/\varepsilon)$.*

Proof. We first bound the cost of all calls Find-π-Exits(π_{I_i}, σ). Clearly, all intervals I_i are disjoint with $\bigcup I_i \subseteq [1..n]$. Hence, by Lemma 4.17, the total time spent in these calls is bounded by $\mathcal{O}(\sum_i |I_i| \log(1/\varepsilon)) = \mathcal{O}(n \log 1/\varepsilon)$. To bound the number greedy steps, let $p_1, \ldots, p_k$ be the distinct indices considered as values of p during the execution of π-exits-from-$\pi(\pi, \sigma)$. Between changing p from each p_i to p_{i+1}, we will make, by maximality, at most one call to MaxGreedyStep$_\sigma$ and at most call to GreedyStep$_\pi$. Since $k \leqslant n$, the total cost of greedy calls is bounded by $\mathcal{O}(n \log 1/\varepsilon)$ as well. The total time spent in all other operations is bounded by $\mathcal{O}(n \log 1/\varepsilon)$. □

4.4.3 Entries on π, Exits on σ

Similar to the previous section, we show how to compute the exits F^σ given entries E on π, by reducing the problem to calls of Find-σ-Exits on subcurves of π and σ. This time, however, the task is more intricate. For any index p on π, let $Q(p) := \min\{q \mid d_{\mathrm{dF}}(\pi_{p..n}, \sigma_{1..q}) \leqslant \delta\}$ be the endpoint of the shortest initial fragment of σ such that the remaining part of π can be traversed together with this fragment[5]. Let $P(p) := \min\{p' \mid d_{\mathrm{dF}}(\pi_{p..p'}, \sigma_{1..Q(p)}) \leqslant \delta\}$ be the endpoint of the shortest initial fragment of π, such that $\sigma_{Q(p)}$ can be reached by a feasible traversal.

Note that by definition, entries p with $Q(p) = \infty$ are irrelevant for determining the exits on σ. In fact, if an entry p is *relevant*, i.e., $Q(p) < \infty$, it is easy to compute $Q(p)$ due to the following lemma.

Lemma 4.21. *Let $Q'(p) := \min\{q \mid \sigma_q \geqslant \max_{i \in [p..n]} \pi_i - \delta\}$. If $Q(p) < \infty$, then $Q(p) = Q'(p)$. Similarly, $Q(p) < \infty$ implies that $P(p) = \min\{p' \mid \pi_{p'} \leqslant \min_{i \in [q..Q(p)]} \sigma_i + \delta\} < \infty$.*

Proof. Assume that $Q(p) < Q'(p)$ holds, then no point in $\sigma_{1..Q(p)}$ sees the highest point in $\pi_{p..n}$. Hence no feasible traversal of these curves can exist, yielding a contradiction. Assume that $Q(p) > Q'(p)$ holds instead and consider the feasible traversal ϕ of the shortest initial fragment of σ that passes through all points in $\pi_{p..n}$. At some point ϕ visits $(\pi_{p'}, \sigma_{Q'(p)})$ for some $p \leqslant p' \leqslant n$. We can alter this traversal to pass through the remaining curve $\pi_{p'..n}$ while staying in $\sigma_{Q'(p)}$, since $\sigma_{Q'(p)}$ sees all points on $\pi_{p'..n}$. This gives a feasible traversal of $\pi_{p..n}$ and $\sigma_{1..Q'(p)}$, which is a contradiction to the choice of ϕ and $Q(p) > Q'(p)$.

The statement for $P(p)$ follows analogously by regarding the curves $\pi_{p..n}$ and $\sigma_{1..Q(p)}$ and switching their roles. □

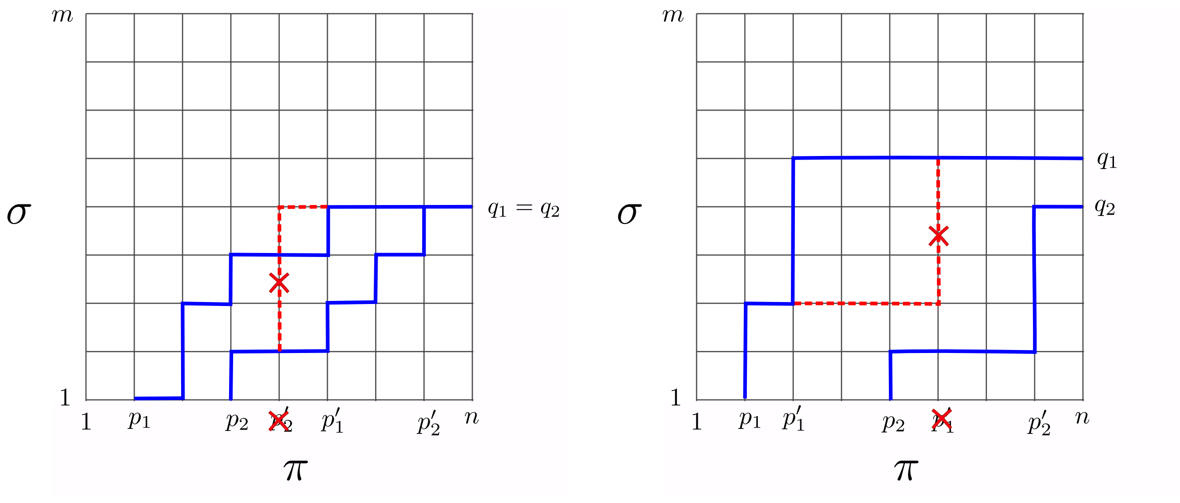
Note that the previous lemma shows that for relevant entries $p_1 < p_2$, we have $Q(p_1) \geqslant Q(p_2)$, since for relevant entries, $Q(p_1) = Q'(p_1) \geqslant Q'(p_2) = Q(p_2)$. We will use the following lemma to argue that (i) if $Q(p_1) = Q(p_2)$, entry p_1 dominates p_2, and (2) if $Q(p_1) > Q(p_2)$, we have $p_2 \notin [p_1..P(p_1)]$. Hence, we can ignore all entries in $[p_1..P(p_1)]$ except for p_1 itself.

Lemma 4.22. *Let $p_1 < p_2$ be indices on π with $q_1 := Q(p_1) < \infty$ and $q_2 := Q(p_2) < \infty$. Let $p'_1 := P(p_1)$ and $p'_2 := P(p_2)$. If $q_1 = q_2$, then $p'_1 \leqslant p'_2$. Otherwise, i.e., if $q_1 > q_2$, we even have $p'_1 < p_2$.*

Proof. See Figure 5 for illustrations. Let $q_1 = q_2$. Assume for contradiction that $p'_1 > p'_2$, then we have $d_{\mathrm{dF}}(\pi_{p_1..p'_1}, \sigma_{1..q_1}) \leqslant \delta$ and $d_{\mathrm{dF}}(\pi_{p_2..p'_2}, \sigma_{1..q_1}) \leqslant \delta$, where $[p_2..p'_2] \subseteq [p_1..p'_1]$. Hence by Lemma 4.13, $d_{\mathrm{dF}}(\pi_{p_1..p'_2}, \sigma_{1..q_1}) \leqslant \delta$ and thus $p'_1 \leqslant p'_2$, which is a contradiction to the assumption.

For the second statement, let p be maximal such that $\pi_p > \sigma_{q_2} + \delta$. If p does not exist or $p < p_1$, we have that $Q'(p_1) = Q'(p_2)$ and hence by Lemma 4.21, $q_1 = q_2$. Note that additionally $p < p_2$, since otherwise $\sigma_{q_2} < \pi_p - \delta$ with $p \geqslant p_2$ shows that $q_2 \neq Q'(p_2)$ contradicting Lemma 4.21. Thus, in what follows, we can assume that $p_1 < p < p_2$.

[5] As a convention, we use $\min \emptyset = \max \emptyset = \infty$.



(a) Case $q_1 = q_2$.

(b) Case $q_1 > q_2$.

Figure 5: Illustration of Lemma 4.22. For both p_i, $i \in \{1,2\}$, a feasible traversal of the curves $\pi_{p_i..p'_i}$ and $\sigma_{1..q_i}$ is depicted as monotone paths in the free-space.

Assume for contradiction that $q_1 > q_2$ and $p'_1 \geqslant p_2$. Then a feasible traversal ϕ of $\pi_{p_1..p'_1}$ and $\sigma_{1..q_1}$ visits (π_p, σ_q) for some $1 \leqslant q \leqslant q_1$. It even holds that $q < q_1$, since otherwise there is a feasible traversal of $\sigma_{1..q_1}$ and $\pi_{p_1..p}$ with $p < p'_1$, contradicting the choice of p'_1. Clearly, $\sigma_q > \sigma_{q_2}$, since π_p sees σ_q, while it does not see σ_{q_2}. Since by choice of p, σ_{q_2} sees all of $\pi_{p+1..n}$ and σ_q sees only more (including π_p), we conclude that we can traverse all points of $\pi_{p..n}$ while staying in σ_q. Concatenating this traversal to the feasible traversal ϕ yields $d_{\mathrm{dF}}(\pi_{p_1..n}, \sigma_{1..q}) \leqslant \delta$ and thus $Q'(p_1) \leqslant q < q_1$, which is a contradiction to Lemma 4.21. This proves that $q_1 > q_2$ implies $p'_1 \leqslant p_2$. □

Lemma 4.23. *Algorithm 7 fulfills the following properties.*

1. *Let (p, q) with $q < Q'(\hat{p})$ be a greedy point pair of $\pi_{\hat{p}..n}$ and $\sigma_{1..Q'(\hat{p})}$ for which no greedy step exists. For all $e \in [\hat{p}, p]$, we have $Q(e) = \infty$.*

2. *For each $\hat{p}$ considered, if $Q(\hat{p}) < \infty$, the algorithm calls* Find-σ-Exits$(\pi_{P(\hat{p})..n}, \sigma_{Q(\hat{p})..\bar{q}})$. *In this case, the point $(P(\hat{p}), Q(\hat{p}))$ is a greedy pair of $\pi_{\hat{p}..n}$ and σ.*

Proof. For the first statement, assume for contradiction that $Q(e) < \infty$. By Lemma 4.21, $Q(e) = Q'(e)$, which implies that for all $q' < Q'(e) \leqslant Q'(\hat{p})$, we have $\sigma_{q'} < \sigma_{Q'(e)}$ and hence $\mathrm{vis}_\pi(p, q') \subseteq \mathrm{vis}_\pi(p, Q'(e))$. Hence, $\textsc{stop}_\sigma(\pi_{p..n}, \sigma_{q..Q'(\hat{p})}) \leqslant Q'(e)$, since otherwise $Q'(e) \leftarrow \textsc{GreedyStep}_\sigma(\pi_{p..n}, \sigma_{q..Q'(\hat{p})})$. By Lemma 4.8, this proves that $d_{\mathrm{dF}}(\pi_{\hat{p}..n}, \sigma_{1..Q'(e)}) > \delta$. Since $d_{\mathrm{dF}}(\pi_{\hat{p}..e}, \sigma_{1..q'}) \leqslant \delta$ for some $q' < Q'(e)$, Lemma 4.14 yields $d_{\mathrm{dF}}(\pi_{e..n}, \sigma_{1..Q'(e)}) > \delta$. This is a contradiction to $Q(e) = Q'(e)$.

For the second statement, note that if $Q(\hat{p}) < \infty$, then by Lemma 4.21, $Q(\hat{p}) = Q'(\hat{p})$. Hence Lemma 4.8 yields that the algorithm finds a feasible traversal of $\pi_{\hat{p}..p}$ and $\sigma_{1..Q'(\hat{p})}$ for some $\hat{p} \leqslant p \leqslant n$. This shows that $P(\hat{p}) \leqslant p < \infty$. Let $\sigma' := \sigma_{1..Q(\hat{p})}$ and assume that there is a $p' < p$ with $d_{\mathrm{dF}}(\pi_{\hat{p}..p'}, \sigma') \leqslant \delta$ and let $(\tilde{p}, \tilde{q})$ be the greedy point of $\pi_{\hat{p}..n}$ and σ' right before the algorithm made a greedy step on π to some index in $(p', p]$. By maximality of the greedy steps on σ, there exists $\tilde{q} < q_{\min} < Q(\hat{p})$ such that $\pi_{\tilde{p}}$ does not see $\sigma_{q_{\min}}$, since otherwise $Q(\hat{p}) \in \mathrm{reach}_{\sigma'}(\tilde{p}, \tilde{\sigma})$ with $\mathrm{vis}_\pi(\tilde{p}, \tilde{q}) \subsetneq \mathrm{vis}_\pi(\tilde{p}, Q(\hat{p}))$, i.e., $Q(\hat{p})$ would be a greedy step on σ'. By minimality of greedy

Algorithm 7 Given entry points E on π, compute all exits on σ.

```
function $\sigma$-EXITS-FROM-$\pi(\pi, \sigma, E)$
    $F \leftarrow \emptyset$, $\overline{q} \leftarrow m$
    repeat
        $\hat{p} \leftarrow$ pop minimal index from $E$
        $p \leftarrow \hat{p}$, $q \leftarrow 1$
        $Q' \leftarrow Q'(p)$
        repeat
            if $q' \leftarrow \textsc{MaxGreedyStep}_{\sigma}(\pi_{p..n}, \sigma_{q..Q'})$ then
                $q \leftarrow q'$
            if $q \neq Q'$ and $p' \leftarrow \textsc{MinGreedyStep}_{\pi}(\pi_{p..n}, \sigma_{q..Q'})$ then
                $p \leftarrow p'$
        until $q = Q'$ or no greedy step was found in the last iteration
        if $q = Q'$ then
            $F \leftarrow F \cup \textsc{Find-}\sigma\textsc{-Exits}(\pi_{p..n}, \sigma_{Q'..\overline{q}})$
            $\overline{q} \leftarrow Q' - 1$
        $E \leftarrow E \cap [p+1, n]$
    until $E = \emptyset$
    return $F$
```

steps on π, $\mathrm{vis}_{\sigma'}(\tilde{p}, \tilde{q}) \supsetneq \mathrm{vis}_{\sigma'}(i, \tilde{q})$ for all $\tilde{p} \leqslant i \leqslant p'$. Hence, no vertex on $\pi_{\tilde{p}..p'}$ sees $\sigma_{q_{\min}}$, which proves $d_{\mathrm{dF}}(\pi_{\tilde{p}..p'}, \sigma') > \delta$. Since $(\tilde{p}, \tilde{\sigma})$ is a greedy pair of $\pi_{\hat{p}..p'}$ and σ', this yields that $d_{\mathrm{dF}}(\pi_{\hat{p}..p'}, \sigma') > \delta$ by Lemma 4.8, which is a contradiction to the assumption. Hence, the algorithm calls $\textsc{Find-}\sigma\textsc{-Exits}(\pi_{p..n}, \sigma_{Q'(\hat{p})..\overline{q}})$, where $p = P(\hat{p})$ and $Q'(\hat{p}) = Q(\hat{p})$.

It remains to show that $(P(\hat{p}), Q(\hat{p}))$ is also a greedy pair of $\pi_{\hat{p}..n}$ and the *complete* curve σ. By Lemma 4.21, every $\hat{p} \leqslant p < P(\hat{p})$ satisfies $\pi_p > \pi_{P(\hat{p})}$ and hence $\mathrm{vis}_{\sigma}(p, q) \subseteq \mathrm{vis}_{\sigma}(P(\hat{p}), q)$ for all $1 \leqslant q \leqslant m$. Hence, if at some greedy pair (p, q), $q \leqslant Q(\hat{p})$, a greedy step $p' \leftarrow \textsc{GreedyStep}_{\pi}(\pi_{p..n}, \sigma)$ with $p' \geqslant P(\hat{p})$ exists, then also $P(\hat{p}) \leftarrow \textsc{GreedyStep}_{\pi}(\pi_{p..n}, \sigma)$, which shows that $(P(\hat{p}), q)$ is a greedy point of $\pi_{\hat{p}..n}$ and σ. If $q = Q(\hat{p})$, then $(P(\hat{p}), Q(\hat{p}))$ is a greedy point pair. Otherwise, by Lemma 4.21, $P(\hat{p})$ sees all of $\sigma_{q..Q(\hat{p})}$ and $\sigma_q < \sigma_{Q(\hat{p})}$, hence $Q(\hat{p}) \in \textsc{GreedyStep}_{\sigma}(\pi_{P(p)..n}, \sigma)$ and $(P(\hat{p}), Q(\hat{p}))$ is a greedy step of $\pi_{\hat{p}..n}$ and σ.

It is left to consider the case that for all greedy pairs (p, q), $q \leqslant Q(\hat{p})$, of $\pi_{\hat{p}..n}$ and σ, no greedy step to some $p' \geqslant P(\hat{p})$ exists. Then there is some (p, q) with $p < P(\hat{p})$ and $q \leqslant Q(\hat{p})$ for which no greedy step exists at all. We have $p_{\mathrm{stop}} := \textsc{stop}_{\pi}(\pi_{p..n}, \sigma_{q..m}) \leqslant P(\hat{p})$, since otherwise $P(\hat{p})$ would be a greedy step. Since Lemma 4.8 shows that $d_{\mathrm{dF}}(\pi_{\hat{p}..p_{\mathrm{stop}}}, \sigma_{1..q}) > \delta$, this contradicts $d_{\mathrm{dF}}(\pi_{\hat{p}..P(\hat{p})}, \sigma_{1..Q(\hat{p})}) \leqslant \delta$.

□

Lemma 4.24. *Algorithm 7 correctly computes F^{σ}.*

Proof. Clearly, any exit found is contained in F^{σ}, since σ-EXITS-FROM-π and $\textsc{Find-}\sigma\textsc{-Exits}$ only use feasible steps. For the converse, let $e \in E$ be an arbitrary entry and consider the set $F^{\sigma}_e = \{q \mid d_{\mathrm{dF}}(\pi_{e..n}, \sigma_{1..q}) \leqslant \delta\}$ of σ-exits corresponding to the entry e.

We first show that if $F^{\sigma}_e \neq \emptyset$ and hence $Q(e) < \infty$, we have $F^{\sigma}_e = \textsc{Find-}\sigma\textsc{-Exits}(\pi_{P(e)..n}, \sigma_{Q(e)..m})$. Let $\overline{e} \in F^{\sigma}_e$. By Lemma 4.23, $(P(e), Q(e))$ is a greedy pair of $\pi_{e..n}$ and σ and hence also of $\pi_{e..n}$ and $\sigma_{1..\overline{e}}$. Lemma 4.8 thus implies $d_{\mathrm{dF}}(\pi_{P(e)..n}, \sigma_{Q(e)..\overline{e}}) \leqslant \delta$ and consequently $\overline{e} \in \textsc{Find-}\sigma\textsc{-Exits}(\pi_{P(e)..n}, \sigma_{Q(e)..m})$. The converse clearly holds as well.

Note that e is not considered as $\hat{p}$ in any iteration of the algorithm if and only if the algorithm considers some $\hat{p}$ with $e \in [\hat{p}+1..p]$, where either (i) the algorithm finds a greedy pair (p,q) of $\pi_{\hat{p}..n}$ and $\sigma_{1..Q'(\hat{p})}$ that allows no further greedy steps, or (ii) the algorithm calls Find-σ-Exits($\pi_{p..n}, \sigma_{Q'(\hat{p})..\overline{q}}$), where $p = P(\hat{p})$ by Lemma 4.23. In the first case, $F_e^\sigma = \emptyset$ since Lemma 4.23 proves $Q(e) = \infty$. In the second case, if $F_e^\sigma \neq \emptyset$, we have $Q(e) < \infty$, and hence by Lemma 4.22, $Q(e) = Q(\hat{p})$ and $P(\hat{p}) \leqslant P(e)$. Since $\sigma_{Q(\hat{p})}$ sees all of $\pi_{P(\hat{p})..n}$, any exit reachable from $(P(e), Q(e))$ is reachable from $(P(\hat{p}), Q(\hat{p}))$ as well. Hence $F_e^\sigma \subseteq F_{\hat{p}}^\sigma$.

Let $\hat{p}_1 \leqslant .. \leqslant \hat{p}_k$ be the entries considered as $\hat{p}$ by the algorithm. It remains to show that the algorithm finds all exits $\bigcup_{i=1}^k F_{\hat{p}_i}^\sigma$. We inductively show that the algorithm computes $F_{\hat{p}_i}^\sigma \setminus \bigcup_{j<i} F_{\hat{p}_j}^\sigma$ in the loop corresponding to $\hat{p} = \hat{p}_i$. The base case $i = 1$ follows immediately. Note that for every $i \geqslant 2$, the corresponding loop computes Find-σ-Exits($\pi_{P(\hat{p}_i)..n}, \sigma_{Q(\hat{p}_i)..Q(\hat{p}_{i-1})-1}$) $= F_{\hat{p}_i}^\sigma \cap [Q(\hat{p}_i)..Q(\hat{p}_{i-1})-1]$. The claim follows if we can show $F_{\hat{p}_i}^\sigma \cap [Q(\hat{p}_{i-1})..m] \subseteq F_{\hat{p}_{i-1}}$. Let $\overline{e} \in F_{\hat{p}_i}^\sigma$ with $\overline{e} \geqslant Q(\hat{p}_{i-1})$. Then $d_{\mathrm{dF}}(\pi_{\hat{p}_i..n}, \sigma_{1..\overline{e}}) \leqslant \delta$. Together with $d_{\mathrm{dF}}(\pi_{\hat{p}_{i-1}..n}, \sigma_{1..Q(\hat{p}_{i-1})}) \leqslant \delta$, Lemma 4.13 shows that $d_{\mathrm{dF}}(\pi_{\hat{p}_{i-1}..n}, \sigma_{1..\overline{e}}) \leqslant \delta$ and hence $e \in F_{\hat{p}_{i-1}}^\sigma$.

□

Lemma 4.25. *Algorithm 7 runs in time $\mathcal{O}((n+m)\log 1/\varepsilon)$.*

Proof. Consider the total cost of the calls Find-σ-Exits(π_{I_i}, σ_{J_i}). Since all J_i are disjoint and $\bigcup_i J_i \subseteq [1..m]$, Lemma 4.17 bounds the total cost of such calls by $\mathcal{O}(\sum_i |J_i| \log(1/\varepsilon)) = \mathcal{O}(m \log(1/\varepsilon))$. Let $p_1, \ldots, p_k$ denote the distinct indices considered as p during the execution of the algorithm. Between changing p_i to p_{i+1}, we will make at most one call to MaxGreedyStep$_\sigma$ (by maximality) and at most once call to MinGreedyStep$_\pi$. Hence $k \leqslant n$ bounds the number of calls to greedy steps by $\mathcal{O}(n \log(1/\varepsilon))$. □

References

[1] P. Agarwal, R. B. Avraham, H. Kaplan, and M. Sharir. Computing the discrete Fréchet distance in subquadratic time. In *Proc. 24th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA'13)*, pages 156–167, 2013.

[2] H. Alt. The computational geometry of comparing shapes. In *Efficient Algorithms*, volume 5760 of *LNCS*, pages 235–248. Springer, 2009.

[3] H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.

[4] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.

[5] H. Alt, C. Knauer, and C. Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38 (1):45–58, 2004.

[6] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. In *Proc. 14th Annu. European Symp. Algorithms (ESA'06)*, pages 52–63. Springer, 2006.

[7] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proc. 31st International Conf. Very Large Data Bases (VLDB'05)*, pages 853–864, 2005.

[8] K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annu. IEEE Sympos. Foundations of Computer Science (FOCS'14)*. IEEE, 2014. To appear, preprint at arXiv:1404.1448.

[9] K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA'09)*, pages 645–654. SIAM, 2009.

[10] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Internat. J. Comput. Geom. Appl.*, 21(3):253–282, 2011.

[11] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog - with an application to Alt's conjecture. In *Proc. 25th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA'14)*, pages 1399–1413, 2014.

[12] E. W. Chambers, É. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. *Computational Geometry*, 43(3):295–311, 2010.

[13] B. Chazelle and L. J. Guibas. Fractional cascading: I. a data structuring technique. *Algorithmica*, 1 (2):133–162, 1986.

[14] D. Chen, A. Driemel, L. J. Guibas, A. Nguyen, and C. Wenk. Approximate map matching with respect to the Fréchet distance. In *Proc. 13th Workshop on Algorithm Engineering and Experiments (ALENEX'11)*, pages 75–83. SIAM, 2011.

[15] A. F. Cook and C. Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms*, 7(1):193–204, 2010.

[16] A. Driemel and S. Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

[17] A. Driemel, S. Har-Peled, and C. Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012.

[18] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.

[19] M. Godau. A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In *Proc. 8th Sympos. Theoret. Aspects Comput. Sci. (STACS'91)*, volume 480 of *LNCS*, pages 127–136. Springer, 1991.

[20] J. Gudmundsson and M. Smid. Fréchet queries in geometric trees. In H. L. Bodlaender and G. F. Italiano, editors, *Algorithms - ESA 2013*, volume 8125 of *LNCS*, pages 565–576. Springer, 2013.

[21] S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. In *Proc. 27th Annu. Symp. Comp. Geometry (SoCG'11)*, pages 448–457. ACM, 2011.

[22] P. Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proc. 18th Annu. Symp. Comp. Geometry (SoCG'02)*, pages 102–106. ACM, 2002.

[23] G. S. Lueker. A data structure for orthogonal range queries. In *Proc. 19th Annu. Sympos. Foundations of Computer Science*, SFCS '78, pages 28–34. IEEE, 1978.

[24] A. Maheshwari, J.-R. Sack, K. Shahbaz, and H. Zarrabi-Zadeh. Fréchet distance with speed limits. *Computational Geometry*, 44(2):110–120, 2011.

[25] M. E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proc. 7th Intl. Conf. Comp. Vision*, volume 1, pages 108–115. IEEE, 1999.

[26] D. E. Willard. *Predicate-oriented database search algorithms.* PhD thesis, Aiken Comput. Lab, Harvard Univ., Cambridge, MA, 1978. Report TR-20-78.