Discrete Comput Geom (2012) 48:94–127
DOI 10.1007/s00454-012-9402-z

Approximating the Fréchet Distance for Realistic Curves in Near Linear Time

Anne Driemel · Sariel Har-Peled · Carola Wenk

Received: 1 April 2010 / Revised: 21 December 2011 / Accepted: 26 January 2012 /
Published online: 18 February 2012
© Springer Science+Business Media, LLC 2012

Abstract We present a simple and practical $(1+\varepsilon)$-approximation algorithm for the Fréchet distance between two polygonal curves in $\mathbb{R}^d$. To analyze this algorithm we introduce a new realistic family of curves, c-packed curves, that is closed under simplification. We believe the notion of c-packed curves to be of independent interest. We show that our algorithm has near linear running time for c-packed polygonal curves, and similar results for other input models, such as low-density polygonal curves.

Keywords Frechet distance · Approximation algorithms · Realistic input models

1 Introduction

Comparing geometric shapes is a task that arises in a wide arena of applications. The Fréchet distance and its variants have been used, to this end, to compare curves in applications such as dynamic time-warping [19], speech recognition [21], signature and

The latest full version of this paper is available online [16].

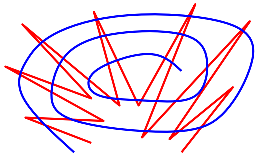
A. Driemel
Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
e-mail: anne@cs.uu.nl

S. Har-Peled (✉)
Department of Computer Science, University of Illinois, 201 N. Goodwin Avenue, Urbana,
IL 61801, USA
e-mail: sariel@uiuc.edu
url: http://www.uiuc.edu/~sariel/

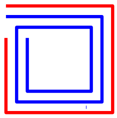
C. Wenk
Department of Computer Science, University of Texas at San Antonio, One UTSA Circle,
San Antonio, TX 78249-0667, USA
e-mail: carola@cs.utsa.edu

handwriting recognition [22, 23], matching of time series in databases [20], as well as geographic applications, such as map-matching of vehicle tracking data [8, 24], and moving objects analysis [10, 11].

Informally, the Fréchet distance between two curves is the maximum distance a point on the first curve has to travel as this curve is being continuously deformed into the second curve, see Sect. 2.2 for the formal definition. Unlike the Hausdorff distance, which is solely based on nearest neighbor distances between points on the curves, the Fréchet distance requires continuous and order-preserving assignments of points and hence is better suited for comparing curves with respect to their intrinsic structure.



The Fréchet distance between two curves might be arbitrarily larger than their Hausdorff distance, as demonstrated by the figure below, and as this example shows, it seems to be a more natural measure of similarity between curves.



Previous Results For two polygonal curves of total complexity n in the plane, their Fréchet distance can be computed in $O(n^2 \log n)$ time [3], and their Hausdorff distance can be computed in $O(n \log n)$ time [2]. It has been an open problem to find a subquadratic algorithm for computing the Fréchet distance for two curves. For the problem of deciding whether the Fréchet distance between two curves is smaller or equal a given value a lower bound of $\Omega(n \log n)$ was given by [9]. Recently, Alt [2] conjectured that the decision problem may be 3SUM-hard. The only subquadratic algorithms known are for quite restricted classes of curves such as for closed convex curves and for κ-bounded curves [4]. For a curve to be κ-bounded means, roughly, that for any two points on the curve the portion of the curve in between them cannot be further away from either point than $\kappa/2$ times the distance between the two points. For closed convex curves the Fréchet distance equals the Hausdorff distance and for κ-bounded curves the Fréchet distance is at most $(1+\kappa)$ times the Hausdorff distance, and hence the $O(n \log n)$ algorithm for the Hausdorff distance applies.

Aronov et al. [6] provided a near linear time $(1+\varepsilon)$-approximation algorithm for the *discrete* Fréchet distance, which only considers distances between vertices of the curves. Their algorithm works for *backbone curves*, which are used to model protein backbones in molecular biology. Backbone curves are required to have, roughly, unit edge length and a minimal distance between any pair of vertices. They use curve simplification to speed up their algorithm. Agarwal et al. [1] studied fast simplification that preserves the Fréchet distance.

The Input Model We introduce a new class of curves, called c-packed curves, for which we can approximate the Fréchet distance quickly, given that the constant c is small. Intuitively, the constant c measures how "*unrealistic*" the input is. We compare this new input model to previous models such as *fatness* and *low density*, as well as *κ-boundedness*. These so-called *realistic input models* are commonly used for the analysis of problems where the worst case complexity is dominated by degenerate or contrived configurations which are highly unlikely to occur in practice, see [15] for an overview.

A curve π is *c-packed* if the total length of π inside any ball is bounded by c times the radius of the ball. A κ-bounded curve might have arbitrary length while maintaining a finite diameter, and as such may not be c-packed, see Sect. 4.3. But unlike κ-bounded curves, the Fréchet distance between two c-packed curves might be arbitrarily larger than their Hausdorff distance. Indeed, c-packed curves are considerably more general and a more natural family of curves. For example, a c-packed curve might self cross and revisit the same location several times, and the class of c-packed curves is closed under concatenation, none of which is true for κ-bounded curves. Intuitively, c-packed curves behave reasonably in any resolution.

See the figure below for a few examples of c-packed curves. The boundary of convex polygons, algebraic curves of bounded maximum degree, the boundary of (α,β)-covered shapes [17], and the boundary of γ-fat shapes [14] are all c-packed. Indeed, the boundaries of (α,β)-covered shapes and γ-fat shapes are assumed to be formed by a constant number of algebraic curves of bounded maximum degree. If one removes the requirement that a γ-fat curve be of bounded descriptive complexity, then also fractal curves, like Koch's snowflake, which can have infinite length within a bounded area, can be fat [7]. Naturally, these curves cannot be c-packed. Interestingly, one can show that (α,β)-covered polygons are c-packed even if they have unbounded complexity, see Appendix A and also the result of Bose et al. [7].

It is easy to verify that c-packed curves are also low density [15], but a low-density curve might not be c-packed, for any bounded c, see Sect. 4.2. However, the class of c-packed curves is closed under simplification, see Lemma 4.3, and this is not true for low-density curves.

Our Results We present a new algorithm for computing a $(1+\varepsilon)$-approximation of the Fréchet distance for polygonal curves in $\mathbb{R}^d$. Underlying the algorithm are several new insights. First, we use the idea of curve simplification to reduce the complexity of the free space diagram, as this simplification results in a contraction of the corresponding rows or columns in the free space diagram. We introduce the notion

Table 1 Summary of new results for computing a $(1+\varepsilon)$-approximation to the Fréchet distance between two curves π and σ with n vertices in $\mathbb{R}^d$

Curves type	Running time	See
c-packed	$O(cn/\varepsilon + cn\log n)$	Theorem 4.5
κ-straight	Same as 2κ-packed	Lemma 4.16
κ-bounded	$O\big((\kappa/\varepsilon)^d n + \kappa^d n\log n\big)$	Theorem 4.20
$O(1)$-low density	$O\left(\frac{n^{2(d-1)/d}}{\varepsilon^2} + n^{2(d-1)/d}\log n\right)$	Theorem 4.14
c-packed & closed	$O\big(c^2 n/\varepsilon^2 + c^2 n\log n\big)$	Theorem 5.5

of *relative free space complexity* in Definition 3.3 to capture the complexity of the free space diagram of two curves, which are simplified to the appropriate resolution. Surprisingly, without simplification, almost any two curves from natural families of curves can have a free space diagram for the value realizing the Fréchet distance that has quadratic complexity (even in the plane). Secondly, we present an efficient construction algorithm for this reduced size free space diagram that enables us to solve the decision problem in linear time in the relative free space complexity of the curves. Thirdly, we prove that monotonicity events are sufficiently close to vertex–edge events or an approximate distance between two vertices of the curves. Therefore, the search for the Fréchet distance can be done efficiently without using parametric search or random sampling, by using approximate distance selection. Carefully combining these insights yields the new algorithm, which has running time near linear in the relative free space complexity of the input curves.

In the second part of the paper, we analyze the relative free space complexity for various families of curves. We prove that c-packed curves have linear relative free space complexity for fixed c and ε. We next prove a subquadratic bound on the relative complexity of the free space of low-density curves. This relies on a new packing lemma showing that, if the simplification of a low-density curve is long inside a relatively small area, then the original curve must contain many vertices in the vicinity of this region. We also prove that the relative free space complexity of κ-bounded curves is linear for a fixed κ, which leads to an improvement of the result by Alt et al. [4].

These bounds imply that the approximation algorithm provides fast approximation for the Fréchet distance for all these types of curves. We also show how to adapt our algorithm to handle closed curves. The new results are summarized in Table 1.

Organization In Sect. 2, we provide some background on the Fréchet distance and the notion of the free space diagram. In Sect. 3, we describe the approximation algorithm that uses simplification. To this end, we show in Sect. 3.1 that it suffices to only compute the reachable parts of the free space diagram and in Sect. 3.2 we present a fuzzy decider procedure and show how it can be used to make exact decisions during a binary search for the Fréchet distance. In Sect. 3.3, we deal with the different subroutines used in the search for the Fréchet distance and in Sect. 3.4 we give the resulting general algorithm and analyze its correctness and running time, which is near linear in the relative free space complexity. In Sect. 4, we bound the relative free space complexity of various families of curves. In particular, in Sect. 4.1, we introduce the notion of c-packed curves, and study their behavior under simplification. In Sect. 4.3, we bound the relative free space complexity of κ-bounded curves, and in Sect. 4.2 we handle low-density curves. In Sect. 5, we extend the algorithm to closed curves. We conclude with discussion in Sect. 6.

2 Preliminaries

2.1 Notations and Definitions

Let π be a curve in $\mathbb{R}^d$; that is, a continuous mapping from $[0,1]$ to $\mathbb{R}^d$. In the following, we will identify π with its range $\pi([0,1]) \subseteq \mathbb{R}^d$ if it is clear from the context. The curve π is *closed* if $\pi(0)=\pi(1)$. We use $\|\cdot\|$ to denote the Euclidean distance as well as the length of a curve. For a polygonal curve π, let $V(\pi)$ denote the set of vertices of π. For two points p and q on a curve π, let $\pi[p,q]$ denote the portion of the curve between the two points.

We denote with $B(p,r)$ the ball of radius r centered at p, and $S(p,r)$ denotes the corresponding sphere. Given a set of numbers $U \subseteq \mathbb{R}$, an *atomic interval* of U is a (possibly infinite) maximal interval on the real line that does not contain any point of U in its interior. Let $\mathcal{D}(\mathsf{P})$ be the set of all pairwise distances of points in P.

2.2 Fréchet Distance and the Free Space Diagram

A *reparameterization* is a bijective and continuous function $f:[0,1]\to[0,1]$. It is *orientation-preserving* if $f(0)=0$ and $f(1)=1$. Given two reparameterizations f and g for two curves π and σ, respectively, define their *width* as

$$\mathrm{width}_{f,g}(\pi,\sigma)=\max_{s\in[0,1]}\bigl\|\pi\bigl(f(s)\bigr)-\sigma\bigl(g(s)\bigr)\bigr\|.$$

This can be interpreted as the maximum length of a leash one needs to walk a dog, where the dog walks monotonically along π according to f, while the handler walks monotonically along σ according to g. In this analogy, the Fréchet distance is the shortest possible leash admitting such a walk.

Formally, given two curves π and σ in $\mathbb{R}^d$, the *Fréchet distance* between them is

$$d_{\mathcal{F}}(\pi,\sigma)=\inf_{\substack{f:[0,1]\to[0,1]\\ g:[0,1]\to[0,1]}}\mathrm{width}_{f,g}(\pi,\sigma),$$

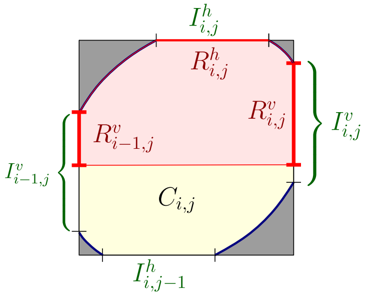
where f and g are orientation-preserving reparameterizations of the curves π and σ, respectively. The Fréchet distance complies with the triangle inequality; that is, for any three curves π,σ and τ we have $d_{\mathcal{F}}(\pi,\tau)\le d_{\mathcal{F}}(\pi,\sigma)+d_{\mathcal{F}}(\sigma,\tau)$.

Let π, σ be curves and $\delta>0$ a parameter, the *free space* of π and σ of radius δ is defined as

$$D_{\le\delta}(\pi,\sigma)=\bigl\{(s,t)\in[0,1]^2 \bigm| \bigl\|\pi(s)-\sigma(t)\bigr\|\le\delta\bigr\}.$$

We are interested only in polygonal curves. Then the square $[0,1]^2$ can be broken into a (not necessarily uniform) grid called the *free space diagram*, where a vertical line corresponds to a vertex of π and a horizontal line corresponds to a vertex of σ. Every two segments of π and σ define a *free space cell* in this grid. In particular, let $C_{i,j} = C_{i,j}(\pi, \sigma)$ denote the free space cell that corresponds to the ith edge of π and the jth edge of σ. The cell $C_{i,j}$ is located in the ith column and jth row of this grid.

It is known that the free space, for a fixed δ, inside such a cell $C_{i,j}$ (i.e., $D_{\leq\delta}(\pi, \sigma) \cap C_{i,j}$) is the clipping of an affine transformation of a disk to the cell [3], see the figure below; as such, it is convex and of constant complexity. Let $I^h_{i,j}$ denote the horizontal *free space interval* at the top boundary of $C_{i,j}$, and $I^v_{i,j}$ denote the vertical free space interval at the right boundary.



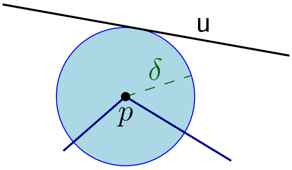
The Fréchet distance between π and σ is at most δ if and only if there is an (x, y)-monotone path in the free space diagram between $(0, 0)$ and $(1, 1)$ that is fully contained in $D_{\leq\delta}(\pi, \sigma)$. Let the *reachability intervals* $R^h_{i,j} \subseteq I^h_{i,j}$ and $R^v_{i,j} \subseteq I^v_{i,j}$ consist of the points (x, y) on the boundary that are reachable by a monotone path from $(0, 0)$ to (x, y).

Such a path to $(1, 1)$ can be computed, if it exists, in $O(n^2)$ time by dynamic programming, where n is the total complexity of the two polygonal curves π and σ, see [3].

2.2.1 Free Space Events

To compute the Fréchet distance consider increasing δ from 0 to ∞. As δ increases, structural changes to the free space happen. We are interested in the radii (i.e., the value of δ) of these events.

Consider a segment u of π and a vertex p of σ, a *vertex–edge event* corresponds to the minimum value δ such that u is tangent to $B(p, \delta)$. In the free space diagram, this corresponds to the event that a free space interval that consists of only one point was just created. The line supporting this boundary edge corresponds to the vertex, and the other dimension corresponds to the edge. Naturally, the event could happen at a vertex of u.



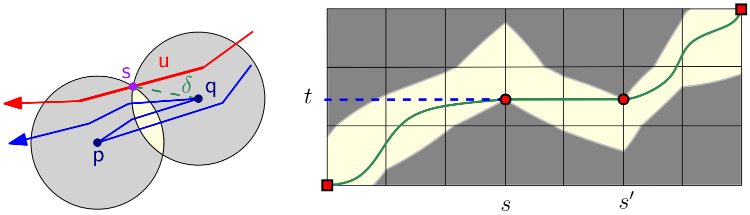


Fig. 1 Two curves π and σ and their free space diagram $D_{\le\delta}(\pi,\sigma)$, where $p=\pi(s)$, $q=\pi(s')$ and $r=\sigma(t)$. Here, δ is the minimal free space parameter, such that a monotone path exists, i.e., in this example $d_{\mathcal{F}}(\pi,\sigma)$ coincides with a monotonicity event

The second type of event, a *monotonicity event*, corresponds to a value δ for which a monotone subpath inside $D_{\le\delta}$ becomes feasible, see Fig. 1. Geometrically, this corresponds to two vertices p and q on one curve and a directed segment u on the other curve such that: (1) u passes through the intersection $S(p,\delta)\cap S(q,\delta)$, and (2) u intersects $B(q,\delta)$ first and $B(p,\delta)$ second, where p comes before q in the order along the curve π.

Other values of δ that would be relevant to our algorithm are the distances between any pair of points of $V(\pi)\cup V(\sigma)$. Technically, apart from the two single events that the endpoints of the curves are being matched to each other, these *vertex–vertex* events are vertex–edge events when they are relevant, but they will be handled naturally by our algorithm.

2.3 Curve Simplification

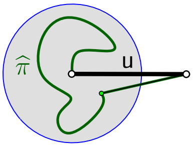
We suggest a straightforward greedy algorithm for curve simplification, which is sufficient for our purposes. We comment that Agarwal et al. [1] suggested a more aggressive (but slightly slower and more complicated) simplification algorithm that can be used instead.

Algorithm 2.1 Given a polygonal curve $\pi=p_1p_2p_3\dots p_k$ and a parameter $\mu>0$, consider the following simplification algorithm: First mark the initial vertex p_1 and set it as the current vertex. Now scan the polygonal curve from the current vertex until it reaches the first vertex p_i that is in distance at least μ from the current vertex. Mark p_i and set it as the current vertex. Repeat this until reaching the final vertex of the curve, and also mark this final vertex. Consider the curve that connects only the marked vertices, in their order along π. We refer to the resulting curve $\pi'=\operatorname{simpl}(\pi,\mu)$ as the *μ-simplification* of π. Note that this simplification can be computed in linear time.

Remark 2.2 The simplified curve has the useful property that all its segments are of length at least μ, except for the last edge that might be shorter. For the sake of simplicity of exposition, we assume that the last segment in the simplified curve also has length at least μ. Our arguments can be easily modified to handle this more general case.

Lemma 2.3 *For any polygonal curve π in $\mathbb{R}^d$, and $\mu \geq 0$, it holds $d_{\mathcal{F}}(\pi, \mathrm{simpl}(\pi, \mu)) \leq \mu$.*

Proof Consider a segment u of $\mathrm{simpl}(\pi, \mu)$ and the portion $\widehat{\pi}$ of π that corresponds to it. Clearly, all the vertices of $\widehat{\pi}$ are contained inside a ball of radius μ centered at the first endpoint of u visited by π, except the last vertex of $\widehat{\pi}$. As such, one can parameterize u and $\widehat{\pi}$, such that initially the point stays on the vertex of u while visiting all vertices of $\widehat{\pi}$ (except the last one), and then simultaneously move in sync on u and the last segment of $\widehat{\pi}$, in such a way that the distance is always at most μ.



□

3 The Approximation Algorithm

3.1 Computing the Reachable Free Space

For two curves π and σ, their *reachable free space*, denoted by $\mathcal{R}_{\leq\delta}(\pi, \sigma)$, is the set of all the points of $D_{\leq\delta}(\pi, \sigma)$ that are reachable from $(0, 0)$ by an (x, y)-*monotone* path.

The set $\mathcal{R}_{\leq\delta}$ has finite descriptive complexity inside each grid cell, and we need to describe it only for the grid cells that have non-empty intersection with $\mathcal{R}_{\leq\delta}$. Clearly, generating only those grid cells is sufficient to decide if there is a monotone path between $(0, 0)$ and $(1, 1)$, which is equivalent to deciding if the Fréchet distance between π and σ is smaller or equal to δ. In particular, to fully describe $\mathcal{R}_{\leq\delta}$, we will specify the reachability intervals $R^h_{i,j} \subseteq I^h_{i,j}$ and $R^v_{i,j} \subseteq I^v_{i,j}$ for each cell $C_{i,j}$, which describe the intersection of $\mathcal{R}_{\leq\delta}$ with the top and right boundary of $C_{i,j}$. These intervals contain all the needed information, since $\mathcal{R}_{\leq\delta} \cap C_{i,j}$ is convex.

The *complexity* of the reachable free space, for distance δ, denoted by $N_{\leq\delta}(\pi, \sigma)$, is the total number of grid cells which have non-empty intersection with $\mathcal{R}_{\leq\delta}$. One can compute this set of cells and extract an existing monotone path in $O(N_{\leq\delta}(\pi, \sigma))$ time, by performing a BFS of the grid cells that visits only the reachable cells. This yields the following relatively easy result. We include the details both for the sake of completeness and because the algorithm we suggest is engagingly simple.

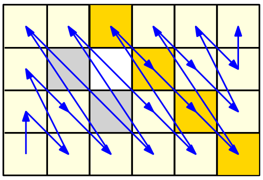
Lemma 3.1 *Given two polygonal curves π and σ in $\mathbb{R}^d$, and a parameter $\delta \geq 0$, one can compute a representation of $\mathcal{R}_{\leq\delta}(\pi, \sigma)$ in $O(N_{\leq\delta}(\pi, \sigma))$ time. Furthermore, one can decide if $d_{\mathcal{F}}(\pi, \sigma) \leq \delta$, and if this is the case also extract reparameterizations in $O(N_{\leq\delta}(\pi, \sigma))$ time.*

Proof We create a directed graph G that has a node $v(i, j)$ for every reachable free space cell $C_{i,j}$. With each node $v(i, j)$ we store the free space intervals $I^h_{i,j}$ and $I^v_{i,j}$ as well as the reachability intervals $R^h_{i,j} \subseteq I^h_{i,j}$ and $R^v_{i,j} \subseteq I^v_{i,j}$.

Each node $v(i,j)$ can have an outgoing edge to its right and top neighbor; an edge between these vertices exists if and only if the corresponding reachability interval between them is nonempty. In particular, a monotone path from $(0,0)$ to a point $(x,y) \in C_{i,j}$ in $\mathcal{R}_{\leq\delta}$ corresponds to a monotone path in the graph G from $v(1,1)$ to $v(i,j)$. Furthermore, any such monotone path has exactly $k = i + j - 2$ edges on it.

We compute the graph G on the fly by performing a BFS on it, starting from $v(1,1)$, and keeping the invariant that when the BFS visits a node $v(i,j)$ it enqueues the vertices $v(i, j+1)$ and $v(i+1, j)$, in this order, to the BFS queue (if they are connected to $v(i,j)$, naturally).

This implies that at any point in time, and for any k, the BFS queue contains the nodes on the kth diagonal (i.e., all nodes $v(i,j)$ such that $i+j = k-1$) of the diagram sorted from left to right. However, the same node might appear twice (consecutively) in this queue.



In every iteration, the BFS dequeues the one or two copies of the same node $v(i,j)$ and merges the two copies of the same vertex into one if necessary. Now, the one or two vertices (i.e., $v(i-1,j)$ and $v(i,j-1)$) that have incoming edges to $v(i,j)$ are known, as are their reachability intervals. Therefore one can compute the reachability intervals for $v(i,j)$ in constant time. Now, $v(i,j+1)$ is enqueued if and only if the top side of the cell $C_{i,j}$ is reachable by a monotone path (i.e., $R^h_{i,j} \neq \emptyset$), and $v(i+1,j)$ is enqueued if and only if the right side of the cell $C_{i,j}$ is reachable by a monotone path (i.e., $R^v_{i,j} \neq \emptyset$). Since $\mathcal{R}_{\leq\delta}(\pi,\sigma) \cap C_{i,j}$ is convex and of constant complexity, this can be done in constant time.

Clearly, the BFS takes time linear in the size of G and it computes the reachability information for all reachable free space cells of $\mathcal{R}_{\leq\delta}(\pi,\sigma)$. Now, one can check if $(1,1)$ is reachable by inspecting the reachability intervals for $C_{\mathsf{n}_\pi - 1, \mathsf{n}_\sigma - 1}$, and checking if the top right corner of this cell is monotonically reachable from the origin, where n_π is the number of vertices of the curve π. The monotone path realizing this can be extracted in linear time, by introducing backward edges in the graph and tracing a path back to the origin. □

Observation 3.2 One can compute all relevant vertex–edge events with radius $\leq \delta$ in $O(N_{\leq\delta}(\pi,\sigma))$ time as follows. We compute the graph representation of $\mathcal{R}_{\leq\delta}(\pi,\sigma)$ using Lemma 3.1. Next, for each reachable cell consider the vertex–edge events at its top and right boundaries and compute their event radii. Recall that a cell boundary corresponds to an edge from the one curve and a vertex from the other curve. Clearly, any cell boundary can be used by the reparameterization of width $\leq \delta$, if and only if the corresponding event radius is smaller or equal δ.

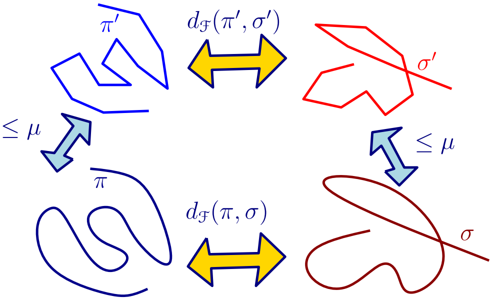

π'
$d_{\mathcal{F}}(\pi', \sigma')$
σ'
$\leq \mu$
$\leq \mu$
π
$d_{\mathcal{F}}(\pi, \sigma)$
σ

Fig. 2 The idea of the fuzzy decision procedure using simplification

3.2 The Approximate Decision Procedure

In the following, we are interested in the maximum complexity of the reachable free space when considering any radius δ and simplifying the curves with radius $\varepsilon\delta$. The reasons will become apparent only shortly after, in Lemma 3.5 and Lemma 3.6, where we show that the simplification radius chosen this way enables us to either (i) compute a $(1+\varepsilon)$-approximation of the Fréchet distance, or (ii) solve the decision problem exactly using the simplified curves (see Sect. 3.3.5).

The idea underlying this approximate decision procedure is depicted in Fig. 2. We simplify the two input curves to a resolution that is (roughly) an ε-fraction of the radius we care about (i.e., δ), and we then use the exact decision procedure on these two simplified curves. Since the Fréchet distance complies with the triangle inequality and by Lemma 2.3, we can infer the original distance from this information. In order for this approach to work, the complexity of the reachable free space for the two simplified curves has to be small. This notion of complexity is captured by the following definition.

Definition 3.3 For two curves π and σ, let

$$\mathsf{N}(\varepsilon, \pi, \sigma) = \max_{\delta \geq 0} N_{\leq\delta}\bigl(\text{simpl}(\pi, \varepsilon\delta), \text{simpl}(\sigma, \varepsilon\delta)\bigr)$$

be the maximum complexity of the reachable free space for the simplified curves. We refer to $\mathsf{N}(\varepsilon, \pi, \sigma)$ as the *ε-relative free space complexity* of π and σ. In order to give a more informative analysis, we will express the asymptotic time complexity of our algorithms not in terms of the size of the input, but instead use the size of the input and the free space complexity of the input as parameters.

We assume that for any $0 < \varepsilon < 1$ the following properties hold for $\mathsf{N}(\cdot, \cdot, \cdot)$.

(P1) For any constant $c' \geq 1$, it holds $\mathsf{N}(\varepsilon/c', \pi, \sigma) = O(\mathsf{N}(\varepsilon, \pi, \sigma))$.
(P2) $\mathsf{N}(\varepsilon, \pi, \sigma) \leq \mathsf{N}(\varepsilon/2, \pi, \sigma)/2$.

The above properties will hold for all the families of curves we consider. In Sect. 4.1 we show that $\mathsf{N}(\varepsilon, \pi, \sigma)$ is a linear function in the number of vertices of the two curves for a fixed $\varepsilon > 0$ if the curves are sufficiently well-behaved (see for example Lemma 4.4). Combining this analysis with the time complexity analysis of the algorithms will yield near linear upper bounds on the running times of these algorithms for the classes of curves considered.

Remark 3.4 In the following, when we state the time complexity of our algorithms, we always assume that $\mathsf{N}(\varepsilon, \pi, \sigma) = \Omega(n)$, where n is the total number of vertices of π and σ.

Lemma 3.5 *Let π and σ be polygonal curves in $\mathbb{R}^d$, and let $\varepsilon > 0$ and $\delta > 0$ be two parameters. Then, the algorithm described below output, in $O(\mathsf{N}(\varepsilon, \pi, \sigma))$ time, one of the following*:

(A) *"$d_{\mathcal{F}}(\pi, \sigma) \le (1+\varepsilon)\delta$", and reparameterizations of π and σ of width $\le (1+\varepsilon)\delta$, and this happens if $d_{\mathcal{F}}(\pi, \sigma) \le \delta$.*
(B) *"$d_{\mathcal{F}}(\pi, \sigma) > \delta$" if $d_{\mathcal{F}}(\pi, \sigma) > (1+\varepsilon)\delta$.*
(C) *If $d_{\mathcal{F}}(\pi, \sigma) \in (\delta, (1+\varepsilon)\delta]$ then the algorithm outputs either of the above outcomes.*

In either case, the statement returned is correct.

Proof Set $\mu = (\varepsilon/4)\delta$. Compute in linear time the curves $\pi' = \mathrm{simpl}(\pi, \mu)$ and $\sigma' = \mathrm{simpl}(\sigma, \mu)$ using Algorithm 2.1. Let $\delta' = \delta + 2\mu$ and observe that $\mu/\delta' = \varepsilon/(4+2\varepsilon)$. Using Lemma 3.1 we can decide whether $d_{\mathcal{F}}(\pi', \sigma') \le \delta'$ in

$$O\bigl(N_{\le\delta'}(\pi', \sigma')\bigr) = O\bigl(\mathsf{N}\bigl(\mu/\delta', \pi, \sigma\bigr)\bigr) = O\bigl(\mathsf{N}\bigl(\varepsilon/(4+2\varepsilon), \pi, \sigma\bigr)\bigr) = O\bigl(\mathsf{N}(\varepsilon, \pi, \sigma)\bigr)$$

time, by assumption (P1). If so, we output the reparameterizations as a proof that

$$\begin{aligned} d_{\mathcal{F}}(\pi, \sigma) &\le d_{\mathcal{F}}(\pi, \pi') + d_{\mathcal{F}}(\pi', \sigma') + d_{\mathcal{F}}(\sigma', \sigma) \\ &\le \delta' + 2\mu = \delta + 4(\varepsilon/4)\delta = (1+\varepsilon)\delta. \end{aligned}$$

On the other hand, if $d_{\mathcal{F}}(\pi', \sigma') > \delta'$, then this implies, by the triangle inequality, that

$$d_{\mathcal{F}}(\pi, \sigma) \ge d_{\mathcal{F}}(\pi', \sigma') - d_{\mathcal{F}}(\pi, \pi') - d_{\mathcal{F}}(\sigma', \sigma) > \delta' - 2\mu = \delta.$$

Therefore, the algorithm outputs "$d_{\mathcal{F}}(\pi, \sigma) > \delta$" in this case. □

3.2.1 How to Use the Approximate Decider in a Binary Search

In order to use Lemma 3.5 to perform a binary search for the Fréchet distance, we can turn the "fuzzy" decision procedure into a precise one as follows.

Lemma 3.6 *Let π and σ be two polygonal curves in $\mathbb{R}^d$, and let $1 \ge \varepsilon > 0$ and $\delta > 0$ be two parameters. Then, there is an algorithm* **`decider`**$(\pi, \sigma, \delta, \varepsilon)$ *that, in $O(\mathsf{N}(\varepsilon, \pi, \sigma))$ time, returns one of the following outputs*: (i) *a $(1+\varepsilon)$-approximation to $d_{\mathcal{F}}(\pi, \sigma)$*, (ii) *$d_{\mathcal{F}}(\pi, \sigma) < \delta$, or* (iii) *$d_{\mathcal{F}}(\pi, \sigma) > \delta$. The answer returned is correct.*

Proof Let $\delta' = \delta/(1+\varepsilon')$, for $\varepsilon' = c\varepsilon$, $c = 1/3$. We run the algorithm of Lemma 3.5 with parameters δ and ε'. If the call returns "$d_{\mathcal{F}}(\pi, \sigma) > \delta$", then we return this result.

Otherwise, we call Lemma 3.5 with parameters δ' and ε'. If it returns that "$d_{\mathcal{F}}(\pi, \sigma) \le (1+\varepsilon')\delta'$" then $d_{\mathcal{F}}(\pi, \sigma) \le (1+\varepsilon')\delta' = \delta$, and we return this result.

The only remaining possibility is that the two calls returned "$d_{\mathcal{F}}(\pi,\sigma) \leq (1+\varepsilon')\delta$" and "$d_{\mathcal{F}}(\pi,\sigma) > \delta'$". But then we have found the required approximation. Therefore, the resulting approximation factor of the reparameterizations returned by the call with δ is $\leq \frac{(1+\varepsilon')\delta}{\delta'} = (1+c\varepsilon)^2 < (1+\varepsilon)$ as can be easily verified, since $0 < \varepsilon \leq 1$. □

3.3 Searching for the Fréchet Distance

3.3.1 Searching in a Fixed Interval

It is now straightforward to perform a binary search on an interval $[\alpha,\beta]$ to approximate the value of the Fréchet distance, if it falls inside this interval. Indeed, partition this interval into subintervals of length $\varepsilon\alpha$ and perform a binary search to find the interval that contains the Fréchet distance. There are $O(\beta/\varepsilon\alpha)$ intervals, and this would require $O(\log(\beta/\varepsilon\alpha))$ calls to **decider**. By using exponential subintervals, one can do slightly better, as testified by the following lemma.

Lemma 3.7 *Given two curves π and σ in $\mathbb{R}^d$, a parameter $1 \geq \varepsilon > 0$, and an interval $[\alpha,\beta]$, one can perform a binary search in $[\alpha,\beta]$ and obtain a $(1+\varepsilon)$-approximation to $d_{\mathcal{F}}(\pi,\sigma)$ if $d_{\mathcal{F}}(\pi,\sigma) \in [\alpha,\beta]$, or report that $d_{\mathcal{F}}(\pi,\sigma) \notin [\alpha,\beta]$. The algorithm, denoted by* **search Interval**$(\pi,\sigma,[\alpha,\beta],\varepsilon)$, *takes $O(\log \frac{\log(\beta/\alpha)}{\varepsilon})$ calls to* **decider**.

Proof Let $\alpha_i = \alpha(1+\varepsilon)^i$ for $i = 0,\dots,M = \lfloor \log_{1+\varepsilon}(\beta/\alpha) \rfloor$ and $\alpha_{M+1} = \beta$. Perform a binary search, using **decider**$(\pi,\sigma,\delta,\varepsilon)$ to find the two values α_i and α_{i+1} such that $\alpha_i \leq \delta = d_{\mathcal{F}}(\pi,\sigma) \leq \alpha_{i+1}$. Since $\alpha_{i+1} = (1+\varepsilon)\alpha_i$, we conclude that we found the required approximation.

It might be that during this procedure one of the calls to **decider**$(\pi,\sigma,\delta,\varepsilon)$ found the required approximation, and in this case we abort the binary search and just return this approximation.

This process requires $O(\log M) = O(\log\log_{1+\varepsilon}(\beta/\alpha))$ calls to **decider**. Observe that

$$M = \log_{1+\varepsilon}\frac{\beta}{\alpha} = \frac{\ln(\beta/\alpha)}{\ln(1+\varepsilon)} = O\left(\frac{1}{\varepsilon}\log\frac{\beta}{\alpha}\right).$$

Indeed, $e^{x/2} \leq 1+x \leq e^x$ for $x \in [0,1]$, and this implies that $x/2 \leq \ln(1+x) \leq x$, which is the inequality used above. □

3.3.2 Searching over Events

Clearly, the procedure **searchInterval**$(\pi,\sigma,[\alpha,\beta],\varepsilon)$ alone does not suffice to solve our main problem, since the interval of distances we are searching over might have arbitrarily large "spread" (i.e., $\log\beta/\alpha$ might be arbitrarily large). However, the Fréchet distance must be sufficiently close to a free space event in one of the "approximate" diagrams, i.e., a free space diagram of the two simplified curves. Thus, we can identify two kinds of critical value to search over, which are candidate values for the approximate Fréchet distance. These are the events where (i) the simplification of an input curve changes, or (ii) the reachability within the approximate free space diagram changes (i.e., a free space event; see Sect. 2.2.1).

The traditional solution to overcome this problem is to use parametric search. However, in our case, since we are only interested in approximation, we can use a simpler, "approximate", search. It is sufficient to search over a set of values which approximate the event values by a constant factor, since we will use Lemma 3.7 to refine the resulting search interval in the main algorithm. Note, for instance, that we can easily use this lemma to turn a constant factor approximation of the Fréchet distance into a $(1+\varepsilon)$-approximation.

Algorithm 3.8 Let **`searchEvents`**(π, σ, Z) denote the algorithm that performs a binary search over the values of Z, to compute the atomic interval of Z that contains the Fréchet distance between π and σ. This procedure uses **`decider`** (Lemma 3.6) to perform the decisions during the search.

3.3.3 Searching over Simplifications

Consider the events when the simplified curves change, see Algorithm 2.1. Consider the set of all pairwise distances between vertices of π and σ. Observe that it breaks the real line into $\binom{n}{2}+1$ atomic intervals, such that in each such interval the simplification does not change. Thus $\text{simpl}(\pi, \mu)$ (resp. $\text{simpl}(\sigma, \mu)$) might result in $O(n^2)$ different curves depending on the value of μ, where n is the total number of vertices of π and σ. As a first step we would therefore like to use Algorithm 3.8 to perform a binary search over those distances to find the atomic interval that contains the required Fréchet distance. Naively, this would require us to perform distance selection. However, it is believed that exact distance selection requires $\Omega(n^{4/3})$ time in the worst case [18]. To overcome this we will perform an approximate distance selection, as suggested by Aronov et al. [6].

Lemma 3.9 *Given a set* P *of* n *points in* $\mathbb{R}^d$. *Then, one can compute in* $O(n \log n)$ *time a set* Z *of* $O(n)$ *numbers, such that for any* $y \in \mathcal{D}(\mathsf{P})$, *there exist numbers* x, $x' \in Z$ *such that* $x \le y \le x' \le 2x$. *Let* **`approxDistances`**(P) *denote this algorithm.*

Proof Compute an 8-well-separated pairs decomposition of P. Using the algorithm of Callahan and Kosaraju [12] this can be done in $O(n \log n)$ time, and results in a set of pairs of subsets $\{(X_1, Y_1), \dots, (X_m, Y_m)\}$, where $m = O(n)$, such that for any two points $p, q \in \mathsf{P}$ there exists a pair (X_i, Y_i) in the above decomposition, such that: (i) $p \in X_i$ and $q \in Y_i$ (or vice versa), and (ii) $\max(\text{diam}(X_i), \text{diam}(Y_i)) \le \min_{p_i \in X_i, q_i \in Y_i} \|p_i - q_i\|/8$.

This implies that the distance of any pair of points in X_i and Y_i, respectively, are the same up to a small constant. As such, for every pair (X_i, Y_i), for $i = 1, \dots, m$, we pick representative points $p_i \in X_i$ and $q_i \in Y_i$, and set $\ell_i = (3/4)\|p_i - q_i\|$. Let $Z = \{\ell_1, \dots, \ell_m, 2\ell_1, \dots, 2\ell_m\}$ be the computed set of values.

Consider any pair of points $p, q \in \mathsf{P}$. For the specific pair (X_i, Y_i) that contains the pair of points p and q that we are interested in, we have $\ell_i = (3/4)\|p_i - q_i\| \le$

$\|p_i - q_i\| - \mathrm{diam}(X_i) - \mathrm{diam}(Y_i) \le \|p - q\| \le \|p_i - q_i\| + \mathrm{diam}(X_i) + \mathrm{diam}(Y_i) \le (5/4)\|p_i - q_i\| \le 2\ell_i$, thus establishing the claim. □

3.3.4 *Monotonicity Events*

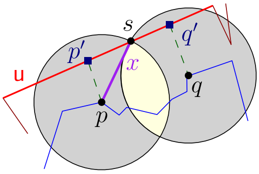
The following lemma testifies that the radius of a monotonicity event must be "close" to either a vertex–edge event or to the distance between two vertices. Since we will approximate the vertex–vertex distances and perform a binary search over them, this implies that we further only need to consider vertex–edge events. Furthermore, by Observation 3.2, the number of those vertex–edge events which remain in the resulting search range can be bounded by the complexity of the reachable free space.

Lemma 3.10 *Let x be the radius of a monotonicity event involving vertices p, q and a segment u. Then there exists a number y such that $y/2 \le x \le 3y$, and y is either in $\mathcal{W} = \mathcal{D}(V(\pi) \cup V(\sigma))$ or y is the radius of a vertex–edge event.*

Proof Let s be the intersection point of $S(p, x) \cap S(q, x)$ which lies on u. Let p' (resp. q') be the closest point on u to p (resp. q).

Clearly $\|p' - q'\| \le \|p - q\|$ (since the projection onto the nearest neighbor of a convex set is a contraction), and since $p' \in B(p, x)$ and $q' \in B(q, x)$, the point s lies on the segment $p'q'$.

This implies that $x = \|p - s\| \le \|p - p'\| + \|p' - s\| \le \|p - p'\| + \|p' - q'\| \le \|p - p'\| + \|p - q\|$, by the triangle inequality.



A similar argument implies that

$$\begin{aligned} x &= \|p - s\| \ge \|p - p'\| - \|p' - s\| \ge \|p - p'\| - \|p' - q'\| \\ &\ge \|p - p'\| - \|p - q\|. \end{aligned}$$

If $\|p - p'\| \ge 2\|p - q\|$ then the above implies that $x \in [1/2, 3/2]\|p - p'\|$. If p' is an endpoint of u then $\|p - p'\|$ is in $\mathcal{W}$. Otherwise, $\|p - p'\|$ is the radius of the vertex–edge event between p and u. In either case, this implies the claim.

If $\|p - p'\| \le 2\|p - q\|$ then $x = \|p - s\| \le \|p - p'\| + \|p - q\| \le 2\|p - q\| + \|p - q\| = 3\|p - q\|$, and of course $\|p - q\| \in \mathcal{W}$. Now, the two balls of radius x centered at p and q, respectively, cover the segment pq, and we have $\|p - q\|/2 \le x$, which implies the claim. □

3.3.5 *Searching with a Fixed Simplification*

Assume that we have found simplifications τ and η, such that the Fréchet distance of those curves yields the desired $(1 + \varepsilon)$-approximation. Clearly, an approximation of $d_{\mathcal{F}}(\tau,\eta)$ suffices for our result. To this end, let $\mathtt{searchIntervalNoSimp}(\pi,\sigma,[\alpha,\beta],\varepsilon)$ be the variant of **`searchInterval`** from Lemma 3.7 that uses Lemma 3.1 directly instead of calling **`decider`**. This version searches for the Fréchet distance in the given interval, but does not perform simplification before calling the decision procedure. It returns a $(1+\varepsilon)$-approximation of the Fréchet distance, given that it is contained in this interval. Note that correctness and running time of Lemma 3.7 are not affected by this modification.

Lemma 3.11 *Let τ and η be two given curves in $\mathbb{R}^d$, with total complexity n, and let $[h^-,h^+]$ be an interval, such that* (i) *$d_{\mathcal{F}}(\tau,\eta)\in[h^-,h^+]$, and* (ii) *there is no value of $\mathcal{W}=\mathcal{D}(V(\tau)\cup V(\eta))$ in the interval $[h^-,h^+]$. Then, for $\varepsilon>0$, one can $(1+\varepsilon)$-approximate $d_{\mathcal{F}}(\tau,\eta)$ and compute reparameterizations in $O((n+N)\log(N/\varepsilon))$ time, where $N=N_{\le h^+}(\tau,\eta)$.*

Let $\mathtt{aprxFréchetNoSimp}(\tau,\eta,[h^-,h^+],\varepsilon)$ *denote this algorithm.*

Proof For two real numbers $x,y>0$, we define $[x/y]=\max(x,y)/\min(x,y)$.

Compute $\mathcal{R}_{\le h^+}(\tau,\eta)$, using Lemma 3.1. Next, using Observation 3.2, compute from $\mathcal{R}_{\le h^+}(\tau,\eta)$ the set Z of all the radii of the vertex–edge events of τ and η with radius at most h^+. Next, we sort Z, and perform a binary search over Z, using Lemma 3.1, for the atomic interval $\mathcal{I}=[\alpha,\beta]$ of Z that contains the Fréchet distance $d_{\mathcal{F}}(\tau,\eta)$. Next, call $\mathtt{searchIntervalNoSimp}(\tau,\eta,[\alpha,4\alpha],\varepsilon)$ and $\mathtt{searchIntervalNoSimp}(\tau,\eta,[\beta/4,\beta],\varepsilon)$. We claim that one of these two searches performed on the respective intervals will discover two consecutive values x and $(1+\varepsilon)x$, such that the two corresponding calls to the algorithm of Lemma 3.7 imply that $d_{\mathcal{F}}(\tau,\eta)\in[x,(1+\varepsilon)x]$.

Indeed, the interior of $[\alpha,\beta]$ does not contain any value in $\mathcal{W}$ or a radius of a vertex–edge event of τ and η. Therefore, the interval $[\alpha,\beta]$ might contain only monotonicity events of τ and η. By Lemma 3.10, for a monotonicity event with radius r there exists a $y\in Z\cup\mathcal{W}$, such that $[\mathsf{r}/y]\le 3$. But since there is no value of $Z\cup\mathcal{W}$ in the interior of $[\alpha,\beta]$, and therefore, for any $\mathsf{r}''\in[4\alpha,\beta/4]$ and $y''\in Z\cup\mathcal{W}$, we have $[\mathsf{r}''/y'']\ge 4$.

We conclude that no monotonicity event, vertex–edge event, or value of $\mathcal{W}$ lies in the interval $[4\alpha,\beta/4]$. Since the Fréchet distance must be equal to one such value, it follows that $d_{\mathcal{F}}(\tau,\eta)\notin(4\alpha,\beta/4)$, but this implies that either $d_{\mathcal{F}}(\tau,\eta)\in[\alpha,4\alpha]$ or $d_{\mathcal{F}}(\tau,\eta)\in[\beta/4,\beta]$. In either case, the above algorithm would have found the approximate distance.

Computing and sorting the set of vertex–edge events takes $O(N\log N)$ time by Observation 3.2. The binary search requires $O(\log|Z|)$ calls to the algorithm of Lemma 3.1. The two calls to **`searchIntervalNoSimp`** require $O(\log(1/\varepsilon))$ calls to Lemma 3.1. Now, observe that all these calls to the algorithm of Lemma 3.1 are done with values of $\delta\le h^+$. Thus the complexity of the reachable free space is bounded (up to a constant factor) by the number of vertex–edge events of values $\le h^+$, and this number is bounded by $|Z|$. Therefore, a call to Lemma 3.1 takes $O(|Z|)$ time. Thus, the overall running time is $O((n+|Z|)\log(|Z|/\varepsilon))$, and by definition $|Z|=O(N_{\le h^+}(\tau,\eta))$. □

```
aprxFréchetI($\pi, \sigma, \varepsilon$ )
    (A) $\mathsf{P} = V(\pi) \cup V(\sigma)$
    (B) $Z \leftarrow$ approxDistances($\mathsf{P}$) (Lemma 3.9).
    (C) $[\alpha, \beta] \leftarrow$ searchEvents($\pi, \sigma, Z, \varepsilon$) (Algorithm 3.8).
    (D) Call searchInterval($\pi, \sigma, [\alpha, 4\alpha'], \varepsilon$), where $\alpha' = (30/\varepsilon)\alpha$
        (Lemma 3.7).
    (E) Call searchInterval($\pi, \sigma, [\beta'/4, \beta], \varepsilon$), where $\beta' = \beta/3$.
    (F) Let $\pi' = \mathrm{simpl}(\pi, \mu)$ and $\sigma' = \mathrm{simpl}(\sigma, \mu)$, for $\mu = 3\alpha$ (Algorithm 2.1)
    (G) $\delta \leftarrow$ aprxFréchetNoSimp($\pi', \sigma', [\alpha', \beta'], \varepsilon/4$) (Lemma 3.11).
    (H) Compute and return the resulting reparameterizations of $\pi$ and $\sigma$ and their
        width as the approximation.
```

Fig. 3 The basic approximation algorithm

3.4 The Approximation Algorithm

The resulting approximation algorithm is depicted in Fig. 3. It will be used by the final approximation algorithm as a subroutine. We first analyze this basic algorithm. We will then show how to use it, in Lemma 3.15 below, to get a faster approximation algorithm. The algorithm depicted in Fig. 3 performs numerous calls to **decider**, with approximation parameter $\varepsilon > 0$. If any of these calls discover the approximate distance, then the algorithm immediately stops and returns the approximation. Therefore, at any point in the execution of the algorithm, the assumption is that all previous calls to **decider** returned a direction where the optimal distance must lie. In particular, a call to **searchInterval**$(\pi, \sigma, \mathcal{I}, \varepsilon)$, would either find the approximate distance in the interval $\mathcal{I}$ and return immediately, or the desired value is outside this interval.

3.4.1 Correctness

Lemma 3.12 *Given two polygonal curves π and σ, and a parameter $1 > \varepsilon > 0$, the algorithm* **aprxFréchetI**$(\pi, \sigma, \varepsilon)$ *computes a $(1+\varepsilon)$-approximation to $d_{\mathcal{F}}(\pi, \sigma)$.*

Proof If the algorithm found the approximation before step (F), then clearly it is the desired approximation, and we are done. (In particular, this must be the case if $4\alpha' > \beta'/4$.)

Otherwise, because of (C), we know that $d_{\mathcal{F}}(\pi, \sigma) \in [\alpha, \beta]$. By steps (D) and (E) it must be that $d_{\mathcal{F}}(\pi, \sigma) \in [4\alpha', \beta'/4]$. Since $\mu = 3\alpha = (\varepsilon/10)\alpha' \le \beta'/4$, it follows, by the triangle inequality, that

$$d_{\mathcal{F}}(\pi', \sigma') \le d_{\mathcal{F}}(\pi', \pi) + d_{\mathcal{F}}(\pi, \sigma) + d_{\mathcal{F}}(\sigma, \sigma') \le 2\mu + \beta'/4 < \beta'.$$

A similar argument shows that $d_{\mathcal{F}}(\pi', \sigma') > \alpha'$. Hence, the algorithm of Lemma 3.11 can be applied to π' and σ' for the range $[\alpha', \beta']$, as $d_{\mathcal{F}}(\pi', \sigma') \in [\alpha', \beta']$.

Now, by Lemma 3.11, we find that the value δ resulting from step (G), is contained in the interval $[d_{\mathcal{F}}(\pi',\sigma'),(1+\varepsilon/4)d_{\mathcal{F}}(\pi',\sigma')]$. By the triangle inequality we conclude that the returned Fréchet distance is

$$\begin{aligned}\Delta &\le d_{\mathcal{F}}(\pi,\pi')+\delta+d_{\mathcal{F}}(\sigma,\sigma')\le d_{\mathcal{F}}(\pi,\pi')+(1+\varepsilon/4)d_{\mathcal{F}}(\pi',\sigma')+d_{\mathcal{F}}(\sigma',\sigma)\\ &\le(1+\varepsilon/4)\bigl(2\mu+d_{\mathcal{F}}(\pi,\sigma)+2\mu\bigr)\le 5\mu+(1+\varepsilon/4)d_{\mathcal{F}}(\pi,\sigma)\\ &\le(1+\varepsilon)d_{\mathcal{F}}(\pi,\sigma),\end{aligned}$$

since $5\mu=15\alpha=(\varepsilon/2)(30/\varepsilon)\alpha=(\varepsilon/2)\alpha'\le(\varepsilon/2)d_{\mathcal{F}}(\pi,\sigma)$.

Note that $\Delta\ge d_{\mathcal{F}}(\pi,\sigma)$ since it is the width of a specific reparameterization between the two curves. □

3.4.2 Running Time

Lemma 3.13 *For any* $x,y\in(2\alpha,\beta/2)$, *we have* $\mathrm{simpl}(\pi,x)=\mathrm{simpl}(\pi,y)$ *and* $\mathrm{simpl}(\sigma,x)=\mathrm{simpl}(\sigma,y)$.

Proof Indeed, the interval (α,β) does not contain any value of Z. As such, by Lemma 3.9, $(2\alpha,\beta/2)$ does not contain any value of the pairwise distances between vertices of the vertex set of π and σ which implies that the simplification is the same for any value inside this interval. □

Lemma 3.14 *Given two polygonal curves* π *and* σ *with a total of* n *vertices in* $\mathbb{R}^d$, *and a parameter* $1>\varepsilon>0$, *the running time of* **aprxFréchetI**(π,σ,ε) *is* $O(\mathsf{N}(\varepsilon,\pi,\sigma)\log n)$.

Proof Computing Z (and sorting it) takes $O(n\log n)$ time by Lemma 3.9. Steps (C), (D) and (E) perform $O(\log n+\log(1/\varepsilon))=O(\log n)$ calls to **decider**, by Lemma 3.7. (Here, we assume that $\varepsilon=\Omega(1/n)$. If $\varepsilon<1/n$ then we can just use the algorithm of Alt and Godau [3] since its running time is faster than our approximation algorithm in this case.) Each call to **decider** takes $O(\mathsf{N}(\varepsilon,\pi,\sigma))$ time, so overall this takes $O(\mathsf{N}(\varepsilon,\pi,\sigma)\log n)$ time. Computing the simplifications in step (F) with Algorithm 2.1 takes $O(n)$ time.

By Lemma 3.11, a call to **aprxFréchetNoSimp**$(\pi',\sigma',[\alpha',\beta'],\varepsilon/4)$ takes $T=O((n+N)\log(N/\varepsilon))$ time, with $N=N_{\le\beta'}(\pi',\sigma')$. Now, 3α and β' are both inside the interval $(2\alpha,\beta/2)$, and as such, by Lemma 3.13, we have $\pi'=\mathrm{simpl}(\pi,3\alpha)=\mathrm{simpl}(\pi,\beta')$ and $\sigma'=\mathrm{simpl}(\sigma,3\alpha)=\mathrm{simpl}(\sigma,\beta')$. Therefore, we have

$$N=N_{\le\beta'}(\pi',\sigma')=N_{\le\beta'}\bigl(\mathrm{simpl}\bigl(\pi,\beta'\bigr),\mathrm{simpl}\bigl(\sigma,\beta'\bigr)\bigr)\le\mathsf{N}(1,\pi,\sigma).$$

Thus, step (G) takes $T=O(\mathsf{N}(1,\pi,\sigma)\log(\mathsf{N}(1,\pi,\sigma)n/\varepsilon))=O(\mathsf{N}(1,\pi,\sigma)\log n)$, time since $\mathsf{N}(1,\pi,\sigma)\le n^2$ and $\varepsilon=\Omega(1/n)$. Observe that $\mathsf{N}(1,\pi,\sigma)\le\mathsf{N}(\varepsilon,\pi,\sigma)$ for $\varepsilon\le 1$.

Finally, in order to compute the resulting reparameterizations in step (H), we compute the reparameterizations of π and π' (resp. σ and σ') as described in the proof of Lemma 2.3 and chain them with the reparameterizations of the simplified curves, which we obtained from step (G). Clearly, this and computing the resulting width takes $O(n)$ time. Note that by the assumption in Remark 3.4 the term $\mathsf{N}(\varepsilon,\pi,\sigma)$ dominates over $O(n)$. □

The running time of Lemma 3.14 can be slightly improved.

Lemma 3.15 *The algorithm* **`aprxFréchetI`** *depicted in Fig.* 3 *can be modified to run in time* $O(\mathsf{N}(\varepsilon,\pi,\sigma)+\mathsf{N}(1,\pi,\sigma)\log n)$ (*see Definition* 3.3).

Proof Use Lemma 3.14, with $\varepsilon_0 = 1/2$, to get a 2-approximation ζ for the Fréchet distance between π and σ. This takes $O(\mathsf{N}(1,\pi,\sigma)\log n)$ time. Let $\mathcal{I}_0 = [\zeta, 2\zeta]$ be the corresponding interval that contains the distance. We could call **`searchInterval`**$(\pi,\sigma,\mathcal{I}_0,\varepsilon)$ and get a $(1+\varepsilon)$-approximation in $O(\mathsf{N}(\varepsilon,\pi,\sigma)\log\frac{1}{\varepsilon} + \mathsf{N}(1,\pi,\sigma)\log n)$ time.

One can do better by starting with a "large" ε and decreasing it during the binary search for the right value performed by **`searchInterval`**. This is a standard idea and it was also used by Aronov and Har-Peled [5].

Indeed, assume that in the beginning of the ith step, we know that the required Fréchet distance lies in an interval $\mathcal{I}_{i-1} = [\alpha_{i-1}, \beta_{i-1}]$ and $\beta_{i-1} - \alpha_{i-1} = \|\mathcal{I}_0\|\varepsilon_{i-1}$, where $\varepsilon_{i-1} = 1/2^{i-1}$.

Let $\Delta_{i-1} = \|\mathcal{I}_{i-1}\| = \beta_{i-1} - \alpha_{i-1}$, and let $x_{i,j} = \alpha_{i-1} + j\Delta_{i-1}/4$, for $j = 0,1,2,3,4$. Call the procedure **`decider`** on three values $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$, with the approximation parameter being $c_1\varepsilon_i$, for $c_1 > 0$ being a sufficiently small constant. Based on the outcome of these three calls, we can determine in constant time which of the three intervals $\mathcal{J}_{i,1} = [x_{i,0}, x_{i,2}]$, $\mathcal{J}_{i,2} = [x_{i,1}, x_{i,3}]$, or $\mathcal{J}_{i,3} = [x_{i,2}, x_{i,4}]$ must contain the Fréchet distance. We set this interval to be $\mathcal{I}_i$.

We repeat this process for M steps, where $M = \lceil \lg 1/\varepsilon \rceil$. It is easy to verify that the final interval now provides the required approximation. The running time of this algorithm is $O(\mathsf{N}(1,\pi,\sigma)\log n + \sum_{i=1}^{M}\mathsf{N}(\varepsilon_i,\pi,\sigma))$. Now, by assumption (P2) (see Definition 3.3), we have

$$O\left(\sum_{i=1}^{M}\mathsf{N}(\varepsilon_i,\pi,\sigma)\right) = O\left(\sum_{i=1}^{M}\frac{1}{2^{M-i}}\mathsf{N}\left(\frac{\varepsilon_i}{2^{M-i}},\pi,\sigma\right)\right) = O\left(\mathsf{N}(\varepsilon,\pi,\sigma)\sum_{i=1}^{M}\frac{1}{2^{M-i}}\right)$$

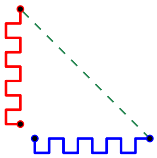
$$= O\big(\mathsf{N}(\varepsilon,\pi,\sigma)\big),$$

and this implies the claim. □

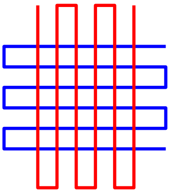
The Result Putting the above together, we get the following result.

Theorem 3.16 *Given two polygonal curves π and σ with a total of n vertices in $\mathbb{R}^d$, and a parameter $1 > \varepsilon > 0$, one can $(1+\varepsilon)$-approximate the Fréchet distance between π and σ in $O(\mathsf{N}(\varepsilon,\pi,\sigma)+\mathsf{N}(1,\pi,\sigma)\log n)$ time* (*see Definition* 3.3).

Interestingly, simplification is critical for the efficiency of the above algorithm. Indeed, consider the two nicely behaved curves depicted below. The reachable portion of the free space diagram of these two curves, for the distance realizing the Fréchet distance, covers a quadratic number of cells.



The use of simplification by itself is not sufficient to guarantee that the presented algorithm is efficient. Indeed, in might not be possible to simplify the input curves at all without losing too much information. In such contrived worst case examples, the free space diagram still has quadratic complexity due to the inherent structure of the curves. See the figure below for one such example. In the next section we will analyze the relative free space complexity using realistic input models and prove the efficiency of the above algorithm, given that the input is "realistic".



4 The Relative Free Space Complexity of Families of Curves

In this section we are going to bound the relative free space complexity for different realistic input models of curves. We will introduce the new class of *c-packed* curves, and we compare this new input model to the previous models of *κ-boundedness* and *low density*.

4.1 On c-Packed Curves

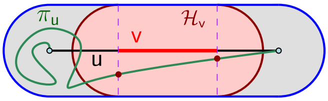
We introduce a new family of curves, c-packed curves, and prove that their relative free space complexity $\mathsf{N}(\varepsilon, \pi, \sigma)$ is linear, for any two curves π and σ in this family. This implies that Theorem 3.16 works in near linear time for c-packed curves, which is one of our main results.

4.1.1 Definition and Basic Properties

Definition 4.1 A curve π in $\mathbb{R}^d$ is *c-packed* if for any point p in $\mathbb{R}^d$ and any radius $r > 0$, the total length of π inside the ball $B(p, r)$ is at most cr.

Lemma 4.2 *Let π be a curve in $\mathbb{R}^d$, $\mu > 0$ be a parameter, and let $\pi' = \mathrm{simpl}(\pi, \mu)$ be the simplified curve. Then $\|\pi \cap B(p, \mathsf{r} + \mu)\| \geq \|\pi' \cap B(p, \mathsf{r})\|$ for any ball $B(p, \mathsf{r})$.*

Proof Let u be a segment of π' that intersects $B(p, \mathsf{r})$ and let $\mathsf{v} = \mathsf{u} \cap B(p, \mathsf{r})$ be this intersection. Let π_{u} be the portion of π that got simplified into u. Observe that π_{u} is a polygonal curve that lies inside a hippodrome of radius μ around u; that is, $\pi_{\mathsf{u}} \subseteq \mathcal{H}_{\mathsf{u}} = \mathsf{u} \oplus B(0, \mu)$, where $\oplus$ denotes the Minkowski sum of the two sets, see the figure below.



In particular, erect two hyperplanes passing through the endpoints of v that are orthogonal to v, and observe that π_{u} must intersect both hyperplanes. Hence, we conclude that the portions of π_{u} in the hippodrome $\mathcal{H}_{\mathsf{v}} = \mathsf{v} \oplus B(0, \mu)$ are of length at least $\|\mathsf{v}\|$. Clearly, $\mathsf{v} \subseteq B(p, \mathsf{r})$ implies that $\mathcal{H}_{\mathsf{v}} \subseteq B(p, \mathsf{r} + \mu)$, which in turn implies that $\pi_{\mathsf{u}} \cap \mathcal{H}_{\mathsf{v}} \subseteq B(p, \mathsf{r} + \mu)$ and thus $\|\pi_{\mathsf{u}} \cap B(p, \mathsf{r} + \mu)\| \geq \|\mathsf{v}\|$.

Summing over all segments v in $\pi' \cap B(p, \mathsf{r})$ implies the claim. □

Lemma 4.3 *Let π be a c-packed curve in $\mathbb{R}^d$, $\mu > 0$ be a parameter, and let $\pi' = \mathrm{simpl}(\pi, \mu)$ be the simplified curve. Then, π' is a $6c$-packed curve.*

Proof Assume, for the sake of contradiction, that $\|\pi' \cap B(p, \mathsf{r})\| > 6c\mathsf{r}$ for some $B(p, \mathsf{r})$ in $\mathbb{R}^d$. If $\mathsf{r} \geq \mu$, then set $\mathsf{r}' = 2\mathsf{r}$ and Lemma 4.2 implies that $\|\pi \cap B(p, \mathsf{r}')\| \geq \|\pi \cap B(p, \mathsf{r} + \mu)\| \geq \|\pi' \cap B(p, \mathsf{r})\| > 6c\mathsf{r} = 3c\mathsf{r}'$, which contradicts that π is c-packed.

If $\mathsf{r} < \mu$ then let U denote the segments of π' intersecting $B(p, \mathsf{r})$ and let $k = |U|$. Observe that $k > 6c\mathsf{r}/2\mathsf{r} = 3c$, as any segment can contribute at most $2\mathsf{r}$ to the length of π' inside $B(p, \mathsf{r})$. Therefore we have $\|\pi' \cap B(p, 2\mu)\| \geq \|\pi' \cap B(p, \mathsf{r} + \mu)\| \geq \|U \cap B(p, \mathsf{r} + \mu)\| \geq k\mu$, since every segment of the simplified curve π' has a minimal length of μ. By Lemma 4.2, this implies that $\|\pi \cap B(p, 3\mu)\| \geq \|\pi' \cap B(p, 2\mu)\| \geq k\mu > 3c\mu$, which is a contradiction to the c-packedness of π. □

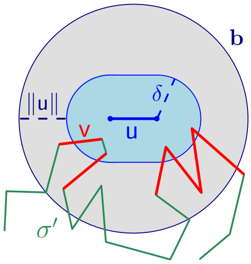
4.1.2 Bounding the Relative Free Space Complexity

Lemma 4.4 *For any two c-packed curves π and σ in $\mathbb{R}^d$, and $0 < \varepsilon < 1$, we have $\mathsf{N}(\varepsilon, \pi, \sigma) = O(cn/\varepsilon)$.*

Proof Let $\delta \geq 0$ be an arbitrary number, $\mu = \varepsilon\delta$, $\pi' = \mathrm{simpl}(\pi, \mu)$ and $\sigma' = \mathrm{simpl}(\sigma, \mu)$

We need to show that the complexity of $D_{\leq\delta}(\pi', \sigma')$ is $O(cn/\varepsilon)$. A free space cell of $D_{\leq\delta}(\pi', \sigma')$ corresponds to two segments $\mathsf{u} \in \pi'$ and $\mathsf{v} \in \sigma'$. The free space in this cell is non-empty if and only if there are two points $p \in \mathsf{u}$ and $q \in \mathsf{v}$ such that

$\|p-q\| \le \delta$. We charge this pair of points to the shorter of the two segments. We claim that a segment cannot be charged too many times.



Indeed, consider a segment $\mathsf{u} \in \pi'$, and consider the ball B of radius $r = (3/2)\|\mathsf{u}\| + \delta$ centered at the midpoint of u, see the figure above. Every segment $\mathsf{v} \in \sigma'$ that participates in a close pair as above and charges u for it, is of length at least $\|\mathsf{u}\|$, and the length of $\mathsf{v} \cap B$ is at least $\|\mathsf{u}\|$. Since σ' is $6c$-packed, by Lemma 4.3, we see that the number of such charges is at most

$$c' = \frac{\|\sigma' \cap B\|}{\|\mathsf{u}\|} \le \frac{6cr}{\|\mathsf{u}\|} = \frac{6c((3/2)\|\mathsf{u}\| + \delta)}{\|\mathsf{u}\|} \le 9c + \frac{6c\delta}{\mu} = O\left(\frac{c}{\varepsilon}\right),$$

since $\|\mathsf{u}\| \ge \mu$.

We conclude that there are at most $c'n$ free space cells that contain a point of $D_{\le\delta}$. The complexity of the free space inside a cell is a constant, thus implying the claim. □

By plugging the above into Theorem 3.16, we get the following result.

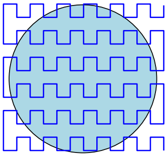
Theorem 4.5 *Given two polygonal c-packed curves π and σ with a total of n vertices in $\mathbb{R}^d$, and a parameter $1 > \varepsilon > 0$, one can $(1+\varepsilon)$-approximate the Fréchet distance between π and σ in $O(cn/\varepsilon + cn\log n)$ time.*

4.2 Relative Free Space Complexity of Low-Density Curves

Definition 4.6 A polygonal curve π in $\mathbb{R}^d$ is *ϕ-low density* if any ball $B(p,r)$ intersects at most ϕ segments of π that are longer than r.

First, observe that this input model is less restrictive than the input model which describes c-packed curves. It can be easily seen by a simple packing argument that a polygonal c-packed curve is ϕ-low density, for $\phi = 2c$. For any ball $B = B(p,r)$, consider the ball with the same center that has radius $r' = 2r$. Any edge intersecting B that is longer than r must contribute at least r to the length of the intersection of the curve with the larger ball, which is bounded by cr'. There can be at most $cr'/r = 2c$ edges of this type.

A curve that is low density, however, is not necessarily c-packed for a small value of c. Indeed, a low-density curve π might have an arbitrarily long intersection with a ball by having sufficiently small segments, see the figure below. However, in this case π must have many vertices in the areas where its length cannot be bounded, as we will show in the following section.



Claim 4.7 *Let π be a ϕ-low density polygonal curve, and let C be a hypercube in $\mathbb{R}^d$ with side length ℓ. Then, the number of edges of length $\geq \ell$ of π that intersect C is bounded by $c_d\phi$, where $c_d = \lceil\sqrt{d}/2\rceil^d$.*

Proof Partition the cube C into a $D \times D \times \cdots \times D$ grid, for $D = \lceil\sqrt{d}/2\rceil$. Clearly, any edge that intersects C that has length $\geq \ell$ must intersect one of the hypercubes in this grid. A hypercube of this grid has diameter

$$\frac{\sqrt{d}\ell}{D} \leq \frac{\sqrt{d}\ell}{\sqrt{d}/2} \leq 2\ell,$$

and is included in a ball of radius ℓ. Thus, a hypercube in this grid intersects at most ϕ such long edges. We conclude that there can be at most ϕD^d long edges intersecting C. □

4.2.1 Low-Density Curves Can Be Long Only if They Pay for It

Lemma 4.8 below testifies that the parts of a low-density curve, where its length cannot be bounded by a constant, can be covered with hypercubes, such that each cube intersects at most a constant number of edges and at most a constant number of other cubes. We use this construction in Lemma 4.9 to relate the length of a low-density curve to the diameter of the covered area to the number of vertices. One can verify Lemma 4.8 using an easy modification of a lemma from [13]. We provide a proof, for the sake of completeness, in Appendix B.

Lemma 4.8 *Let π be a ϕ-low density curve, of which n edges are intersecting a given hypercube C of $\mathbb{R}^d$. The hypercube C can be covered by a set of hypercubes $\mathcal{K}$, such that* (i) $\bigcup\mathcal{K} = C$, (ii) $|\mathcal{K}| \leq 2^{2d+1}n$, (iii) *any point $p \in C$ is covered by at most 2^d hypercubes, and* (iv) *each hypercube of $\mathcal{K}$ intersects at most $c_d\phi$ edges of π, where c_d is a constant that depends only on the dimension d.*

Lemma 4.9 *Let π be a ϕ-low density curve in $\mathbb{R}^d$, and let C be a cube in $\mathbb{R}^d$ with side length r. Let $\alpha = \|\pi \cap C\|$. There must be at least $\Omega((\alpha/r)^{1+1/(d-1)})$ vertices of π contained in $3C$, where $3C$ is the scaling of C by a factor of 3 around its center.*

Proof We will first give a lower bound on the number n of edges intersecting C (i.e., the edges that contribute to α). Then we will account for the edges that have endpoints outside $3C$. So, take the n edges of π that intersect C and construct the cover of C resulting from Lemma 4.8 with respect to these edges.

Let $C_1, \ldots, C_N$ denote the cubes in this cover, where $r_1 \le r_2 \le \cdots \le r_N$ are the side lengths of the cubes used by the cover, respectively. Lemma 4.8 implies that $N \le 2^{d+1} dn$, and, therefore, a lower bound on N would provide a lower bound on n.

So, the sum of the diameters of those N cubes bounds the length of the intersection $\alpha \le \sum_{i=1}^{N} c_d \phi \sqrt{d} r_i$, since every cube in this cover can intersect at most $c_d \phi$ edges of π. Setting $p = d$ and $q = d/(d-1)$, we observe that $1/p + 1/q = 1/d + (d-1)/d = 1$, and by Hölder's inequality,[1] we have

$$\sum_{i=1}^{N} r_i = \sum_{i=1}^{N} r_i \cdot 1 \le \left(\sum_{i=1}^{N} r_i^d\right)^{1/d} \left(\sum_{i=1}^{N} 1^q\right)^{1/q} = \left(\sum_{i=1}^{N} r_i^d\right)^{1/d} N^{(d-1)/d}.$$

Lemma 4.8 also implies that the sum of the volumes of the cubes is at most $2^d \mathrm{vol}(C)$, since every point in C is covered at most 2^d times by this cover. Therefore we have $\sum_{i=1}^{N} r_i^d = \sum_{i=1}^{N} \mathrm{vol}(C_i) \le 2^d \mathrm{vol}(C) = 2^d r^d$. Hence

$$\alpha \le \sum_{i=1}^{N} c_d \phi \sqrt{d} r_i \le c_d \phi \sqrt{d} \left(\sum_{i=1}^{N} r_i^d\right)^{1/d} N^{(d-1)/d} \le c_d \phi \sqrt{d} \left(2^d r^d\right)^{1/d} N^{(d-1)/d}.$$

This implies that $c_2 (\alpha/r)^{d/(d-1)} \le N$, where $c_2 = (2 c_d \phi \sqrt{d})^{-d/(d-1)}$. Since $N \le 2^{2d+1} n$, we have $c_3 (\alpha/r)^{d/(d-1)} \le n$, where $c_3 = \frac{1}{2^{2d+1}} (2 c_d \phi \sqrt{d})^{-d/(d-1)}$.

Now, some of these n edges intersecting C can have both endpoints outside $3C$. Such edges are longer than the side length of C and by Claim 4.7 their number is bounded by $c_d \phi$.

Hence, the number of vertices of π inside $3C$ is at least $n - c_d \phi \ge c_3 (\alpha/r)^{d/(d-1)} - c_d \phi$. □

Remark 4.10 One can also prove Lemma 4.9 directly, by building a quadtree and arguing that for a low-density curve to be sufficiently long, many edges in it have to be (sufficiently) short, thus implying the same bound. However, the current proof is more intuitive and cleaner.

Observation 4.11 The bound in Lemma 4.9 is tight. For any $m > 0$ and any $d > 0$, consider the integer grid in $\mathbb{R}^d$ with coordinates in the range $1, \ldots, m$, and compute a path that visits all these grid points using only the grid edges of unit length, which is clearly possible.

Now, the resulting curve is 2^d-low density and has length $\alpha = m^d - 1$ and its diameter is $r = \sqrt{d} m$. Lemma 4.9 implies that it has $\Omega((\alpha/r)^{d/(d-1)}) = \Omega(m^d)$ vertices. Since this grid has m^d vertices, this is tight.

[1] Hölder's inequality states that $\sum_{i=1}^{n} |a_i b_i| \le (\sum_{i=1}^{n} |a_i|^q)^{1/q} (\sum_{i=1}^{n} |b_i|^p)^{1/p}$ if $1/p + 1/q = 1$.

4.2.2 Accounting for Many Reachable Free Space Cells

If many columns of the free space diagram of the two simplified low-density curves contain a linear number of reachable cells, then the curve must be "long" in the vicinity of the edges corresponding to those columns, since the simplification ensures a minimal edge length. A similar argument holds for the rows. Therefore, using Lemma 4.9, we can charge the additional reachable cells to vertices of the original curves. This yields the following result.

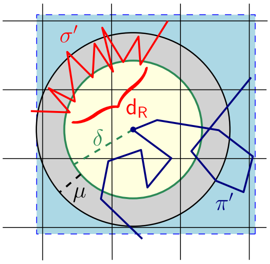
Lemma 4.12 *For any two low-density curves π and σ in $\mathbb{R}^d$, and $0 < \varepsilon < 1$, we have* $\mathsf{N}(\varepsilon, \pi, \sigma) = O(\frac{n^{2(d-1)/d}}{\varepsilon^2})$.

Proof Let $\delta \geq 0$ be an arbitrary radius, and let $\pi' = \mathrm{simpl}(\pi, \mu)$ and $\sigma' = \mathrm{simpl}(\sigma, \mu)$ be their simplifications, where $\mu = \varepsilon\delta$. Then, we need to prove that $N_{\leq\delta}(\pi', \sigma') = O(\frac{n^{2(d-1)/d}}{\varepsilon^2})$.

To this end, it suffices to bound the number of vertex–edge pairs (p, u), where p is a vertex of π', u is an edge of σ', and the distance between p and u is at most δ (naturally, we need to apply the same argument to pairs with vertices in σ' and edges in π'). The total number of such pairs bounds the total complexity of $\mathcal{R}_{\leq\delta} = \mathcal{R}_{\leq\delta}(\pi', \sigma')$.

Set $M = O(n^{1-2/d}/\varepsilon^2)$, and associate every vertex–edge pair (p, u) that appears in the free space diagram $\mathcal{R}_{\leq\delta}$ with the vertex p.

Consider the grid $\mathcal{G}$ of side length δ. For a grid cell R, consider the vertex of π' in R that is associated with the largest number of such vertex–edge pairs, and say it is being associated with $\mathsf{d_R}$ such vertex–edge pairs, and let v_{R} denote this "popular" vertex of π'. The total number of vertex–edge pairs associated with vertices of π' inside R is bounded by $U_{\mathsf{R}} = |\pi' \sqcap \mathsf{R}|\mathsf{d_R}$, where $|\pi' \sqcap \mathsf{R}|$ denotes the number of vertices of π' that lie inside R.



If $\mathsf{d_R} \leq M$ then $U_{\mathsf{R}} \leq |\pi' \sqcap \mathsf{R}|M$, and we charge M units to each vertex of π inside R.

If $\mathsf{d_R} > M$ then the length of σ' inside $C/3$ is at least $\mathsf{d_R}\mu$, where C is a cube centered at R with side length $O(\delta)$. Indeed, all the charges $\mathsf{d_R}$ rise from different segments of σ' that are in distance at most δ from v_{R}, and each such segment has length at least μ.

By Lemma 4.9, we find that σ must have at least $\Omega((\mathsf{d}_{\mathsf{R}}\mu/\delta)^{d/(d-1)}) = \Omega((\mathsf{d}_{\mathsf{R}}\varepsilon)^{d/(d-1)})$ vertices inside C. There is some constant c such that

$$\begin{aligned}
&c(\varepsilon\,\mathsf{d}_{\mathsf{R}})^{d/(d-1)} \leq |\sigma \sqcap C| \\
&\implies \quad \mathsf{d}_{\mathsf{R}} \leq \frac{1}{\varepsilon}\left(\frac{|\sigma \sqcap C|}{c}\right)^{(d-1)/d} \\
&\implies \quad \mathsf{d}_{\mathsf{R}}^2 \leq \frac{1}{\varepsilon^2}\left(\frac{|\sigma \sqcap C|}{c}\right)^{2-2/d} \\
&\implies \quad \mathsf{d}_{\mathsf{R}}^2 \leq |\sigma \sqcap C| \frac{1}{c\varepsilon^2}\left(\frac{|\sigma \sqcap C|}{c}\right)^{1-2/d} \leq \frac{1}{c\varepsilon^2}\left(\frac{n}{c}\right)^{1-2/d} |\sigma \sqcap C| \leq M|\sigma \sqcap C|,
\end{aligned}$$

by picking M to be sufficiently large. In particular, if $|\pi' \sqcap \mathsf{R}| \leq \mathsf{d}_{\mathsf{R}}$, then $U_{\mathsf{R}} = |\pi' \sqcap \mathsf{R}|\,\mathsf{d}_{\mathsf{R}} \leq \mathsf{d}_{\mathsf{R}}^2 \leq M|\sigma \sqcap C|$. Hence, we charge M units to each vertex of σ inside the cube C.

Otherwise, $|\pi' \sqcap \mathsf{R}| > \mathsf{d}_{\mathsf{R}} > M$. But then, the length of π' inside C is at least $|\pi' \sqcap \mathsf{R}|\mu$, and by Lemma 4.9, we see that π must have at least $\Omega((|\pi' \sqcap \mathsf{R}|\varepsilon)^{d/(d-1)})$ vertices inside C. Arguing as above, this implies that $|\pi' \sqcap \mathsf{R}|^2 \leq M|\pi \sqcap C|$. As such, we have $U_{\mathsf{R}} = |\pi' \sqcap \mathsf{R}|\,\mathsf{d}_{\mathsf{R}} \leq |\pi' \sqcap \mathsf{R}|^2 \leq M|\pi \sqcap C|$. Again, we charge M units to each vertex of π inside the cube C.

Since C intersects a constant number of cells of the grid, no vertex would get charged more than a constant number of times by the above scheme. Thus, every vertex, of either curve, gets charged $O(M)$ units overall, and the total number of vertex–edge pairs present in $\mathcal{R}_{\leq\delta}$ is $O(nM)$, as claimed. □

Observation 4.13 One can extend the construction of Observation 4.11 to show that Lemma 4.12 is close to being tight. Indeed, consider the grid curve of Observation 4.11 in $d-1$ dimensions, for an integer m. We now lift it to d dimensions by considering the $[1,m]^d$ cube and placing two copies of the above curve on two opposite faces of the cube, denoted by f and f'. Let π_1 and π_2 denote these two copies.

Next, delete the even edges from π_1 and the odd edges from π_2. Connect every vertex v_1 of π_1 to its corresponding (copied) vertex v_2 in π_2 by a path made out of the $m-1$ unit edges along the grid line connecting the two vertices. This results in a curve π that is similar to the curve constructed in Observation 4.11, but has the advantage that when simplified for the distance $\mu = m$ it results in a curve with m^{d-1} segments of length $\geq m$ that connects points that lie on f and on f', respectively.

Let σ be a copy of π. For a fixed $\varepsilon > 0$, we can add a single segment to π such that the Fréchet distance between the resulting curves is exactly $\delta = m/\varepsilon$. Now, these two curves have $n = 2m^d + 2$ vertices overall, and furthermore, when we simplify them for the distance $\mu = \varepsilon\delta = m$, we end up with two curves such that every long edge of π' is going to be in distance $\leq \delta = m/\varepsilon$ from a constant fraction of the edges of σ' (this would be all the edges if $1/\varepsilon > \sqrt{d}$). Therefore the complexity of the reachable free space is $\Omega(\mathsf{n}_{\pi'}\mathsf{n}_{\sigma'}) = \Omega((m^{d-1})^2) = \Omega(n^{2(d-1)/d})$, where $\mathsf{n}_{\pi'}$ denotes the number of vertices of π'. The upper bound of Lemma 4.12 is (only) larger by a factor of $O(1/\varepsilon^2)$.

By plugging the above into Theorem 3.16, we get the following result.

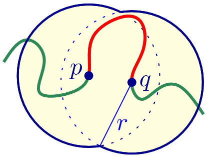
Theorem 4.14 *Given two low-density curves π and σ with a total of n vertices in $\mathbb{R}^d$, and a parameter $\varepsilon > 0$, there exists an algorithm which $(1+\varepsilon)$-approximates the Fréchet distance between π and σ in $O(\frac{n^{2(d-1)/d}}{\varepsilon^2} + n^{2(d-1)/d}\log n)$ time.*

4.3 Relative Free Space Complexity of κ-Bounded Curves

We revisit the definitions of Alt et al. [4] of κ-bounded and κ-straight curves. Note that these definitions describe an extremely restricted class of curves while c-packed curves form a fairly general and natural class of curves. However, it is not true that any κ-bounded curve is $O(\kappa)$-packed. We therefore give a separate proof to bound the relative free space complexity of κ-bounded curves in order to improve upon the result in [4].

Definition 4.15 Let $\kappa \geq 1$ be a given parameter. A curve π is *κ-straight* if for any two points p and q on the curve, we have $\|\pi[p,q]\| \leq \kappa\|p-q\|$.

A curve π is a *κ-bounded* if for all $p,q \in \pi$ we find that the curve $\pi[p,q]$ is contained inside $B(p,r) \cup B(q,r)$, where $r = (\kappa/2)\|p-q\|$, see the figure below.



Lemma 4.16 *A κ-straight curve is 2κ-packed.*

Proof Let π be a κ-straight curve in $\mathbb{R}^d$, and consider any ball $B(p,r)$ that intersects it. Let q and s be the first and last points, respectively, along π that are in $B(p,r)$. Clearly, $\|q-s\| \leq 2r$, and by the κ-straightness $\|\pi \cap B(p,r)\| \leq \|\pi[q,s]\| \leq \kappa\|q-s\| \leq 2\kappa r$. □

Remark 4.17 It is easy to verify that a κ-straight curve is also κ-bounded. However, κ-bounded curves, counterintuitively, can have infinite length even when contained inside a finite domain. An example of this is *Koch's snowflake*, which is a fractal curve depicted in Fig. 4.

To see, intuitively, why Koch's snowflake is κ-bounded, let π_i be the ith polygonal curve generated by this process. There is a natural mapping between any point of π_i and π_{i+1}, for all i. In particular, consider two points p and q on the final curve π^*, and consider the two sequences of points $p_i, q_i \in \pi_i$, where $p_{i+1} \in \pi_{i+1}$ (resp. $q_{i+1} \in \pi_{i+1}$) is the natural image of p_i (resp. q_i), $\lim_{i\to\infty} p_i = p$, and $\lim_{i\to\infty} q_i = q$.

Now, assume that $r = \|p-q\|$. Observe that, for all i, the polygonal curve π_i is made out of segments that are all of the same length. In particular, consider the first index k, such that this segment length of π_k is of length $\leq r/20$. It is easy to argue

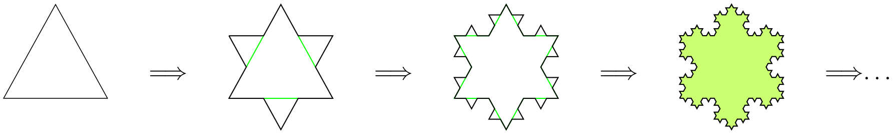


Fig. 4 Koch's snowflake is an example of a κ-bounded curve that has infinite length but a finite diameter

that $\|p_k - p\| \leq r/5$ and $\|q_k - q\| \leq r/5$. In fact, one can argue that no point of π_k moves more than a distance larger than $r/5$ to its final location on π^*.

Now, a tedious argument shows that there are $O(1)$ segments of π_k separating p_k from q_k. Therefore this portion of the curve π_k is covered by a disk of radius $O(r)$, and the corresponding portion of the final curve between p and q is also covered by a disk of radius $O(r)$. This implies that Koch's snowflake is κ-bounded.

A formal proof of this fact is considerably more tedious and is omitted.

Lemma 4.18 *Let π be a κ-bounded polygonal curve in $\mathbb{R}^d$, and let $\mu \leq \delta$ be parameters. Let $\pi' = \operatorname{simpl}(\pi, \mu)$. Then the number of segments of π' intersecting $B(s, \delta)$ is bounded by $O(\kappa^d(1 + \delta/\mu)^d)$, for any $s \in \mathbb{R}^d$.*

Proof For $\pi = \mathsf{u}_1\mathsf{u}_2 \dots \mathsf{u}_k$, let $Y_O = \{\mathsf{u}_1, \mathsf{u}_3, \dots\}$ and $Y_E = \{\mathsf{u}_2, \mathsf{u}_4, \dots\}$ be the sets of odd and even segments of π', respectively.

Let $X_O \subseteq Y_O$ be the set of odd segments of π' intersecting $B(s, \delta)$. For all i, pick an arbitrary point p_i on the ith segment of X_O that lies inside $B(p, \delta)$. Next, pick an original point q_i of π in distance at most μ from p_i, for $i = 1, \dots, M = |X_O|$. Observe that for all i we have $\|s - q_i\| \leq \delta + \mu$. Furthermore, between any two distinct points p_i and p_j on the simplified curve π' there must lie an even segment of Y_E in between them along the curve, and the length of this segment is at least μ (because the simplification algorithm generates segments of length at least μ). Also, the endpoints of this even segment lie on the original curve π.

We claim that no two points of $\mathsf{Q} = \{q_1, \dots, q_M\}$ can be too close to each other; that is, there are no two points $q', q'' \in \mathsf{Q}$, such that $r = \|q' - q''\| \leq \mu/(4\kappa)$. Indeed, assume for the sake of contradiction, that there are two such points. Then, by the above, the portion of π connecting them contains two points t', t'' that are at least μ apart. Observe that $\pi[t', t''] \subseteq X = B(q', (\kappa/2)r) \cup B(q'', (\kappa/2)r)$. However, the maximum distance between two points that are included inside X is bounded by its diameter. We have

$$\mu \leq \|t - t'\| \leq \operatorname{diam}(X) = 2(\kappa/2)r + \|q' - q''\| \leq \frac{\mu}{4} + \frac{\mu}{4\kappa} \leq \frac{\mu}{2},$$

since $\kappa > 1$. A contradiction.

However, all the points of Q lie inside a ball of radius $\delta + \mu$ centered at s. Now, placing a ball of radius $\mu' = \mu/(8\kappa)$ around each point of Q, results in a set of interior disjoint balls. This implies, by a standard packing argument, that the number of points of Q is bounded by $\operatorname{vol}(B(s, \delta + \mu))/\operatorname{vol}(B(\cdot, \mu')) = O((\delta + \mu)^d/(\mu/\kappa)^d) = ((1 + \delta/\mu)^d\kappa^d)$.

This bounds the number of odd segments of π' intersecting the ball $B(s,\delta)$, and a similar argument holds for the even segments intersecting this ball. □

Lemma 4.19 *For any two κ-bounded polygonal curves in $\mathbb{R}^d$ π and σ, $0<\varepsilon<1$, we have $\mathsf{N}(\varepsilon,\pi,\sigma)=O((\kappa/\varepsilon)^d n)$.*

Proof Let $\delta \geq 0$ be an arbitrary radius, and set $\mu=\varepsilon\delta$. Let $\pi'=\operatorname{simpl}(\pi,\mu)$ and $\sigma'=\operatorname{simpl}(\sigma,\mu)$. We need to show that the complexity of the reachable free space $\mathcal{R}_{\leq\delta}(\pi',\sigma')$ is $O(\kappa^d(1+\delta/\mu)^d n)=O((\kappa/\varepsilon)^d n)$.

The boundary of a reachable cell in the free space diagram has a non-empty intersection with $D_{\leq\delta}(\pi',\sigma')$. Otherwise its interior could not be reached by a monotone path from $(0,0)$. Therefore, using an argument similar to the proof of Lemma 4.4, Lemma 4.18 implies the desired bound. □

By plugging the above into Theorem 3.16, we get the following result.

Theorem 4.20 *Given two κ-bounded polygonal curves π and σ with a total of n vertices in $\mathbb{R}^d$, and a parameter $1>\varepsilon>0$, there exists an algorithm which $(1+\varepsilon)$-approximates the Fréchet distance between π and σ in $O((\kappa/\varepsilon)^d n+\kappa^d n\log n)$ time.*

5 Extension to Closed *c*-Packed Curves

The Fréchet distance for closed curves is defined as in Sect. 2 with the altered condition that the reparameterizations f and g are orientation-preserving homeomorphisms on the one-dimensional sphere. Computing the Fréchet distance for closed curves is more difficult, as the constraint that the endpoints of the curves have to be matched to each other is dropped in this case and therefore the set of reparameterizations one has to consider is larger.

Observation 5.1 The decision problem for closed curves can be reduced to the previously considered case of open curves. Given two closed c-packed curves π and σ and a parameter δ. Pick a vertex p of the curve π, and assume that we know a point q on σ that is being matched to p by a pair of reparameterizations of π and σ of width at most δ. Clearly, if we break π open at p, and σ at q, we retrieve two open curves $\widehat{\pi}$ and $\widehat{\sigma}$, and we can use the previous method to decide if $d_{\mathcal{F}}(\widehat{\pi},\widehat{\sigma})\leq\delta$. Hence we only need to generate a suitable set of candidates for q to determine if the Fréchet distance between π and σ is at most δ within a certain approximation error.

Lemma 5.2 *Let π be a closed c-packed polygonal curve in $\mathbb{R}^d$, and let $\mu\leq\delta$ be parameters. Let $\pi'=\operatorname{simpl}(\pi,\mu)$. Then the number of edges of π' intersecting $B(p,\delta)$ is bounded by $O(c\delta/\mu)$, for any $p\in\mathbb{R}^d$.*

Proof Consider the ball $B=B(p,r)$ of radius $r=\mu+\delta$. Any edge u of π' that intersects $B(p,\delta)$ has to contribute at least μ to the length of the intersection with B, as the simplification guarantees that every edge of π' is of length at least μ. Since π' is $6c$-packed, by Lemma 4.3, we have $\|B \cap \pi'\| \le 6cr$, and the number of intersections of π' with $B(p,\delta)$ is $N \le \|B \cap \pi'\|/\mu \le 6cr/\mu = 6c(\mu+\delta)/\mu = O(c + c\delta/\mu)$, which implies the claim. □

Lemma 5.3 *Given two closed c-packed polygonal curves π and σ with a total number of n vertices and parameters δ and $1 > \varepsilon > 0$. Let $\pi' = \mathrm{simpl}(\pi, \mu)$ and $\sigma' = \mathrm{simpl}(\sigma, \mu)$ denote the curves simplified with $\mu \le \varepsilon\delta$ and let p be a vertex of π'. We can compute a set of points $\mathcal{K} \subseteq \sigma'$ of size $O(c/\varepsilon)$, in $O(n + c/\varepsilon)$ time, such that if $d_{\mathcal{F}}(\pi', \sigma') \le \delta$ then there exists a pair of reparameterizations of width at most $(1+\varepsilon)\delta$ that matches p to an element of $\mathcal{K}$.*

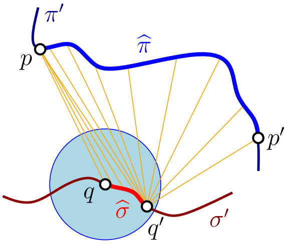
Proof We walk along the curve σ' starting from an arbitrary point. If the starting point is in distance δ from p, then we add it to the candidate set $\mathcal{K}$. As we follow along the curve we create a candidate if we

(a) (re-)enter the ball $B(p,\delta)$, or
(b) have traveled a distance $\varepsilon\delta$ along σ' since the last creation of a candidate, unless we have exited the ball $B(p,\delta)$ in the meantime.

Clearly this takes $O(n + |\mathcal{K}|)$ time.

The number of events of type (a) is bounded (up to a factor of 2) by the number of intersections of σ' with the sphere $S(p,\delta)$, and by Lemma 5.2, this number is bounded by $O(c\delta/\mu) = O(c/\varepsilon)$. By Lemma 4.3 the simplified curve σ' is $6c$-packed and therefore the length of its intersection with $B(p,\delta)$ is at most $6c\delta$. This implies that we can have at most $O(6c\delta/\mu) = O(6c/\varepsilon)$ candidates that were created at events of type (b).

Consider reparameterizations of π' and σ' of width at most δ. Next, consider a point $q \in \sigma'$ that is matched to $p \in \pi'$ by these reparameterizations. Observe that $q \in B(p,\delta)$ and there exists, by construction, a point $q' \in \mathcal{K}$ such that $\|q - q'\| \le \varepsilon\delta$. Let p' be a point on π' that is matched to q' by the given reparameterizations.



We match the curve segment $\widehat{\sigma}$ between q and q' to p and the curve segment $\widehat{\pi}$ between p and p' to q, see the figure above. Clearly this preserves the monotonicity of the matching. By the triangle inequality, any point on $\widehat{\sigma}$ has distance at most $(1+\varepsilon)\delta$ to p. Similarly, for any point on $\widehat{\pi}$ there is a point on $\widehat{\sigma}$ that is in distance δ, therefore q' is in distance $(1+\varepsilon)\delta$ of $\widehat{\pi}$.

We conclude that the Fréchet distance between π' and σ' is at most $(1+\varepsilon)\delta$ when restricted to reparameterizations matching p to q'. □

One can adapt Lemma 3.5 to the closed curves case, by considering the $O(cn/\varepsilon)$ open curves that result from breaking σ' at any point of $\mathcal{K}$. The details of the adaption are straightforward, and we only state the result.

Lemma 5.4 *Given two closed polygonal c-packed curves π and σ with a total of n vertices in $\mathbb{R}^d$, and parameters δ and $1 > \varepsilon > 0$. Then, there exists an algorithm which, in $O((c/\varepsilon)^2 n)$ time, correctly outputs one of the following*:

(A) *If $d_{\mathcal{F}}(\pi,\sigma) \le \delta$ then the algorithm outputs "$\le (1+\varepsilon)\delta$".*
(B) *If $d_{\mathcal{F}}(\pi,\sigma) > (1+\varepsilon)\delta$ then the algorithm outputs "$d_{\mathcal{F}}(\pi,\sigma) > \delta$".*
(C) *If $d_{\mathcal{F}}(\pi,\sigma) \in [\delta,(1+\varepsilon)\delta]$ then the algorithm outputs either of the above outcomes.*

Plugging Lemma 5.4 into the algorithm of Theorem 3.16, we get the following result.

Theorem 5.5 *Given two closed polygonal c-packed curves π and σ with a total of n vertices in $\mathbb{R}^d$, and a parameter $1 > \varepsilon > 0$, one can $(1+\varepsilon)$-approximate the Fréchet distance between π and σ in $O(c^2 n(\varepsilon^{-2} + \log n))$ time.*

6 Conclusions

We presented a new approximation algorithm for the Fréchet distance for polygonal curves in any fixed dimension. The new algorithm is surprisingly simple and should be practical. Furthermore it works for any kind of polygonal curves. Since the algorithm simplifies the curves to the "right" resolution during the execution, we expect the algorithm to be fast in practice. The algorithm's analysis relies on the concept of the relative free space complexity of curves, which tries to capture the complexity of the free space diagram when simplification is being used.

Next, we introduced the c-packed family of curves. While not all curves are c-packed, it seems that most real life curves are c-packed. The family of c-packed curves is closed under simplification, and the property of a curve being c-packed is independent of the ambient dimension of the space containing the curve. We expect this concept to be used to analyze other algorithms in the future.

In particular, the relative free space complexity of c-packed curves is linear. We gave bounds for the relative free space complexity for several other types of curves, from low-density curves to κ-bounded curves. Finally, we also showed that the algorithm can be modified to handle closed curves efficiently.

Lower Bound Our solution to the decision problem "beats" the lower bound of $\Omega(n \log n)$ [9], by a factor of $\log n$ (see Lemma 3.5). Since our decision procedure is approximated this is not too surprising. However, it is enlightening to consider where this proof breaks for our settings. Indeed, Buchin et al. [9] generate two curves such that the Fréchet distance might be realized by one vertex of one curve matching the whole other curve. On the other hand, in our case, the input model coupled with simplification guarantees that the number of segments matching a single vertex is only a constant.

Acknowledgements The authors thank Mark de Berg and Marc van Kreveld for insightful discussions on the problems studied in this paper and related problems. The authors would also like to thank the anonymous referees for their insightful comments.

This research was initiated during a workshop supported by the Netherlands Organization for Scientific Research (NWO) under BRICKS/FOCUS grant number 642.065.503.

The work of A. Driemel has been supported by the Netherlands Organisation for Scientific Research (NWO) under RIMGA (Realistic Input Models for Geographic Applications). The work of S. Har-Peled was partially supported by a NSF AF award CCF-0915984. The work of C. Wenk has been supported by NSF CAREER award CCF-0643597.

Appendix A: Fatness Implies *c*-Packedness

We show that the boundary of an (α, β)-covered shape is c-packed even if the shape does not have a finite descriptive complexity. A somewhat similar result (which however is too weak to prove this result) is the packing lemma of de Berg [14] that shows that the boundary of the union of γ-fat shapes has low density. This implies that a connected component of this boundary has low density.

As mentioned in the introduction, since Koch's snowflake is γ-fat, if the finiteness requirement is removed, it follows that the boundary of γ-fat shapes with unbounded descriptive complexity are not c-packed, for any finite c.

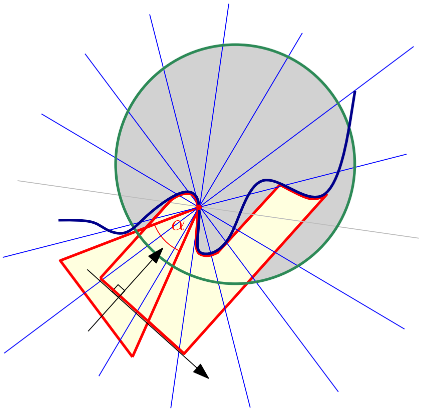
Definition A.1 A bounded simply connected region P in the plane is *(α, β)-covered* if for each point $p \in \partial P$, there exists a triangle T_p, called a *good triangle* of p, such that: (i) p is a vertex of T_p, (ii) $T_p \subseteq P$, (iii) all the angles of T_p are at least α, and (iv) the length of all the edges of T_p is at least $\beta \operatorname{diam}(P)$.

Note that our definition is different from the standard definition of (α, β)-covered shapes, since we do not require that the region P has a finite descriptive complexity.

Lemma A.2 *Let S be a set of segments contained inside a disk with radius r, such that for any point p lying on a segment of S, there is an infinite cone $\mathcal{V}$ of angle at least $\alpha \leq \pi$ with an apex at p, such that the intersection of the interior of $\mathcal{V}$ with S is empty. Then, $\|S\| \leq 10\pi r/(\alpha \sin(\alpha/4))$.*

Proof Let $\mathcal{F}$ be a family of $\lceil 2\pi/(\alpha/2) \rceil$ cones, centered at the origin, such that they cover all directions, and each cone has angle $\alpha/2$. Clearly, for any point p lying on a segment of S, there must be a cone $\mathcal{V} \in \mathcal{F}$, such that the interior of $p + \mathcal{V}$ does not intersect S. We will say that p is *exposed* by $\mathcal{V}$.

So, fix such a cone $\mathcal{V} \in \mathcal{F}$ and consider the direction $\vec{v}$ that splits the angle of $\mathcal{V}$ into two. Rotate the plane such that $\vec{v}$ is the direction of the negative y axis, and observe that any point of S that is exposed by (the rotated) $\mathcal{V}$ lies on the lower envelope of the segments of S. Furthermore, the segment $\mathsf{u} \in S$ that contains this point must have an angle in the range $(-\pi/2+\alpha/4, \pi/2-\alpha/4)$ with the positive direction of the x-axis (we assume u is oriented from left to right).



Now, since the projection of S on the x-axis has length at most $2r$, it follows that the total length of the segments exposed by $\mathcal{V}$ is at most $2r/\sin(\alpha/4)$.

Hence, the total length of segments of S is bounded by

$$|\mathcal{F}|\left(\frac{2r}{\sin(\alpha/4)}\right) = \left(\frac{4\pi}{\alpha}+1\right)\left(\frac{2r}{\sin(\alpha/4)}\right) \leq \frac{10\pi r}{\alpha \sin(\alpha/4)}. \qquad \square$$

Lemma A.3 *If P is an (α, β)-covered polygon in the plane then it is c-packed, for*

$$c = O\left(\frac{1}{\alpha\beta \sin(\alpha/4)\tan(\alpha)}\right).$$

Proof Let $S = \partial P$, and consider any disk D of radius r in the plane. Observe that the height of a good triangle is at least $\rho = (s/2)\tan(\alpha)$, for $s = \beta \mathrm{diam}(P)$, and this also bounds the distance of any vertex of a good triangle to its facing edge. If $r \leq \rho/2$, then any good triangle for a point of S behaves like a cone as far as $S \cap D$ is concerned, and Lemma A.2 implies that $\|S \cap D\| \leq 10\pi r/(\alpha \sin(\alpha/4))$ as desired.

If $r \leq \mathrm{diam}(P)$, then cover D by $m = (2\sqrt{2}r/\rho + 1)^2$ disks of radius $\rho/2$. Clearly, for each such disk, the total length of segments of S inside it, by Lemma A.2, is at most $5\pi\rho/(\alpha\sin(\alpha/4))$. Therefore the total length of S inside D is

$$\frac{5\pi\rho}{(\alpha \sin(\alpha/4))}\left(\frac{2\sqrt{2}r}{\rho}+1\right)^2 \leq \frac{160\pi}{\alpha\sin(\alpha/4)} \cdot \frac{r}{\rho} \cdot r \leq \frac{320\pi}{\alpha\beta\sin(\alpha/4)\tan(\alpha)} r.$$

Observe that the total length of ∂P is bounded by the above bound, by taking D to be a disk of radius $r = \mathrm{diam}(P)$ centered at some point of P. Therefore the claim trivially holds in the case $r \geq \mathrm{diam}(P)$. □

Appendix B: A Proof of Lemma 4.8

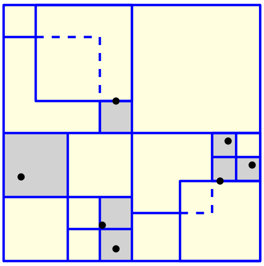
Lemma B.1 *Let* P *be a set of* n *points in* $\mathbb{R}^d$, *contained inside a hypercube* C. *Then one can cover* C *by a set of cubes* $\mathcal{K}$, *such that the following properties hold.*

(A) $\bigcup \mathcal{K} = C$.
(B) $|\mathcal{K}| \le 2^{d+1} dn$.
(C) *Each* $p \in C$ *is covered by at most* 2^d *cubes.*
(D) *Each cube contains at most one point from* P.

Proof We can use the following algorithm to construct a d-dimensional reduced quadtree, of which the set of cubes corresponding to the leaf nodes satisfies the requirements for $\mathcal{K}$.

Take C as the root node. Split the current node recursively into 2^d subcubes, until there is only one point left in the current node, while abiding to the following rule.

In each step, either (A) do a proper quadtree split if at least two of the immediately resulting subcubes contain a point of P, or (B) perform a *reduced split* otherwise, such that all points are contained in exactly one minimal subcube. A reduced split is formed by allowing the cubes the cube to overlap, by shrinking one of the 2^d subcubes containing the points, and enlarging all the others. Such a reduced split is depicted in the figure below. However, we can ensure that only those subcubes will overlap that do not contain any point of P and will therefore not be split further. Clearly each point in the covered area is covered by at most 2^d leaf nodes.



A split of type (A) separates the set of points into at least two non-empty subsets. A split of type (B) results in a point on the splitting plane. Both events can happen at most dn times and produce each at most 2^d extra nodes. Therefore the size of $\mathcal{K}$ is bounded by $2^{d+1}dn$. □

Proof of Lemma 4.8 This follows directly from Lemma B.1. Indeed, for every edge of π add the corners of the axis parallel cube containing it to a set of points P. Next, consider the respective quadtree construction of Lemma B.1 for $\mathsf{P} \subseteq C$. The cover uses at most $2^{d+1}m$ boxes, where $m \le 2^d n = |\mathsf{P}|$.

Consider a cube C' in the resulting decomposition of C, and an edge u of π that intersects it. If the length of u is shorter than the side length of C', then one of the corners of the bounding cube of u must be in C', and C' cannot be a leaf of the quadtree. This implies that C' can be intersected only by edges that are at least as long as its side length.

By the low-density property of π and by Claim 4.7, C' can intersect at most $c_d\phi$ edges of π, which implies the lemma. □

References

1. Agarwal, P.K., Har-Peled, S., Mustafa, N., Wang, Y.: Near-linear time approximation algorithms for curve simplification in two and three dimensions. Algorithmica **42**, 203–219 (2005)
2. Alt, H.: The computational geometry of comparing shapes. In: Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday, pp. 235–248. Springer, Berlin (2009)
3. Alt, H., Godau, M.: Computing the Fréchet distance between two polygonal curves. Int. J. Comput. Geom. Appl. **5**, 75–91 (1995)
4. Alt, H., Knauer, C., Wenk, C.: Comparison of distance measures for planar curves. Algorithmica **38**(1), 45–58 (2004)
5. Aronov, B., Har-Peled, S.: On approximating the depth and related problems. SIAM J. Comput. **38**(3), 899–921 (2008)
6. Aronov, B., Har-Peled, S., Knauer, C., Wang, Y., Wenk, C.: Fréchet distance for curves, revisited. In: Proc. 14th Annu. European Symp. Algorithms, pp. 52–63 (2006)
7. Bose, P., Cheong, O., Dujmović, V.: A note on the perimeter of fat objects. Comput. Geom. Theory Appl. **44**(1), 1–8 (2011)
8. Brakatsoulas, S., Pfoser, D., Salas, R., Wenk, C.: On map-matching vehicle tracking data. In: Proc. 31st VLDB Conference, pp. 853–864 (2005)
9. Buchin, K., Buchin, M., Knauer, C., Rote, G., Wenk, C.: How difficult is it to walk the dog. In: Proc. 23rd Euro. Workshop on Comput. Geom., pp. 170–173 (2007)
10. Buchin, K., Buchin, M., Gudmundsson, J.: Detecting single file movement. In: Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. GIS, pp. 288–297 (2008)
11. Buchin, K., Buchin, M., Gudmundsson, J., Maarten, L., Luo, J.: Detecting commuting patterns by clustering subtrajectories. In: Proc. 19th Annu. Internat. Symp. Algorithms Comput., pp. 644–655 (2008)
12. Callahan, P.B., Kosaraju, S.R.: A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. J. Assoc. Comput. Mach. **42**, 67–90 (1995)
13. de Berg, M.: Linear size binary space partitions for uncluttered scenes. Algorithmica **28**, 353–366 (2000)
14. de Berg, M.: Improved bounds on the union complexity of fat objects. Discrete Comput. Geom. **40**(1), 127–140 (2008)
15. de Berg, M., Katz, M.J., van der Stappen, A.F., Vleugels, J.: Realistic input models for geometric algorithms. Algorithmica **34**, 81–97 (2002)
16. Driemel, A., Har-Peled, S., Wenk, C.: Approximating the Fréchet distance for realistic curves in near linear time. In: Proc. 26th Annu. ACM Symp. Comput. Geom., pp. 365–374 (2010). arXiv:1003.0460
17. Efrat, A.: The complexity of the union of (α, β)-covered objects. SIAM J. Comput. **34**(4), 775–787 (2005)
18. Erickson, J.: On the relative complexities of some geometric problems. In: Proc. 7th Canad. Conf. Comput. Geom., pp. 85–90 (1995)
19. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping to massive dataset. In: Proc. of the Third Euro. Conf. Princip. Data Mining and Know. Disc., pp. 1–11 (1999)
20. Kim, M.S., Kim, S.W., Shin, M.: Optimization of subsequence matching under time warping in time-series databases. In: Proc. ACM Symp. Appl. Comput., pp. 581–586 (2005)
21. Kwong, S., He, Q.H., Man, K.F., Tang, K.S., Chau, C.W.: Parallel genetic-based hybrid pattern matching algorithm for isolated word recognition. Int. J. Pattern Recognit. Artif. Intell. **12**(5), 573–594 (1998)
22. Munich, M.E., Perona, P.: Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In: Proc. 7th Int. Conf. Comp. Vision, pp. 108–115 (1999)
23. Sriraghavendra, E., Karthik, K., Bhattacharyya, C.: Fréchet distance based approach for searching online handwritten documents. In: Proc. 9th Int. Conf. Doc. Anal. Recogn., pp. 461–465 (2007)
24. Wenk, C., Salas, R., Pfoser, D.: Addressing the need for map-matching speed: localizing global curve-matching algorithms. In: Proc. 18th Int. Conf. Sci. Statist. Database Managm., pp. 879–888 (2006)