Computing the Fréchet Gap Distance*

Chenglin Fan[1] and Benjamin Raichel[1]

1 **Department of Computer Science, University of Texas at Dallas**
Richardson, TX 75080, USA
cxf160130@utdallas.edu, benjamin.raichel@utdallas.edu.

Abstract

Measuring the similarity of two polygonal curves is a fundamental computational task. Among alternatives, the Fréchet distance is one of the most well studied similarity measures. Informally, the Fréchet distance is described as the minimum leash length required for a man on one of the curves to walk a dog on the other curve continuously from the starting to the ending points. In this paper we study a variant called the Fréchet gap distance. In the man and dog analogy, the Fréchet gap distance minimizes the difference of the longest and smallest leash lengths used over the entire walk. This measure in some ways better captures our intuitive notions of curve similarity, for example giving distance zero to translated copies of the same curve.

The Fréchet gap distance was originally introduced by Filtser and Katz [18] in the context of the discrete Fréchet distance. Here we study the continuous version, which presents a number of additional challenges not present in discrete case. In particular, the continuous nature makes bounding and searching over the critical events a rather difficult task.

For this problem we give an $O(n^5 \log n)$ time exact algorithm and a more efficient $O(n^2 \log n + \frac{n^2}{\varepsilon} \log \frac{1}{\varepsilon})$ time $(1+\varepsilon)$-approximation algorithm, where n is the total number of vertices of the input curves. Note that for (small enough) constant ε and ignoring logarithmic factors, our approximation has quadratic running time, matching the lower bound, assuming SETH [10], for approximating the standard Fréchet distance for general curves.

1998 ACM Subject Classification F.2.2, I.1.2, I.3.5

Keywords and phrases Fréchet Distance, Approximation, Polygonal Curves.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Polygonal curves arise naturally in the modeling of a number computational problems, and for such problems assessing the similarity of two curves is one of the most fundamental tasks. There are several competing measures for defining curve similarity. Among these, there has been strong interest in the Fréchet distance, particularly from the computational geometry community, as the Fréchet distance takes into account the continuous "shape" of the curves rather than just the set of points in space they occupy. The Fréchet distance and related measures have been used for a variety of applications [20, 9, 23, 22, 11], and it is typically illustrated as follows. Let the two polygonal curves be denoted π and σ, with n vertices in total. Imagine a man and a dog are respectively placed at the starting vertices of π and σ, and they must each move continuously along their curves to their respective ending points. The man and dog are connected by a leash, and the Fréchet distance is the minimum leash length required over all possible walks of the man and dog, where the man and dog can independently control their speed but cannot backtrack.

* Work on this paper was partially supported by NSF CRII Award 1566137.

© Chenglin Fan and Benjamin Raichel;
licensed under Creative Commons License CC-BY
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper we consider a variant called the *Fréchet gap distance*, originally introduced by Filtser and Katz in the context of the discrete Fréchet distance [18]. In the man and dog analogy, this variant minimizes the difference of the lengths of the longest and shortest leashes used over the entire walk. As discussed in [18], since this measure considers both the closest and farthest relative positions of the man and dog, in many cases it is closer to our intuitive notion of curve similarity. Notably, two translated copies of the *same* curve have Fréchet gap distance zero, as opposed to the magnitude of the translation under the standard Fréchet distance. Though this is not to say that it is the same as minimizing the standard Fréchet distance under translation. For instance, fix any two points on a rigid body in two or three dimensions. The pair of curves traced out by these points as we arbitrarily rotate and translate the rigid body will always have Fréchet gap distance zero (see Figure 1.1).

A natural scenario for the gap distance is planning the movement of military units, where one wants them to be sufficiently close to support each other in case of need, but sufficiently far from each other to avoid unintended interaction (i.e., friendly fire). Such units might move on two major roads that are roughly parallel to each other, thus matching our setup.

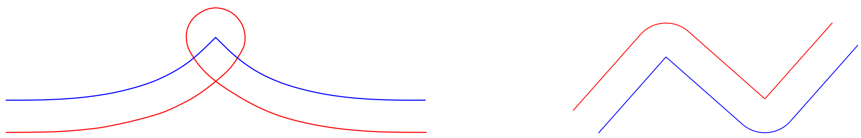


Figure 1.1 Left: A 2D "airplane roll". Right: Turning in 2D by pivoting on one side at a time.

Previous Work. Alt and Godau [4] presented an $O(n^2 \log(n))$ time algorithm to compute the standard Fréchet distance. More recently Buchin et al. [12] improved the logarithmic factor in the running time (building on [1]), however Bringmann [10] showed that assuming the Strong Exponential Time Hypothesis (SETH), no strongly subquadratic time algorithm is possible. Moreover, Bringmann showed that assuming SETH there is no strongly subquadratic 1.001-approximation algorithm, thus ruling out the possibility of a strongly subquadratic PTAS for general curves. On the other hand, there are fast approximation algorithms for several families of nicely behaved curves, for example Driemel et al. [16] gave an $O(cn/\varepsilon + cn \log n)$ time algorithm for the case of c-packed curves.

Many variants of the Fréchet distance between polygonal curves have been considered. Alt and Godau [4] gave a quadratic time algorithm for the weak Fréchet distance, where backtracking on the curves is allowed. Driemel and Har-Peled [15] considered allowing shortcuts between vertices, and for this more challenging variant, they give a near linear time 3-approximation for c-packed curves. Later Buchin et al. [14] proved the general version, where shortcutting is also allowed on edge interiors, is NP-hard (and gave an approximation for the general and an exact algorithm for the vertex case). The *discrete* Fréchet distance only considers distances at the vertices of polygonal curves, i.e. rather than a continuously walking man and dog, there is a pair of frogs hopping along the vertices. This somewhat simpler variant can be solved in $O(n^2)$ time using dynamic programming [17]. Interestingly, Agarwal et al. [1] showed the discrete variant can be solved in weakly subquadratic $O(n^2 \log\log n/ \log n)$ time, however the above results of Bringmann [10] also imply there is no strongly subquadratic algorithm for the discrete case, assuming SETH. Avraham et al. [6] considered shortcuts in the discrete case, providing a strongly subquadratic running time, showing shortcuts make it more tractable, which was the reverse for the continuous case.

Minimizing Fréchet distance under translation (and other transformations) was previously considered, though running times are typically large. For example, Alt et al. [5] gave a roughly $O(n^8)$ time algorithm, though they also gave a $O(n^2/\varepsilon^2)$ time $(1+\varepsilon)$-approximation. Avraham et al. [7] consider the discrete case, and provide a nice summary of other previous work. The Fréchet distance has also been extended to more general inputs, such as graphs [3], piecewise smooth curves [21], simple polygons [13], surfaces [2], and complexes [19]. In general there are too many Fréchet distance results to cover, and the above is just a sampling.

The most relevant previous work is that of Filtser and Katz [18], who first proposed the Fréchet gap distance. The technical content of the two papers differs significantly however, as [18] considers the discrete case, avoiding many of the difficulties faced in our continuous setting. In particular, a solution to the gap problem is a distance interval. In the continuous case the challenge is bounding the number of possible intervals, while in the discrete case a bound of $O(n^4)$ holds, as each interval endpoint is a vertex to vertex distance. Using a result of Avraham et al. [7], Filtser and Katz improve this to an $O(n^3)$ time algorithm to compute the minimum discrete Fréchet gap. They also provide $O(n^2 \log^2 n)$ time algorithms for one-sided discrete Fréchet gap with shortcuts and the weak discrete Fréchet gap distance.

Contributions and Overview. Here we consider the continuous Fréchet gap distance problem (defined informally above, and formally below). This is the first paper to consider the more challenging continuous version of this problem. For this problem we provide an $O(n^5 \log n)$ time exact algorithm and a more efficient $O(n^2 \log n + \frac{n^2}{\varepsilon} \log \frac{1}{\varepsilon})$ time $(1+\varepsilon)$-approximation algorithm, and we now outline our approach and main contributions.

The standard approach for computing the Fréchet distance starts by solving the decision version for a given query distance $\delta \geq 0$, by using the free space diagram, which describes the pairs of points (one from each curve) which are within distance δ. The convexity of the free space cells allows one to efficiently propagate reachibility information, leading to a quadratic time proceedure overall. For the Fréchet gap problem the free space cells are no longer convex, but despite this we show that they have sufficient structure to allow efficient reachability propagation, again leading to a quadratic time decider, which in our case determines whether a given query interval $[s, t]$ is feasible.

The next step in computing the Fréchet distance is to find a polynomially sized set of critical events, determined by the input curves, to search over. For the standard Fréchet distance this set has $O(n^3)$ size. For the Fréchet gap case however the number of critical events can be much larger as they are determined by two rather than one distance value. As mentioned above, for the discrete case only pairs of vertex distances are relevant and so there are $O(n^4)$ events. On the other hand, for the continuous case there can now be "floating" monotonicity events where increasing (or decreasing) the gap interval endpoint values simultaneously may lead to an entire continuum of optimum intervals. Despite this we show there is an $O(n^6)$ sized set of canonical intervals containing an optimum solution.

The last step is efficiently searching over the critical events. For the standard Fréchet distance this can be done via parametric search [4] or sampling [19], yielding an $O(n^2 \log n)$ running time. Searching in the gap case however is more challenging, as there is no longer a natural linear ordering of events. Specifically, the set of feasible intervals may not appear contiguously when ordering candidate intervals by width. Despite this, we similarly get a near linear factor speed up, by using a more advanced version of the basic approach in [19].

Our approximation uses the observation that all feasible intervals share a common value. Roughly speaking, at the cost of a 2-approximation, this allows us to consider the radius of intervals centered at this common value, rather than two independent interval endpoints, reducing the number of critical events. This is improved to a $(1+\varepsilon)$-approximation, and finally the running time is reduced by a linear factor, again using a modified version of [19].

2 Preliminaries

Throughout, given points $p, q \in \mathbb{R}^d$, $||p - q||$ denotes their Euclidean distance. Moreover, given two (closed) sets $P, Q \subseteq \mathbb{R}^d$, $dist(P, Q) = \min_{p \in P, q \in Q} ||p - q||$ denotes their distance.

2.1 Fréchet Distance and Fréchet Gap Distance

A *polygonal curve* π of length n is a continuous mapping from $[0, n]$ to $\mathbb{R}^d$, such that for any integer $1 \leq i \leq n$, the restriction of π to the interval $[i-1, i]$ is defined by $\pi((i-1)+\alpha) = (1-\alpha)\pi(i-1)+\alpha\pi(i)$ for any $\alpha \in [0, 1]$, i.e. a straight line segment. When it is clear from the context, we often use π to denote the image $\pi([0, n])$. The set of vertices of π is defined as $V(\pi) = \{\pi_0, \pi_1 \ldots, \pi_n\}$, where $\pi_i = \pi(i)$, and the set of edges is $E(\pi) = \{\pi_0\pi_1, \ldots, \pi_{n-1}\pi_n\}$, where $\pi_{i-1}\pi_i$ is the line segment connecting π_{i-1} and π_i.

A reparameterization for a curve π of length n is a continuous non-decreasing bijection $f : [0, 1] \to [0, n]$ such that $f(0) = 0, f(1) = n$. Given reparameterizations f, g of an n length curve π and an m length curve σ, respectively, the *width* between f and g is defined as

$$width_{f,g}(\pi, \sigma) = \max_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))||$$

The (standard) *Fréchet distance* between π and σ is then defined as

$$d_{\mathcal{F}}(\pi, \sigma) = \min_{f,g} width_{f,g}(\pi, \sigma)$$

where f, g range over all possible reparameterizations of π and σ.

A *gap* is an interval $[s, t]$ where $0 \leq s \leq t$ are real numbers, and the *gap width* is $t - s$. Similarly, given reparameterizations f, g for curves π, σ, define their gap and gap width as

$$gap_{f,g}(\pi, \sigma) = \left[\min_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))||, \max_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))||\right]$$

$$gapwidth_{f,g}(\pi, \sigma) = \max_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))|| - \min_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))||$$

The *Fréchet gap distance* between two curves π and σ is then defined as

$$d_{\mathcal{G}}(\pi, \sigma) = \min_{f,g} gapwidth_{f,g}(\pi, \sigma)$$

where f, g range over all possible reparameterizations of π and σ.

If there exist reparameterizations f and g for curves π and σ satisfying the inequalities,

$$\max_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))|| \leq t \qquad \min_{\alpha \in [0,1]} ||\pi(f(\alpha)) - \sigma(g(\alpha))|| \geq s,$$

we say $[s, t]$ is a *feasible gap* between curves π and σ. Throughout the paper $[s^*, t^*]$ denotes an arbitrary optimal gap, that is $t^* - s^* = d_{\mathcal{G}}(\pi, \sigma)$. (Note there may be more than one such optimal gap, and moreover a feasible gap does not necessarily contain an optimal gap.)

Note that in the later sections of the paper we refer to gaps or intervals $[s, t]$ instead as parametric points or pairs (s, t), in which case feasibility is defined analogously.

2.2 Free Space

To compute the standard Fréchet distance one normally looks at the so called *free space*. The t free space between curves π and σ, with n and m edges respectively, is defined as

$$F_t = \{(\alpha, \beta) \in [0, n] \times [0, m] \mid ||\pi(\alpha) - \sigma(\beta)|| \leq t\}.$$

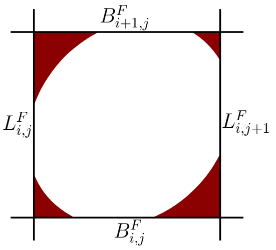


Figure 2.1 Free space cell

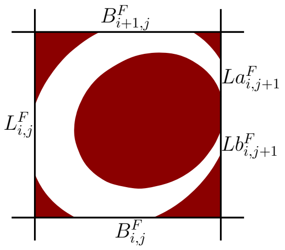


Figure 2.2 Relative free space cell

Similarly define $F_t^< = \{(\alpha, \beta) \in [0, n] \times [0, m] \mid ||\pi(\alpha) - \sigma(\beta)|| < t\}$ to be F_t without its boundary. $C(i, j) = [i-1, i] \times [j-1, j]$ is referred to as the cell of the free space diagram determined by edges $\pi_{i-1}\pi_i$ and $\sigma_{j-1}\sigma_j$, and the free space within this cell is

$$F_t(i, j) = \{(\alpha, \beta) \in [i-1, i] \times [j-1, j] \mid ||\pi(\alpha) - \sigma(\beta)|| \leq t\}.$$

Alt and Godau [4] showed that the free space within a cell is always a convex set (specifically, the clipping of an affine transformation of a disk to the cell). Moreover, any x, y monotone path in the free space from $(0, 0)$ to (n, m) corresponds to a pair of reparameterizations f, g of π, σ such that $width_{f,g}(\pi, \sigma) \leq t$. The converse also holds and hence $d_{\mathcal{F}}(\pi, \sigma) \leq t$ if and only if such a monotone path exists. These two statements together imply that in order to determine if $d_{\mathcal{F}}(\pi, \sigma) \leq t$, it suffices to restrict attention to the free space intervals on the boundaries of the cells. Specifically, let $L^F_{i,j}$ (resp. $B^F_{i,j}$) denote the left (resp. bottom) free space interval of $C(i, j)$, i.e. $L^F_{i,j} = F_t(i, j) \cap (\{i-1\} \times [j-1, j])$ (resp. $B^F_{i,j} = F_t(i, j) \cap ([i-1, i] \times \{j-1\})$). See Figure 2.1.

2.3 Relative Free Space

We extend the standard free space definitions of the previous section to the Fréchet gap distance problem. First we define the s, t relative free space between π and σ to be

$$F_{[s,t]} = \{(\alpha, \beta) \in [0, n] \times [0, m]| \mid s \leq ||\pi(\alpha) - \sigma(\beta)|| \leq t\} = F_t \setminus F_s^<,$$

describing all pairs of points, one on π and one on σ, whose distance is contained in $[s, t]$. For a point (α, β) in a cell of $F_{[s,t]}$ or F_t, throughout we use the colloquial terms higher or lower (resp. right or left) to refer larger or smaller value of α (resp. β).

Again we seek an x, y monotone path in the relative free space from $(0, 0)$ to (n, m), since such a path corresponds to a pair of reparameterizations f, g of π, σ such that $gapwidth_{f,g}(\pi, \sigma) \leq t - s$, and hence $d_{\mathcal{G}}(\pi, \sigma) \leq t - s$. Conversely, if no such path exists then $[s, t]$ is not a feasible gap for π and σ, implying that $[s^*, t^*] \not\subseteq [s, t]$, but note however that unlike the standard Fréchet distance, it may still hold that $t^* - s^* \leq t - s$.

The relative free space in the cell $C(i, j)$ determined by edges $\pi_{i-1}\pi_i$ and $\sigma_{j-1}\sigma_j$ is,

$$F_{[s,t]}(i, j) = \{(\alpha, \beta) \in [i-1, i] \times [j-1, j] \mid s \leq ||\pi(\alpha) - \sigma(\beta)|| \leq t\} = F_t(i, j) \setminus F_s^<(i, j).$$

Another technical challenge with the Fréchet gap problem arises from the fact that relative free space in a cell may not be convex (see Figure 2.2). However, there is some structure. Observe that $F_{[s,t]}(i, j) = F_t(i, j) \setminus F_s^<(i, j)$, and hence is the set difference of two convex sets, where one is contained in the other. In other words, it looks like a standard free space cell with a hole removed. In particular, we can again look at the free space intervals on the cell boundaries. As $F_t(i,j)$ is convex, it still determines a single interval on each cell boundary, however, this interval may be broken into two subintervals by the removal of $F_s(i,j)$ (whose convexity implies it is at most two subintervals). Let $L^F_{i,j} = Lb^F_{i,j} \cup La^F_{i,j}$ denote the relative free space on the left boundary of $C(i,j)$, where $Lb^F_{i,j}$ denotes the bottom and $La^F_{i,j}$ the top interval (note if $F_s(i,j)$ does not intersect the boundary then $Lb^F_{i,j} = La^F_{i,j} = L^F_{i,j}$). Similarly, let $B^F_{i,j} = Bl^F_{i,j} \cup Br^F_{i,j}$ denote the relative free space on the bottom boundary of $C(i,j)$, where $Bl^F_{i,j}$ denotes the left and $Br^F_{i,j}$ the right interval.

3 The Fréchet Gap Decision Problem

The Fréchet gap decision problem is defined as follows.

▶ **Problem 1.** *Given polygonal curves π, σ, is a given interval $[s,t]$ a feasible gap for π, σ?*

As discussed in Section 2.3, $[s,t]$ is a feasible gap for π and σ if and only if there exists an x,y monotone path from $(0,0)$ to (n,m) in the $[s,t]$ relative free space $F_{[s,t]}$. This motivates the definition of the reachable relative free space,

$$RF_{[s,t]} = \{(\alpha,\beta) \in [0,n] \times [0,m] \mid \text{there exists an } x,y \text{ monotone path from}(0,0) \text{ to } (\alpha,\beta)\}.$$

Hence the answer to Problem 1 is 'yes' if and only if $(n,m) \in RF_{[s,t]}$. As was the case with the relative free space, the relevant information for the reachable relative free space is contained on the cell boundaries. We now describe how to propagate the reachable information from the left and bottom boundary to the right and top boundary of a cell, which ultimately will allow us to propagate the reachable information from $(0,0)$ to (n,m). (Note this is the typical approach to solving the standard Fréchet distance decision problem.)

Let $L^R_{i,j}$ and $B^R_{i,j}$ denote the reachable subsets of the left and bottom boundaries of $C(i,j)$. First we argue that like $L^F_{i,j}$, $L^R_{i,j}$ is composed of at most two disjoint intervals. Let $Lx^F_{i,j}$ be either $La^F_{i,j}$ or $Lb^F_{i,j}$. The reachable subset of $Lx^F_{i,j}$ is a single connected interval. To see this, observe that wherever the lowest reachable point in $Lx^F_{i,j}$ lies, all points above it in $Lx^F_{i,j}$ are reachable by a monotone path. As $L^R_{i,j}$ is a subset of $L^F_{i,j}$, this implies it is composed of at most two intervals denoted $La^R_{i,j}$ and $Lb^R_{i,j}$ (if $L^F_{i,j}$ is single interval then $L^R_{i,j} = La^R_{i,j} = Lb^R_{i,j}$). $Bl^R_{i,j}$ and $Br^R_{i,j}$ are defined similarly.

Propagating in a cell: Given $L^R_{i,j}$ and $B^R_{i,j}$, we now describe how to compute $L^R_{i,j+1}$ ($B^R_{i+1,j}$ is handled similarly). There are four cases, determined by whether we are propagating $L^R_{i,j}$ or $B^R_{i,j}$, and whether we are going above or below the hole $F_s(i,j)$. First, some notation.

▶ **Definition 2.** Label the leftmost and rightmost vertical lines tangent to the hole $F_s(i,j)$ as $v\ell^l_{i,j}$ and $v\ell^r_{i,j}$, and label the topmost and bottommost horizontal tangent lines as $h\ell^a_{i,j}$ and $h\ell^b_{i,j}$ (see Figure 3.1). Similarly define the leftmost point $\mathcal{H}^l_{i,j}$, the rightmost point $\mathcal{H}^r_{i,j}$, the topmost point $\mathcal{H}^a_{i,j}$, and the bottommost point $\mathcal{H}^b_{i,j}$, of $F_s(i,j)$ (Note any one of these points may be undefined if $F_s(i,j)$ intersects the boundary in more than a single point, as is the case for $\mathcal{H}^r_{i,j}$ in Figure 3.1.) Finally, let $\mathcal{I}^a_{i,j}$ be the highest and $\mathcal{I}^b_{i,j}$ the lowest point of $L^F_{i,j+1}$. When i,j is fixed, the subscript is often dropped.

The above notation will be used throughout the paper as it defines the relevant extent measures of the relative free space. Here we also define the point w_a to be the intersection point of $La^F_{i,j+1}$ with $h\ell^a$, or more generally if they do no intersect w_a is the lowest point of $La^F_{i,j+1}$ that is above $h\ell^a$. (Note w_a is the lowest reachable point when passing over the hole $F_s(i,j)$, and may possibly be undefined.) For the four cases below we consider four points

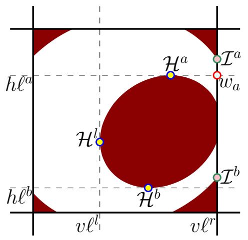


Figure 3.1 Free space cell

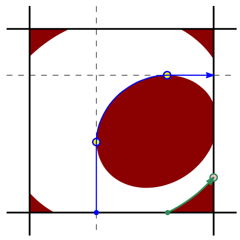


Figure 3.2 $B^R_{i,j}$ to $L^R_{i,j+1}$

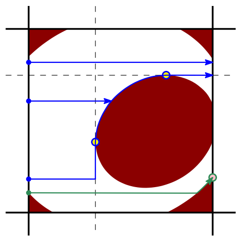


Figure 3.3 $L^R_{i,j}$ to $L^R_{i,j+1}$

p_r, p_l, p_b, and p_a. We assume these points are defined, though they may not be depending on the structure of $L^R_{i,j}$ and $B^R_{i,j}$, in which case there is nothing to propagate.

1) Propagating $B^R_{i,j}$

   a) Below $F_s(i,j)$: Let p_r be the rightmost point of $Br^R_{i,j}$ (note we may have $Br^F_{i,j} = B^F_{i,j}$). In this case there is a monotone path along the boundary of $F_t(i,j)$ from p_r to $\mathcal{I}^b$, and hence all of $Lb^F_{i,j+1}$ is reachable, i.e. $Lb^R_{i,j+1} = Lb^F_{i,j+1}$. See green path in Figure 3.2.

   b) Above $F_s(i,j)$: Let p_l be the intersection point of $Bl^R_{i,j}$ and the line $v\ell^l$, and let w_a be as described above. If either p_l or w_a is undefined there is nothing to propagate. Otherwise there is a monotone path from p_l to w_a, specifically follow the line of $v\ell^l$ from p_l to $\mathcal{H}^l$, then continue along the boundary of $F_s(i,j)$ to $\mathcal{H}^a$, and then follow $h\ell^a$ to w_a. Hence all the points in $La^F_{i,j+1}$ that are at least as high as w_a are reachable from p_l, see the blue path in Figure 3.2.

2) Propagating $L^R_{i,j}$

   a) Below $F_s(i,j)$: Let p_b be the lowest point of $Lb^R_{i,j}$ (note we may have $Lb^F_{i,j} = L^R_{i,j}$). If p_b lies above $h\ell^b$, then there is nothing to propagate. Otherwise, the reachable points on $Lb^F_{i,j+1}$ coming from monotone paths from p_b (that pass below $F_s(i,j)$) can be found by walking as low as possible through the cell. Specifically, if there is a point $Lb^F_{i,j+1}$ at the same height as p_b then we can walk horizontally directly to it, otherwise when we walk horizontally we bump into the boundary of $F_t(i,j)$ and follow it up to $\mathcal{I}^b$ (green path in Figure 3.3). In either case all higher points on $Lb^F_{i,j+1}$ are reachable.

   b) Above $F_s(i,j)$: Let p_a be the lowest point of $La^R_{i,j}$. If p_a lies above $h\ell^a$ then by walking horizontally to the right boundary of the cell (top blue path in Figure 3.3), we can reach all points of $La^F_{i,j+1}$ that are at least as high as p_a (note there may be no such points). Otherwise, there is a monotone path from p_a to w_a (if w_a is defined). There are two cases based on the relative heights of p_a and the point $\mathcal{H}^l$. If p_a lies below $\mathcal{H}^l$, then the monotone path walks horizontally from p_a to $v\ell^l$, then vertically on $v\ell^l$ to $\mathcal{H}^l$, then continues along the boundary of $F_s(i,j)$ to $\mathcal{H}^a$, and then horizontally to w_a. If p_a lies above $\mathcal{H}^l$, then the monotone path walks horizontally from p_a to the boundary of $F_s(i,j)$, then continues along the boundary of $F_s(i,j)$ to $\mathcal{H}^a$, and then horizontally to w_a. In either case all the points in $La^F_{i,j+1}$ that are at least as high as w_a are reachable.

▶ **Theorem 3.** *Given polygonal curves π of length n, σ of length m, and an interval $[s,t]$, the Fréchet gap decision problem, Problem 1, can be solved in $O(nm)$ time.*

Proof. First compute $L^F_{i,j}$ and $B^F_{i,j}$ for all $1 \leq i \leq n$, $1 \leq j \leq m$. Next initialize the reachable subset of left boundary of the entire relative free space diagram, i.e. $\cup_i L^R_{i,1}$. To do so, consider the entire free space of the left boundary, $\cup_i L^F_{i,1}$, and mark all points in this set that are reachable (by paths restricted to $\cup_i L^F_{i,1}$) from $(0,0)$. Note if $(0,0)$ is not in the free space, we return 'no' as the answer to the decision problem. Similarly initialize the reachable subset of bottom boundary of the entire relative free space diagram. Now propagate the reachable sets using any topological ordering of the cells (e.g. go in increasing column order, and for each column go by increasing row order). Specifically, for each cell $C(i,j)$ we use $L^R_{i,j}$ and $B^R_{i,j}$ (and $L^F_{i,j+1}$ and $B^F_{i+1,j}$) to compute $L^R_{i,j+1}$ and $B^R_{i+1,j}$, as described above. We then return 'yes' if (n,m) is in the reachable space and 'no' otherwise.

As for the running time, $L^F_{i,j}$ and $B^F_{i,j}$ take $O(1)$ time to compute per cell, and there are $O(nm)$ cells. Initializing the reachable sets then takes $O(n+m)$ time. As argued above, for any i,j, $L^R_{i,j}$ and $B^R_{i,j}$ are composed of at most two disjoint intervals hence propagating the reachable information to $L^R_{i,j+1}$ and $B^R_{i+1,j}$ takes $O(1)$ time per cell, and again there are $O(nm)$ cells, so this is the total running time. ◀

4 Finding the Relative Free Space Critical Events

In this section we describe the relative free space critical events, that is a polynomially sized subset of possible intervals, which must contain an optimal interval $[s^*, t^*]$. The relative free space events are significantly more complicated than the free space events for the standard Fréchet distance. The following definitions will be used throughout this section.

▶ **Definition 4.** Two free space cells $C(i,j)$ and $C(k,l)$ are *adjacent* if they share a horizontal or vertical boundary, i.e. $k=i$ and $|l-j|=1$, or $l=j$ and $|k-i|=1$. Call any monotone path from $(0,0)$ to (n,m) in the relative free space a *valid* path. Given any valid path p, the *cell sequence* of p, denoted $\mathsf{cp}(p) = (C_1, \ldots, C_{n+m-1})$, is the ordered sequence of cells p intersects (so $C_1 = C(1,1)$, $C_{n+m-1} = C(n,m)$).

We now define a number of other sequences determined by p. Let the entry point e_i be the point where p first intersects the cell C_i, and define the entry sequence of p as $\mathsf{entry}(p) = (e_1, \ldots, e_{n+m})$, where $e_1 = (0,0)$ and e_{n+m} is defined as (n,m). Let $\mathsf{int}(p) = (I_2, I_3, \ldots, I_{n+m-1})$ denote the sequence of boundary free space intervals passed by p, i.e. e_i lies on I_i. For horizontally adjacent cells $C(i,j)$ and $C(i,j+1)$ in the cell sequence, p either passes above or below $F_s(i,j)$, specifically if p intersects the vertical segment connecting $\mathcal{H}^a$ to the top boundary of $C(i,j)$ then p *passes above* $F_s(i,j)$, and otherwise p *passes below*. (Similarly define passing left or right for vertically adjacent cells.) This defines the *passing sequence* of p, denoted $\mathsf{pass}(p) = (h_1, \ldots, h_{n+m-1})$, where $h_i \in \{\text{above, below, left, right}\}$.

For the standard Fréchet distance, Alt and Godau [4] specified the following set of distance values, called the critical events, which must contain the optimal Fréchet distance.

- *Initialization* event: The minimum value ε such that $(0,0) \in F_\varepsilon$ and $(n,m) \in F_\varepsilon$.
- *Connectivity* events: For any cell C_i, the minimum ε such that L^F_i or B^F_i is non-empty, corresponding to the distance between a vertex of one curve and an edge of the other.
- *Monotonicity* events: Let I_j and I_k be two non-empty vertical free space boundary intervals in the same row with I_j left of I_k (or horizontal intervals in the same column). The minimum ε such that $\mathcal{I}^b_j \leq \mathcal{I}^a_k$, that is there is a monotone path between I_j and I_k.

Since any valid path can be decomposed into a set of row and column subpaths, proving that $d_{\mathcal{F}}(\pi, \sigma)$ is one the above defined critical events is a straightforward task.

For the Fréchet gap distance, the critical events will be a super-set of the standard Fréchet events. As an optimal gap is defined by an interval $[s,t]$, the events below can either be a value of s or a value of t. A *critical interval* is then any valid $s \leq t$ pair from the first

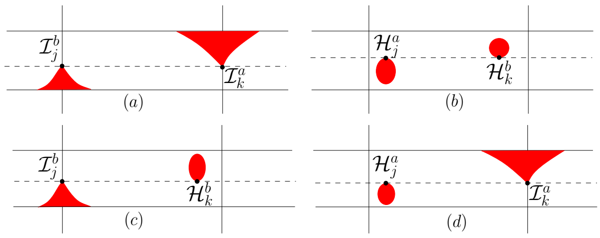


Figure 4.1 Opening of a horizontal passage.

three critical event types defined below. Additionally, there is now a fourth type called a floating monotonicity event. These events directly specify the s, t pair (i.e. these "events" are also "critical intervals"), and there are potentially an infinite number of such events.

1) *Initialization* events: The values $s = \min\{||\pi_0 - \sigma_0||, ||\pi_n - \sigma_m||\}$ and $t = \max\{||\pi_0 - \sigma_0||, ||\pi_n - \sigma_m||\}$. That is, the supremum of values for s such that $(0, 0) \notin F_s$ and $(n, m) \notin F_s$, and the minimum value of t such that $(0, 0) \in F_t$ and $(n, m) \in F_t$.
2) *Connectivity* events: For any row i and column j, the values $dist(\pi_{i-1}, \sigma_{j-1}\sigma_j)$, $dist(\pi_i, \sigma_{j-1}\sigma_j)$, $dist(\pi_{i-1}\pi_i, \sigma_{j-1})$, $dist(\pi_{i-1}\pi_i, \sigma_j)$, for either s or t. In other words for cell $C_{i,j}$, the maximum value s such that $\mathcal{H}^a$, $\mathcal{H}^b$, $\mathcal{H}^l$, or $\mathcal{H}^r$ are defined, or minimum value t such that $\mathcal{I}^a, \mathcal{I}^b$ (or similarly any of the other three cell boundary intervals) are defined. Note $\mathcal{I}^a, \mathcal{I}^b$ are first defined at the same location/value where $\mathcal{H}^r$ is last defined, yet we still regard these as separate events, one for s and the other for t. (For s this is when the free space intervals may break into two, and for t it is when the interval is first non-empty.)
3) *Standard Monotonicity* events: For any cells C_j, C_k in the same row with C_j left of C_k:
   (a) The value t such that $height(\mathcal{I}_j^b) = height(\mathcal{I}_k^a)$.
   (b) The value s such that $height(\mathcal{H}_j^a) = height(\mathcal{H}_k^b)$.
4) *Floating Monotonicity* events: For any cells C_j, C_k in the same row with C_j left of C_k:
   (a) Any pair s, t such that $height(\mathcal{I}_j^b) = height(\mathcal{H}_k^b)$.
   (b) Any pair s, t such that $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a)$.

Here $height()$ denotes the vertical coordinate of a point in the relative free space. Analogous definitions apply to the case when cells are in the same column. Note that depending on the geometry such events may not be defined.

Let S_s and S_t denote the set of values for s and t, respectively, determined by the initialization, connectivity and standard monotonicity critical events, and let $S_s \times S_t$ denote the corresponding set of valid critical intervals determined by these values. Let S_F be the set of s, t intervals determined by floating monotonicity events. The set of all critical intervals is then $S_I = S_F \cup (S_s \times S_t)$.

▶ **Lemma 5.** *S_I contains any optimal Fréchet gap interval $[s^*, t^*]$.*

Proof. For the sake of contradiction, suppose $[s^*, t^*] \notin S_I$. As $[s^*, t^*]$ is a feasible Fréchet gap, there must exist a valid path p in $F_{[s^*,t^*]}$. Let the cell, passing, entry, and interval sequences of p be $\mathsf{cp}(p) = (C_1, \ldots, C_{n+m-1})$, $\mathsf{pass}(p) = (h_1, \ldots, h_{n+m-1})$, $\mathsf{entry}(p) = (e_1, \ldots, e_{n+m})$, and $\mathsf{int}(p) = (I_2, I_3, \ldots, I_{n+m-1})$ (see Definition 4). In this proof we will show that if $[s^*, t^*] \notin S_I$ then the canonical form of p which, subject to having the same cell and passing sequences as p, locally remains as low and left as possible (i.e. follows the reachable free space propagation rules of Section 3), will also define a valid path in the free space after either increasing s^* or decreasing t^*.

Let $S_F^* = \{t \mid (s^*, t) \in S_F\}$ (which may possibly be empty). Since $[s^*, t^*] \notin S_I$ it must be that $t^* \notin S_F^*$. Also since $[s^*, t^*] \notin S_I$, either $s^* \notin S_s$ or $t^* \notin S_t$, and we will assume it is the

$t^* \notin S_t$ case (the $s^* \notin S_s$ case is argued similarly). Let $t_{init} = \max\{||\pi_0 - \sigma_0||, ||\pi_n - \sigma_m||\}$ be the corresponding t value of the initialization event, and hence $t_{init} \in S_t$. Note that because $[s^*, t^*]$ is feasible, $t_{init} \in [s^*, t^*]$ (as any valid path contains (π_0, σ_0) and (π_n, σ_m)). Since we assumed $t^* \notin S_t$, this implies $t_{init} \in [s^*, t^*)$, and so $[s^*, t^*) \cap S_t \neq \emptyset$ (moreover, $s^* \neq t^*$). So let $T = S_t \cup S_F^*$, and let t^- be the largest value in T which is $\leq t^*$, which we just argued must exist and $t^- \in [s^*, t^*)$ (in particular we later use that $(t^-, t^*] \cap T = \emptyset$).

We now show there is a valid path in the free space $F_{[s^*,t^-]}$, which is a contradiction as $t^- < t^*$, but $[s^*, t^*]$ was optimal. Specifically, we argue there is a valid canonical path p' in $F_{[s^*,t^-]}$ with the same cell and passing sequences as p. To this end, define $\mathsf{entry}(p') = (e'_1, \ldots, e'_{n+m})$ such that (i) $e'_1 = e_1 = (0,0)$, (ii) $e'_{n+m} = e_{n+m} = (n,m)$, and (iii) for $1 < i < n+m$, if I_i is vertical (resp. horizontal) then e'_i is the lowest (resp. leftmost) point in $F_{[s^*,t^-]} \cap I_i$ that is above (resp. right of) e'_{i-1}, and also above $\mathcal{H}^a_{i-1}$, (resp. right of $\mathcal{H}^r_{i-1}$) if h_{i-1} equals *above* (resp. *right*). We now argue that the points in this entry sequence are well defined, and hence p' is a valid path through $F_{[s^*,t^-]}$.

First observe that as $(t^-, t^*] \cap T = \emptyset$, initialization events cannot lie in this interval, and so e'_1 and e'_{n+m} must be in $F_{[s^*,t^-]}$. Now we inductively argue that for all other $1 < i < n+m$, e'_i is well defined. Again since $(t^-, t^*] \cap T = \emptyset$, $F_{[s^*,t^-]} \cap I_i$ is not empty for all $1 < i < n+m$. So fix an index β, and assume $e'_{\beta-1}$ is well defined. Without loss of generality assume I_β is a vertical edge (the horizontal case is handled similarly). If $I_{\beta-1}$ is on a horizontal edge then clearly, if $h_{\beta-1}$ equals *below*, $e'_\beta = \mathcal{I}^b_{\beta-1}$, which is well defined as s^* is fixed and $(t^-, t^*] \cap T = \emptyset$. If $h_{\beta-1}$ equals *above*, then in $F_{[s^*,t^-]}$ it must be that $e'_{\beta-1}$ is on the left of $\mathcal{H}^l_{\beta-1}$ and $height(\mathcal{H}^a_{\beta-1}) \leq height(\mathcal{I}^a_{\beta-1})$, as this was true in $F_{[s^*,t^*]}$ and we assumed there were no monotonicity events in $(t^-, t^*] \cap T$. In other words we can pass to the left and above the hole, and thus e'_β is well defined, Also note in this case that either $height(e'_\beta) = height(\mathcal{H}^a_{\beta-1})$ or $height(e'_\beta) = height(\mathcal{I}^b_{\beta-1})$.

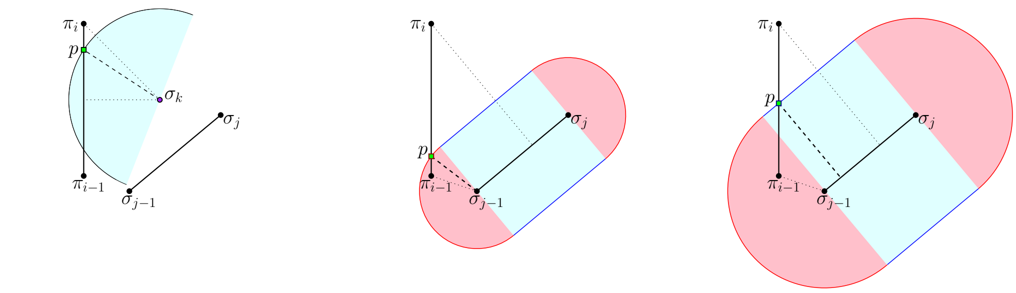
Now suppose $I_{\beta-1}$ is a vertical edge and moreover, let $I_\alpha, I_{\alpha+1}, \ldots, I_\beta$ be the earlier boundary intervals in this row, i.e. it is the maximal length contiguous subsequence of vertical boundary intervals ending at I_β. As $e'_{\beta-1}$ is well defined, we must have either $height(e'_{\beta-1}) = height(\mathcal{H}^a_j)$ for $j < \beta' - 1$ or $height(e'_{\beta-1}) = height(\mathcal{I}^b_j)$ for $j \leq \beta - 1$, as p' locally remained as low as possible (i.e. there must be a free space object responsible for pushing $e'_{\beta-1}$ to that height). Therefore if the point e'_β is not well defined then it must be that either $height(\mathcal{I}^a_{\beta-1}) < height(\mathcal{H}^a_j)$ or $height(\mathcal{I}^a_{\beta-1}) < height(\mathcal{I}^b_j)$ for $j \leq \beta - 1$. The latter case is not possible, as it implies there must have been a standard monotonicity event in $(t^-, t^*]$ (as clearly for $F_{[s^*,t^*]}$ there was a monotone path and so $height(\mathcal{I}^a_{\beta-1}) \geq height(\mathcal{I}^b_j)$), but we assumed $(t^-, t^*] \cap S_t = \emptyset$. For the former case, observe that $height(\mathcal{H}^a_j)$ is unchanged from $F_{[s^*,t^*]}$ to $F_{[s^*,t^-]}$, and so it would imply there was a floating monotonicity event in $(t^-, t^*] \cap S_F^*$, but we assumed this intersection was empty. ◀

4.1 Bounding the number of critical intervals

We now bound the number of critical intervals, i.e. $|S_I|$. An interval $[s,t] \in S_I$, is either in $S_s \times S_t$ or S_F. Now S_s (resp. S_t) has size[1] $O(n^3)$ as it contains one initialization event, $O(n^2)$ connectivity events, and $O(n^3)$ monotonicity events (just like the standard Fréchet case). As we consider all valid pairs from S_s and S_t, this gives an $O(n^6)$ bound on $|S_s \times S_t|$.

Bounding the size of S_F is significantly more complicated. In particular, the floating

[1] For simplicity, from this point onwards we assume without loss of generality that $m \leq n$ and only write sizes and running times with respect to n.



(a) Distance from σ_k **(b)** Distance from σ_{j-1} **(c)** Distance from $\sigma_{j-1}\sigma_j$

Figure 4.2 How point p determines s and t. In general segments may not lie in a single plane.

monotonicity events may give rise to an entire continuum of critical intervals. For example, consider the second type of floating monotonicity event (4b), shown in Figure 4.1. The value of $height(\mathcal{H}_j^a)$ is governed only by a function of s and the value of $height(\mathcal{I}_k^a)$ only by a function of t. These functions might be such that if we increase or decrease s, but keep $t - s$ constant (i.e. the gap value we are optimizing), $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a)$ remains an invariant. (Hence the term "floating" events.)

In this section we describe the functions which govern how s and t can vary such that $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a)$ remains an invariant. Ultimately our understanding of these function will yield a polynomially sized set of canonical critical intervals (determined by vertices of the arrangement of these functions), which must contain an optimum gap interval.

4.1.1 Function Description of Floating Monotonicity Events

Consider the floating monotonicity event type (4b) (similar statements will hold for type (4a)). Such an event is specified by a triple of indices, i, j, k, where i specifies an edge $\pi_{i-1}\pi_i$ (i.e. a row of the relative free space), j specifies an edge $\sigma_{j-1}\sigma_j$ (i.e. a column), and $k \geq j$ specifies a vertex σ_k (i.e. the right boundary of a column). The event occurs when $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a) = h$.

Geometrically, a fixed height h corresponds to a point p on $\pi_{i-1}\pi_i$. The point $\mathcal{H}_j^a$ is determined by s, and $\mathcal{I}_k^a$ by t. First lets understand $\mathcal{I}_k^a$. In order to have $h = height(\mathcal{I}_k^a)$, t must be such that $t = ||\sigma_k - p||$, and moreover p must be the higher (i.e. closer to π_i) of the possibly two points on $\pi_{i-1}\pi_i$ satisfying this condition (the other point determining $\mathcal{I}_k^b$). Consider the plane determine by π_{i-1}, π_i, and σ_k, and let $\pi_{i-1} = (0,0)$, $p = (0,h)$, and $\sigma_k = (\chi, \gamma)$ (see Figure 4.2a). Then as a function of h, t is described by the equation $t = \sqrt{\chi^2 + (\gamma - h)^2}$. Note that $\mathcal{I}_k^a$ is only defined when $t \in [t_1, t_2]$, where $t_1 = dist(\sigma_k, \pi_{i-1}\pi_i)$ and $t_2 = ||\sigma_k - \pi_i||$, and hence this equation is only relevant in this interval.

$height(\mathcal{H}_j^a)$ on the other hand is determined by s, however the relationship is a bit more complicated. Observe that $\mathcal{H}_j^a$ is the only point on the horizontal line $h\ell_j^a$ that is in the set $F_s(i,j)$, meaning the point on $\sigma_{j-1}\sigma_j$ that $\mathcal{H}_j^a$ corresponds to must be the closest point on $\sigma_{j-1}\sigma_j$ to p (see Figure 4.2b and Figure 4.2c). If this closest point is either σ_{j-1} or σ_j, then the form of the equation for s in terms of h is the same as it was t, namely $s = \sqrt{\alpha^2 + (\beta - h)^2}$ (where α, β are now the coordinates of either σ_{j-1} or σ_j). Otherwise this closest point is in the interior of $\sigma_{j-1}\sigma_j$ in which case the equation is of the form $s = c \cdot h + d$, for some constants c and d (since as one walks along a line, the distance to another fixed line is given by a linear equation). Similar to $\mathcal{I}_k^a$, $\mathcal{H}_j^a$ is only defined when $s \in [s_1, s_2]$, where $s_1 = dist(\sigma_{j-1}\sigma_j, \pi_{i-1}\pi_i)$ and $s_2 = dist(\sigma_{j-1}\sigma_j, \pi_i)$, and hence this

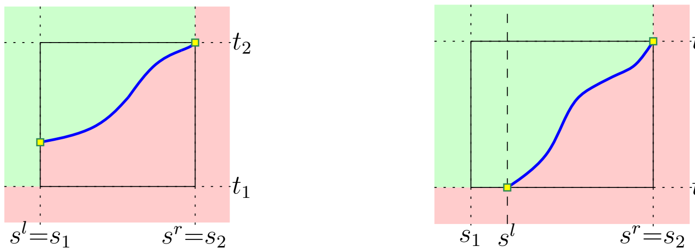


Figure 4.3 Two cases for curve piece $f_{i,j,k}$, and shaded satisfying points in s, t parametric space.

equation is only relevant in this interval.

Now that we have a description of $height(\mathcal{H}_j^a)$ in terms of s and $height(\mathcal{I}_k^a)$ in terms of t, we can describe the function for t in terms of s, denoted $f_{i,j,k}(s)$, which describes when $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a) = h$. There are two cases based on the form of the function describing s.

Interior of $\sigma_{j-1}\sigma_j$ case:

$$s = c \cdot h + d, \quad t = \sqrt{\chi^2 + (\gamma - h)^2} \quad \Rightarrow \quad f_{i,j,k}(s) = \sqrt{\left(\frac{s-d}{c} - \gamma\right)^2 + \chi^2}$$

Endpoint of $\sigma_{j-1}\sigma_j$ case:

$$s = \sqrt{\alpha^2 + (\beta - h)^2}, \quad t = \sqrt{\chi^2 + (\gamma - h)^2} \quad \Rightarrow \quad f_{i,j,k}(s) = \sqrt{(\sqrt{s^2 - \alpha^2} + (\beta - \gamma))^2 + \chi^2}$$

To summarize, $f_{i,j,k}(s)$ is composed of at most three hyperbola[2] pieces, and is only (possibly) defined within the region $s \in [s_1, s_2]$ and $t \in [t_1, t_2]$, see Figure 4.3. Also, the geometry of the problem implies that when $f_{i,j,k}(s)$ is defined it is a monotone increasing function. Hence the intersection of $f_{i,j,k}$ with the bounding box $[s_1, s_2] \times [t_1, t_2]$ is connected, and so rather than using this box to define $f_{i,j,k}$, we instead say $f_{i,j,k}$ is either completely undefined or is defined only in the interval $[s^l, s^r]$ where s^l and s^r are the s coordinate where $f_{i,j,k}$ respectively enters and leaves the bounding box. Note that one can argue if $f_{i,j,k}$ is defined then $s^r = s_2$, however, it may be that $s^l > s_1$ (if the closest point to $\sigma_{j-1}\sigma_j$ is lower on $\pi_{i-1}\pi_i$ than the closest point to σ_k).

The exact form of the equation $f_{i,j,k}(s)$ is not needed in our analysis, however, the above discussion implies the following simple observation which will be used later.

▶ **Observation 6.** *In the s, t parametric space $f_{i,j,k}$ is either undefined or defines a constant complexity monotonically increasing curve piece, with endpoints at values $s^l_{i,j,k} \leq s^r_{i,j,k}$. In particular, $f_{i,j,k}$ has only a constant number of local minima and maxima (i.e. points of tangency) with respect to translations of the line $t = s$.*

Note that for (4a), i.e. when $height(\mathcal{I}_j^b) = height(\mathcal{H}_k^b)$, $f_{i,j,k}$ can be defined similarly, and the above observation again holds. One must also define functions for the analogous events in the free space columns. Such functions are again determined by triples i, j, k, however now i, j refer to rows and k to the column. Below we will denote these functions by $g_{i,j,k}$.

[2] Technically, the endpoint case is not a hyperbola, though it is similar.

4.1.2 Events minimizing the gap

As discussed above each $f_{i,j,k}$, if defined, gives an entire continuum of critical intervals. However, ultimately we are only interested in feasible intervals which minimize the gap, and this will allow us to reduce this continuum to a polynomial number of canonical intervals. This polynomially sized set is determined not only by the $f_{i,j,k}$, but also by the other types of critical events. Note that initialization (1), connectivity (2), and standard monotonicity events (3) only define constraints on either just s or t, whereas the $f_{i,j,k}$ and $g_{i,j,k}$ define a continuum of $[s,t]$ intervals. Hence to put them on equal footing we think of all of them as defining constraints in the two dimensional s,t parametric space.

First observe that in the parametric space, for any point (s,t) of interest, $0 \le s \le t$, and so we only consider points in the first quadrant that are above the line $t = s$. Initialization, connectivity, and standard monotonicity events are simply defined by horizontal or vertical lines. Specifically, for each such event the points satisfying the corresponding constraint are those above (resp. left of) the corresponding horizontal (resp. vertical) line:

1) Initialization events: $\quad s \le \alpha_0, \quad t \ge \beta_0$
   Where $\alpha_0 = \min\{||\pi_0 - \sigma_0||, ||\pi_n - \sigma_m||\}$ and $\beta_0 = \max\{||\pi_0 - \sigma_0||, ||\pi_n - \sigma_m||\}$.
2) Connectivity events: $\quad s \le \alpha^l_{i,j}$ or $s \le \alpha^b_{i,j}, \quad t \ge \beta^l_{i,j}$ or $t \ge \beta^b_{i,j}$
   Where the $\alpha_{i,j}$ and $\beta_{i,j}$ are *vertex-edge* distances, that is $\alpha^l_{i,j} = \beta^l_{i,j} = dist(\pi_{i-1}\pi_i, \sigma_{j-1})$ or $\alpha^b_{i,j} = \beta^b_{i,j} = dist(\pi_{i-1}, \sigma_{j-1}\sigma_j)$. Note defining both $\alpha_{i,j}$ and $\beta_{i,j}$ is not necessary but useful to distinguish constraints on s from those on t.
3) Standard Monotonicity events: $\quad s \le \alpha_{i,(j,k)}$ or $s \le \alpha_{(i,j),k}, \quad t \ge \beta_{i,(j,k)}$ or $t \ge \beta_{(i,j),k}$
   Which happens when the free space is such that $\alpha_{i,(j,k)} = height(\mathcal{H}^a_{i,j}) = height(\mathcal{H}^b_{i,k})$ or $\alpha_{(i,j),k} = height(\mathcal{H}^a_{i,k}) = height(\mathcal{H}^b_{j,k})$, and when $\beta_{i,(j,k)} = height(\mathcal{I}^b_{i,j}) = height(\mathcal{I}^a_{i,k})$ or $\beta_{(i,j),k} = height(\mathcal{I}^b_{i,k}) = height(\mathcal{I}^a_{j,k})$.
4) Floating Monotonicity events: $t \ge f_{i,j,k}(s)$ for $s \in [s^{lf}_{i,j,k}, s^{rf}_{i,j,k}]$, or $t \ge g_{i,j,k}(s)$ for $s \in [s^{lg}_{i,j,k}, s^{rg}_{i,j,k}]$. Note depending on the geometry such constraints may not be defined.

Note that the first three event types each partition the entire parametric space into two connected sets, those which either satisfy or do not satisfy the constraint. The $f_{i,j,k}$ (and $g_{i,j,k}$) can also be thought of in this way, see the shaded regions in Figure 4.3. Specifically, (s,t) satisfies the constraint if $t \ge t_1$, $s \le s_2$, and if $s \in [s^l, s^r]$ then (s,t) must lie above the curve $f_{i,j,k}$. Otherwise (s,t) does not satisfy the constraint.

Any valid path in the relative free space must have a well defined cell sequence $(C_1, \ldots, C_{n+m-1})$ and passing sequence $\mathsf{pass}(p) = (h_1, \ldots, h_{n+m-1})$ (see Definition 4). Moreover, such a pair of sequences precisely determine a subset of the constraints defined above, such that there is a valid path with this cell and passing sequence if and only if all constraints in the subset are satisfied (this is implied by Lemma 5). In other words, for a given cell and passing sequence we want to solve the optimization problem:

$$
\begin{aligned}
\min \quad & t - s \\
\text{subject to} \quad & \\
& t \geq s \geq 0, \\
& s \leq \alpha, \text{ where } \alpha = \min\{\alpha_0, \alpha^l_{i_1,j_1}, \alpha^b_{i_2,j_2}, \alpha_{i_3,(j_3,k_3)}, \alpha_{(i_4,j_4),k_4}\} \text{ over all} \\
& \qquad (i_1,j_1) \in Z_1, (i_2,j_2) \in Z_2, (i_3,j_3,k_3) \in Z_3, (i_4,j_4,k_4) \in Z_4 \\
& t \geq \beta, \text{ where } \beta = \max\{\beta_0, \beta^l_{i_5,j_5}, \beta^b_{i_6,j_6}, \beta_{i_7,(j_7,k_7)}, \beta_{(i_8,j_8),k_8}\} \text{ over all} \\
& \qquad (i_5,j_5) \in Z_5, (i_6,j_6) \in Z_6, (i_7,j_7,k_7) \in Z_7, (i_8,j_8,k_8) \in Z_8 \\
& t \geq f_{i,j,k}(s), \text{ for } s \in [s^{lf}_{i,j,k}, s^{rf}_{i,j,k}] \;\; \forall (i,j,k) \in Z_9 \\
& t \geq g_{i,j,k}(s), \text{ for } s \in [s^{lg}_{i,j,k}, s^{rg}_{i,j,k}] \;\; \forall (i,j,k) \in Z_{10}
\end{aligned}
$$

where the sets $Z_1, \ldots, Z_{10}$ of index tuples are determined naturally by the cell and passing sequences. (We skip the tedious and unenlightening description of these sets.)

Clearly the optimal value of this optimization problem must lie on the boundary of at least one constraint. In particular, the optimum lies either at the intersection point of the boundaries of two constraints, or at a local minimum of one of the boundary constraints, with respect to the objective of minimizing $t-s$. By Observation 6, each $f_{i,j,k}$ or $g_{i,j,k}$ has at most a constant number of local minima, and as the boundaries of all other constraints are straight lines, this is also true for every boundary function. Thus we have now determined the set of canonical critical intervals discussed earlier in this section.

▶ **Lemma 7.** *The above defined constraints, determined by all types of critical events, determine an $O(n^6)$ sized set of canonical critical intervals, i.e. (s,t) pairs, that must contain an optimal gap $[s^*, t^*]$.*

Proof. Any optimal gap determines a cell and passing sequence of some valid path in the corresponding relative free space. Above it was discussed how such sequences determine a subset of constraints, where the optimum gap width is determined either at an intersection of the boundaries of two constraints or at a local minimum of an $f_{i,j,k}$ or $g_{i,j,k}$. Now a priori we do not know the cell and passing sequence of a path determining an optimal gap, hence we will consider them all. So consider the arrangement of all planar curves defined by the boundaries of any of the possible constraints defined above. There are a constant number of initialization constraints, $O(n^2)$ possible connectivity constraints, and $O(n^3)$ possible standard or floating monotonicity constraints. Due to the particularly nice form of these curves, each pair intersect at most a constant number of times, and hence there are $O(n^6)$ intersections overall. Moreover, as discussed above, each curve has only a constant number of local minima with respect the the objective of minimizing $t-s$. Hence this arrangement determines as set of $O(n^6)$ points, at least one of which realizes the minimum gap width. ◀

The above discussion also implies the following observation.

▶ **Observation 8.** *Whether or not a given (s,t)-pair is feasible for the Fréchet gap problem, is solely determined by which constraints the point satisfies or does not satisfy. So consider the arrangement of curves determined by the boundaries of all the constraint types discussed above. Then within the interior of a given cell of the arrangement all (s,t)-pairs are thus either all feasible or all infeasible.*

5 Exact Computation of the Fréchet Gap Distance

The $O(n^6)$ critical intervals given by Lemma 7 together with the $O(n^2)$ decider of Theorem 3, naively give only an $O(n^8)$ algorithm for computing the Fréchet gap distance, as there is no immediate linear ordering to search over the events. However, here we give a much faster $O(n^5 \log n)$ time algorithm to compute the Fréchet gap distance exactly.

The standard Fréchet distance is computed in $O(n^2 \log n)$ time by searching over the $O(n^3)$ critical events with an $O(n^2)$ time decision procedure. This searching originally was done with parametric search [4], though for our purposes the simpler sampling based approach of [19] is more relevant.

Searching is a far more challenging task in the Fréchet gap setting. Specifically, in the standard Fréchet case there is a linear ordering of the critical events, and in this ordering all events are infeasible up until the true Fréchet distance, and then feasible afterwards. However, in our two dimensional parametric space there is no such natural linear ordering. Moreover, recall that even if an interval $[s, t]$ is feasible, it does *not* imply $[s, t]$ contains an optimal gap as a subinterval. Crucially however, it still holds that the feasible points form a connected set in the parametric space.

▶ **Lemma 9.** *In the parametric space, the set of feasible (s, t) pairs is a connected set.*

Proof. Given any two feasible points (s_1, t_1) and (s_2, t_2), we describe a path between them consisting of only feasible points (thus implying the feasible set is connected). Observe that if a point (s, t) is feasible, then any point (s', t') such that $s' \leq s$ and $t' \geq t$ is also feasible. This implies the line segments $(0, t_1)(s_1, t_1)$ and $(0, t_2)(s_2, t_2)$ consist solely of feasible points. Again applying this observation, the segment $(0, t_1)(0, t_2)$ consists solely of feasible points. Thus the path $((s_1, t_1), (0, t_1), (0, t_2), (s_2, t_2))$, consists only of feasible points. ◀

The algorithm for exactly computing the Fréchet gap distance uses the following subroutines:

- **deciderPoint**(s, t): Decides whether or not the pair (s, t) is feasible, in $O(n^2)$ time.
- **deciderLine**(c): Given a positive number c, returns "below" if there is any feasible (s, t)-pair with $t - s \leq c$, and returns "above" otherwise. The running time is $O(n^5)$.
- **sample**(r): Samples r (s, t)-pairs, independently and uniformly at random, from the set of $O(n^6)$ canonical critical pairs of Lemma 7. The running time is $O(r)$.
- **sweep**(c_1, c_2): Returns the set of all canonical critical (s, t)-pairs of Lemma 7 such that $c_1 \leq t - s \leq c_2$, in $O((n^3 + k) \log n)$ time, where k is the number of such critical pairs.

First observe the subroutine **deciderPoint**(s, t) is given by Theorem 3. **deciderLine**(c) is computed as follows. First compute the intersection points of the line, $t - s = c$, with the $O(n^3)$ boundaries of all the constraints discussed in Section 4.1.2. Since these constraints are horizontal/vertical lines or $f_{i,j,k}/g_{i,j,k}$, by Observation 6, there are $O(n^3)$ intersection points. Thus calling **deciderPoint** on each of these intersection points, takes $O(n^5)$ time as **deciderPoint** takes $O(n^2)$ time. By Observation 8, if all these point queries return infeasible, then all points on the line $t - s = c$ are infeasible. In this case, since by Lemma 9 the feasible region is connected, any optimal gap pair must lie above the the line $t - s = c$. On the other hand, again by by Lemma 9, if one of the point queries returned true then any optimal gap pair must lie below (or on) the line $t - s = c$.

The subroutine **sample**(r) is also straightforward. Specifically, every canonical critical pair is either a local minima or an intersection of the boundaries of two constraints from Section 4.1.2. Thus in order to sample a canonical critical pair, we sample either one or two constraints[3], where whether we sample one or two is done in proportion to the number of pairs versus single constraints. Each constraint is determined by either a pair or triple of indices (and a few bits, such as whether the side of bottom of a cell, etc.), and hence each can be sampled in $O(1)$ time (again done proportionally to the number of triples versus pairs of indices). Thus r canonical pairs can be sampled in $O(r)$ time.

Thus what remains is to describe the subroutine **sweep**, for which we have the following.

▶ **Lemma 10.** *Given two real values $0 \le c_1 \le c_2$, one can compute the set of all canonical critical (s,t)-pairs of Lemma 7 such that $c_1 \le t-s \le c_2$, in $O((n^3+k)\log n)$ time, where k is the number of such critical pairs. This algorithm is denoted* **sweep**(c_1, c_2).

Proof. It is well known that one can compute the set of all k intersection points of a set of m x-monotone constant-complexity curves in $O((m+k)\log m)$ time using a horizontal sweep line in the standard sweep line algorithm of Bentley and Ottmann [8]. In our case the curves are given by the $O(n^3)$ constraints of Section 4.1.2, clipped to only be defined in the region bounded by the lines $t-s=c_1$ and $t-s=c_2$. The constraints with straight line boundaries are s-monotone, and by Observation 6 so are the $f_{i,j,k}$ and $g_{i,j,k}$. Thus the claim follows by applying the standard sweep line algorithm to our case. Note that our problem involves degenerate horizontal line segments, which can still be handled (with some care) when using a horizontal sweep line, though alternatively one could avoid such issues using a diagonal $t-s=c$ sweep line in which case one should first cut the $f_{i,j,k}/g_{i,j,k}$ into pieces at their local maxima/minima (with respect to the line $t-s=c$), to maintain monotonicity. ◀

$\mathcal{R}$ = **sample**(αn^4) // α a sufficiently large constant
Sort $\widehat{\mathcal{R}} = \{c = t-s \mid (s,t) \in \mathcal{R}\}$ in increasing order
Binary search over $\widehat{\mathcal{R}}$ using **deciderLine**(c) for the interval $[c_1, c_2]$ s.t. **deciderLine**(c_1) = above and **deciderLine**(c_2) = below
// Set initial values $c_1 = 0$, $c_2 = \infty$
$\mathcal{S}$ = **sweep**(c_1, c_2)
Call **deciderPoint**(s,t) on each $(s,t) \in \mathcal{S}$, and return the feasible pair with smallest $t-s$ value.

Algorithm 1: Computing the Fréchet gap distance

The algorithm for computing the Fréchet gap distance is shown in Algorithm 1. We need the following lemma to bound the number of critical pairs that we end up searching over.

▶ **Lemma 11.** *Let $[c_1, c_2]$ be the interval described in Algorithm 1. Then with exponentially high probability, this interval contains $O(n^3)$ canonical critical pairs.*

Proof. Let $\mathcal{C}$ be the set of all canonical critical pairs as described in Lemma 7, thus $|\mathcal{C}| \le \beta n^6$ for some constant β. Let $\widehat{\mathcal{C}} = \{c = t-s \mid (s,t) \in \mathcal{C}\}$, and let $\widehat{\mathcal{R}}$ be the sampled subset of $\widehat{\mathcal{C}}$ as described in Algorithm 1. Note that multiple values in $\mathcal{C}$ may map to a single value in $\widehat{\mathcal{C}}$. This technicality is discussed below, but for now assume $|\mathcal{C}| = |\widehat{\mathcal{C}}|$.

Consider the sorted placement of the values in $\widehat{\mathcal{C}}$ along the real line, and let x be any point on the real line. Let U^+ (resp. U^-) be the closest n^3 values from $\widehat{\mathcal{C}}$ larger (resp.

[3] Note the number of local minima per constraint and the number of times two constraints intersect is a constant, but the constant may be more than one. Thus technically the described sampling is not truly uniform. One can make it uniform, though this distinction is irrelevant for our asymptotic analysis.

smaller) than x. Suppose $|U^+| = n^3$ (note it may happen that $|U^+| < n^3$, if x is large enough). Then the probability it does not contain a value in $\widehat{\mathcal{R}}$ is

$$\left(1 - \frac{|U^+|}{\beta n^6}\right)^{\alpha n^4} = \left(1 - \frac{n^3}{\beta n^6}\right)^{\alpha n^4} \leq \exp\left(\frac{-\alpha n^7}{\beta n^6}\right) = \exp(-(\alpha/\beta)n) \leq e^{-\hat{c}n}$$

for any constant $\hat{c}$, by choosing α sufficiently large. A similar statement holds for $|U^-|$, and thus taking the union bound, a similar statement holds simultaneously for both U^- and U^+.

Note there are only a polynomial number of possibilities for each value in $\widehat{\mathcal{R}}$ (specifically $O(n^6)$). Thus by setting x to each one of these values, and taking the union bound, it holds that between any two adjacent values in $\widehat{\mathcal{R}}$ (or in the unbounded end intervals), with high probability there are at most $O(n^3)$ values of $\widehat{\mathcal{C}}$, thus implying the claim.

As mentioned above, technically multiple pairs in $\mathcal{C}$ may map to a single value in $\widehat{\mathcal{C}}$. This can be remedied by treating $\widehat{\mathcal{C}}$ as a multi-set. Then by defining an arbitrary order over multi-values, the above analysis will still hold, except potentially at the endpoints of the interval $[c_1, c_2]$ (as we include *all* critical pairs with these values). Observe however that (in the absolute worst case) there are at most $O(n^3)$ pairs which get mapped to either value c_1 or c_2 in $\widehat{\mathcal{C}}$, and so the lemma holds. ◀

Note that there is a huge amount of slack in the above argument, and in more than one way. Specifically, even though we have an exponentially high probability bound, it can be further improved by taking a larger random sample. Also we could have argued, with polynomially high probability, that the number of canonical critical pairs in $[c_1, c_2]$ is only $O(n^2 \log n)$ (taking more care in the argument at the endpoints). However, ultimately this would not change the asymptotic running time, as the real bottleneck for the algorithm is in searching with the $O(n^5)$ time **deciderLine**.

▶ **Theorem 12.** *Given polygonal curves π and σ, each of length at most n, Algorithm 1 computes the Fréchet gap distance in $O(n^5 \log n)$ time.*

Proof. The correctness of Algorithm 1 has essentially already been argued. Specifically, the random sample $\mathcal{R}$ partitions the real line into intervals based on the values in $\widehat{\mathcal{R}}$. One of these intervals contains the optimal gap width, implying the interval $[c_1, c_2]$ found by searching using **deciderLine**(c) is well defined. Moreover, $\mathcal{S}$ contains a canonical critical pair with optimal gap width as **sweep**(c_1, c_2) returns all canonical critical pairs in the region bounded by the lines $t - s = c_1$ and $t - s = c_2$, and by Lemma 7 the set of canonical critical pairs contains a pair with optimal gap width. As **deciderPoint** is called on all pairs in $\mathcal{S}$, the algorithm will find this optimal gap pair.

For the running time, calling **sample**(αn^4) takes $O(n^4)$ time. Sorting $\widehat{\mathcal{R}}$ takes $O(n^4 \log n)$ time, and searching over $\widehat{\mathcal{R}}$ takes $O(n^5 \log n)$ time as **deciderLine** takes $O(n^5)$ time. By Lemma 10, **sweep**(c_1, c_2) takes $O((n^3 + |\mathcal{S}|) \log n)$ time. Calling **deciderPoint** on each pair in $\mathcal{S}$ takes $O(|\mathcal{S}|n^2)$ time, as **deciderPoint** takes $O(n^2)$ time. By Lemma 11, with high probability $|\mathcal{S}| = O(n^3)$, so sweeping and all **deciderPoint** calls combined take $O(n^5)$ time. Thus the overall time is $O(n^5 \log n)$, i.e. dominated by the time to search with **deciderLine**. ◀

6 Approximation

In this section, we propose an efficient algorithm to approximate the Fréchet gap distance, based on the following simple fact. Let d_o be the average of the starting and ending vertex pair distances of π and σ, that is $d_o = \frac{d_b + d_e}{2}$ where $d_b = ||\pi_0 - \sigma_0||$ and $d_e = ||\pi_n - \sigma_m||$.

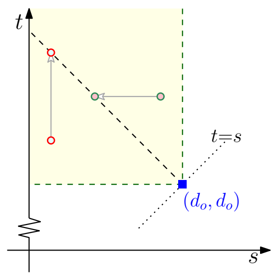


Figure 6.1 Two centered points and their projections.

▶ **Observation 13.** *If a parametric point* (s,t) *is feasible then* $s \le d_o \le t$.

This implies we only need to consider parametric points such that $s \le d_o \le t$, which we call *centered* points. Define the *radius* of any such point (s,t) to be $\mathsf{r}_{s,t} = \max\{t - d_o, d_o - s\}$, and define the *projection* to be $proj(s,t) = (d_o - \mathsf{r}_{s,t}, d_o + \mathsf{r}_{s,t})$. See Figure 6.1.

Observe that in order to get a 2-approximation it suffices to restrict our attention to projected points (as $[s,t] \subseteq [d_o - \mathsf{r}_{s,t}, d_o + \mathsf{r}_{s,t}]$ for any centered point (s,t)), and the advantage is that projected points are more nicely behaved. Specifically, projected points define a linear ordering by the parameter r with the nice property that if $(d_o - \mathsf{r}, d_o + \mathsf{r})$ is feasible then for any $\mathsf{r}' \ge \mathsf{r}$ it holds that $(d_o - \mathsf{r}', d_o + \mathsf{r}')$ is also feasible. Moreover, below we show that the $O(n^6)$ critical intervals of Lemma 7, can be reduced to $O(n^3)$ in this setting, intuitively since now there is only a single parameter r, rather than independent s and t parameters.

6.1 Simplification of Critical Events

In Section 4.1.2 we described a set of four different types of constraints over the (s,t) parametric space (relating to initialization, connectivity, and standard and floating monotonicity events), and saw that the Fréchet gap distance is realized by minimizing $t - s$ over some (unknown) subset of these constraints. Recall from that section that each such constraints partitions the parametric space into two sets, those satisfying and those violating that constraint. Label these $O(n^3)$ constraints in an arbitrary fashion from $1, \dots, cn^3$, and for the $\imath$th constraint let $\mathcal{P}_\imath$ denote the set of satisfying points in the parametric space (which is a single connected region). We will assume that the $\mathcal{P}_\imath$ are clipped to the subset of centered points such that $s \le d_o \le t$, as we know any optimal gap pair must lie in this region.

Note that any given constraint in Section 4.1.2 is satisfied by lying to the left and above a straight line or nicely behave monotonically increasing function, hence we have the following.

▶ **Observation 14.** *If* $(s,t) \in \mathcal{P}_\imath$ *then* $(s',t') \in \mathcal{P}_\imath$ *for any* $s' \le s$, $t' \ge t$. *This implies:*
1. $proj(\mathcal{P}_\imath) \subseteq \mathcal{P}_\imath$, *where* $proj(\mathcal{P}_\imath) = \{proj(s,t) \mid (s,t) \in \mathcal{P}_\imath\}$.
2. *If* $(d_o - \mathsf{r}, \mathsf{r} - d_o) \in proj(\mathcal{P}_\imath)$ *then* $(d_o - \mathsf{r}', \mathsf{r}' - d_o) \in proj(\mathcal{P}_\imath)$ *for any* $\mathsf{r}' \ge \mathsf{r}$.

For each $\mathcal{P}_\imath$ let $D_\imath$ be the minimum radius of a point in $proj(\mathcal{P}_\imath)$, and let $\mathcal{D}$ denote the set of all $D_\imath$. Note that each $\mathcal{P}_\imath$ is a constant complexity region, and so computing $D_\imath$ is a constant time operation.

▶ **Lemma 15.** *There is a value* $D \in \mathcal{D}$ *such that* $[d_o - D, d_0 + D]$ *is a 2-approximation to an optimal Fréchet gap, that is* $[d_o - D, d_0 + D]$ *is feasible and* $D \le d_{\mathcal{G}}(\pi, \sigma)$.

Proof. Let (s^*, t^*) be a point realizing the Fréchet gap distance. Then (s^*, t^*) is determined by some k constraints from Section 4.1.2, and let $I = \{\imath_1, \imath_2, \dots \imath_k\}$ be the set of indices of these constraints. Specifically, in that section we argued any point in $\bigcap_{\imath \in I} \mathcal{P}_\imath$ is feasible and (s^*, t^*) is the point in the intersection with minimum gap.

Let $D = \max_{\imath \in I} D_\imath$. Since for all $\imath$, $(d_o - D_\imath, d_o + D_\imath) \in proj(\mathcal{P}_\imath)$, the second part of Observation 14 implies $(d_o - D, d_o + D) \in \bigcap_{\imath \in I} proj(\mathcal{P}_\imath)$. The first part of Observation 14 implies any point in $\bigcap_{\imath \in I} proj(\mathcal{P}_\imath)$ is also in $\bigcap_{\imath \in I} \mathcal{P}_\imath$, and hence $(d_o - D, d_o - D)$ is feasible.

Let $proj(s^*, t^*) = (d_o - \mathsf{r}^*, d_o + \mathsf{r}^*)$, which is in $\bigcap_{\imath \in I} proj(\mathcal{P}_\imath)$ as $(s^*, t^*) \in \bigcap_{\imath \in I} \mathcal{P}_\imath$. For some index $\imath$, $D = D_\imath$, and since each $D_\imath$ is defined as the minimum radius of a point in $proj(\mathcal{P}_\imath)$, $(d_o - D, d_o + D)$ must therefore be the minimum radius point in $\bigcap_{\imath \in I} proj(\mathcal{P}_\imath)$, and so clearly $\mathsf{r}^* \geq D$. By definition, $r^* = \max\{d_o - s^*, t^* - d_o\}$ and so $D \leq \max\{d_o - s^*, t^* - d_o\} \leq t - s = d_\mathcal{G}(\pi, \sigma)$. ◀

6.2 Approximate Decider

Here we show how to efficiently convert any constant factor approximation into a $(1+\varepsilon)$-approximation, which is relevant as the previous section proved one of the $O(n^3)$ values in $\mathcal{D}$ is a 2-approximation. Specifically, we seek an efficient version of the following decider.

▶ **Definition 16.** **appDeciderLine**(c, ε): Given positive numbers c, ε, returns "true" if $d_\mathcal{G}(\pi, \sigma) \leq c$, and returns "false" if $d_\mathcal{G}(\pi, \sigma) > (1+\varepsilon)c$. Either "true" or "false" can be returned if $d_\mathcal{G}(\pi, \sigma) \in (c, (1+\varepsilon)c]$.

▶ **Lemma 17.** *There exist an $O(n^2/\varepsilon)$ time algorithm for* **appDeciderLine**(c, ε).

Proof. By Observation 13, $d_o \in [s, t]$ for any feasible interval $[s, t]$. Thus any feasible interval $[s, t]$ with $t - s \leq c$ is contained in the interval $[d_o - c, d_o + c]$, and hence we restrict our attention to this interval. We cover this interval with successive overlapping subintervals of width $(1+\varepsilon)c$, and each shifted by $c\varepsilon$ from the previous one. Specifically, let S_g be the set of subintervals of the form $[d_o - c + i(c\varepsilon), d_o - c + i(c\varepsilon) + (1+\varepsilon)c]$, for $i = 0, \ldots, \lceil \frac{1}{\varepsilon} \rceil$ (Note to make calculations easier below we stop at $d_o - c + \lceil \frac{1}{\varepsilon} \rceil (c\varepsilon) + (1+\varepsilon)c \geq d_o + (1+\varepsilon)c$ rather than $d_o + c$). Our algorithm for **appDeciderLine**(c, ε) simply checks each one of these intervals for feasibility using **deciderPoint**, and if any interval returns "true" then it returns "true" and otherwise it returns "false". **deciderPoint** correctly checks feasibility in $O(n^2)$ time by Theorem 3 and we are testing, $O(1/\varepsilon)$ intervals, so the running time bound is immediate. We now prove this procedure satisfies the properties of Definition 16.

If $d_\mathcal{G}(\pi, \sigma) \leq c$, then there is a feasible interval $[s, t]$ with $t - s \leq c$, which implies $[s, t] \subseteq [s, s+c] \subseteq [d_o - c + j(c\varepsilon), d_o - c + j(c\varepsilon) + (1+\varepsilon)c]$ where $j = \lfloor \frac{(s - d_o + c)}{c\varepsilon} \rfloor$. One can easily verify that $0 \leq j \leq \lceil 1/\varepsilon \rceil$, and so this interval is in S_g. Hence at least one interval in S_g is feasible (as containing a feasible subinterval implies feasibility), and so **appDeciderLine**(c, ε) returns "true". On the other hand, if $d_\mathcal{G}(\pi, \sigma) > (1+\varepsilon)c$, then no interval in S_g is feasible as each interval in S_g has width $(1+\varepsilon)c$, and so **appDeciderLine**(c, ε) returns "false". Finally, if $c < d_\mathcal{G}(\pi, \sigma) \leq c(1+\varepsilon)$, then **appDeciderLine**(c, ε) returns "false" or "true", and we don't care which one. ◀

Using binary search the above **appDeciderLine**(c, ε) can be used to convert a constant factor approximation (i.e. constant spread interval) into a $(1+\varepsilon)$-approximation.

▶ **Lemma 18.** *Given a value $c \geq 0$, one decide whether $d_\mathcal{G}(\pi, \sigma) > c$, or $d_\mathcal{G}(\pi, \sigma) < c$, or obtain $(1+\varepsilon)$ approximation to $d_\mathcal{G}(\pi, \sigma)$, which can be done in $O(n^2/\varepsilon)$ time.*

Proof. If $c = 0$, just test whether $[d_o, d_o]$ is feasible, and if not return $d_\mathcal{G}(\pi, \sigma) > c$. Otherwise, call **appDeciderLine**(c, ε') and **appDeciderLine**$(c/(1+2\varepsilon'), \varepsilon')$, for a value ε'

to be determined shortly. Taking the contrapositive of the statements in Definition 16, if **appDeciderLine**(c, ε) returns "true" then $d_{\mathcal{G}}(\pi, \sigma) \le c(1+\varepsilon)$, and if **appDeciderLine**(c, ε) returns "false" then $d_{\mathcal{G}}(\pi, \sigma) > c$.

So if **appDeciderLine**(c, ε') returns "false" then $d_{\mathcal{G}}(\pi, \sigma) > c$, and if **appDeciderLine**$(c/(1+2\varepsilon'), \varepsilon')$ returns "true", then $d_{\mathcal{G}}(\pi, \sigma) \le \frac{c}{(1+2\varepsilon')}(1+\varepsilon') < c$. Otherwise $d_{\mathcal{G}}(\pi, \sigma) \in (\frac{c}{(1+2\varepsilon')}, c(1+\varepsilon')] = \frac{c}{1+2\varepsilon'}(1, (1+\varepsilon')(1+2\varepsilon')] \subset \frac{c}{1+2\varepsilon'}(1, 1+5\varepsilon') = \frac{c}{1+2\varepsilon/5}(1, 1+\varepsilon)$ where $\varepsilon' = \varepsilon/5 < 1$. ◀

▶ **Lemma 19.** *Given an interval $[\alpha, \beta]$, with $\alpha > 0$, one can either report "$d_{\mathcal{G}}(\pi, \sigma) \notin [\alpha, \beta]$" in $O(n^2/\varepsilon)$ time, or obtain $(1+\varepsilon)$ approximation to $d_{\mathcal{G}}(\pi, \sigma)$ in $O(\frac{n^2}{\varepsilon} \log \frac{\beta}{\alpha\varepsilon})$ time, which simplifies to $O(\frac{n^2}{\varepsilon} \log \frac{1}{\varepsilon})$ time when $\beta = O(\alpha)$.*

Proof. By using the Lemma 18, one can decide whether $d_{\mathcal{G}}(\pi, \sigma) < \alpha$, $d_{\mathcal{G}}(\pi, \sigma) > \beta$, or obtain $(1+\varepsilon)$ approximation to $d_{\mathcal{G}}(\pi, \sigma)$ in $O(n^2/\varepsilon)$ time.

If $d_{\mathcal{G}}(\pi, \sigma) \in [\alpha, \beta]$, divide the interval $[\alpha, \beta]$ into sub intervals with equal step distance $\alpha\varepsilon$ and perform a binary search over these sub intervals. When we try the sub interval $[\gamma, \gamma + \alpha\epsilon]$, by using Lemma 18, one can decide whether $d_{\mathcal{G}}(\pi, \sigma) < \gamma$, $d_{\mathcal{G}}(\pi, \sigma) > \gamma + \alpha\epsilon$. If $d_{\mathcal{G}}(\pi, \sigma) < \gamma$, continue binary search on the median sub interval between α and γ. Else if $d_{\mathcal{G}}(\pi, \sigma) > \gamma + \alpha\epsilon$, then continue binary search on the median sub interval between $\gamma + \alpha\epsilon$ and β, otherwise $d_{\mathcal{G}}(\pi, \sigma) \in [\gamma, \gamma + \alpha\varepsilon] \subset [\gamma, \gamma(1+\epsilon)]$.

We searched over $O(\frac{\beta}{\alpha\varepsilon})$ values, requiring $O(\log \frac{\beta}{\alpha\varepsilon})$ calls to **appDeciderLine**$(c, \varepsilon/5)$, hence our procedure takes $O(\frac{n^2}{\varepsilon} \log \frac{\beta}{\alpha\varepsilon})$ time overall as claimed. ◀

▶ **Corollary 20.** *One can $(1+\varepsilon)$-approximate $d_{\mathcal{G}}(\pi, \sigma)$ in $O(n^3 + n^2 \log(n) \frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ time.*

Proof. By Lemma 15 there is some $D \in \mathcal{D}$ such that $d_{\mathcal{G}}(\pi, \sigma) \in [D, 2D]$. Thus if for each $D \in \mathcal{D}$ we call the procedure of Lemma 19 to search the interval $[D, 2D]$, then we are guaranteed find a $(1+\varepsilon)$-approximation. Note that we may query the value $D = 0$, which does not satisfy the conditions of Lemma 19, though we can easily check if $d_{\mathcal{G}}(\pi, \sigma) = 0$, as then $s = d_o = t$. Each interval we query takes $O(\frac{n^2}{\varepsilon} \log \frac{1}{\varepsilon})$ time. Thus by using standard linear time median selection over the $O(n^3)$ values in $\mathcal{D}$, the running time follows. ◀

6.3 Improving the running time

Here we show how to use sampling to improve the running time of Corollary 20 by nearly a linear factor. Specifically, our goal is to find the value $D \in \mathcal{D}$ for which by Lemma 15 $[d_o - D, d_o + D]$ is a 2-approximation to an optimum gap interval.

The approach is similar to that in Section 5 (and even closer to the algorithm in [19]). The main difference is that the description of the functions used in the sweeping procedure is more involved, and so we describe this subroutine first before describing the full algorithm.

6.3.1 Sweeping

Given values $\alpha \le \beta$, we seek a procedure **sweep**(α, β) which returns a superset of all values $D_\imath \in \mathcal{D}$ such that $D_\imath \in [\alpha, \beta]$. The $\mathcal{P}_\imath$ regions differ based on which constraint type from Section 4.1.2 they correspond to. In particular, $\mathcal{D} = \mathcal{D}_{conn} \cup \mathcal{D}_{mono}$, where $\mathcal{D}_{mono}$ is the set of $D_\imath \in \mathcal{D}$ corresponding to regions which represent monotonicity events (either standard or floating), and $\mathcal{D}_{conn}$ is the set corresponding to all connectivity events (plus the initialization events). The set $\mathcal{D}_{conn}$ has size $O(n^2)$, and thus all values from this set in $[\alpha, \beta]$ can be found by brute force in quadratic time, so from now on we only consider the set $\mathcal{D}_{mono}$.

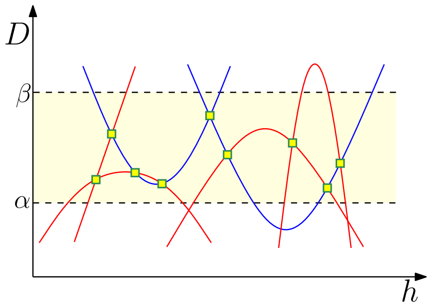


Figure 6.2 Possible $d_o - s$ functions (in red) and $t - d_o$ functions (in blue). The horizontal axis is the height h and the vertical axis is the radius D, i.e. distance from d_o.

Consider all $D_\imath \in \mathcal{D}_{mono}$ corresponding to a fixed row, i.e. a fixed edge $\pi_{i-1}\pi_i$, of the free space. As discussed in Section 4.1.1, for any column indices $j \leq k$, a standard monotonicity event occurs at a value of t such that $height(\mathcal{I}_j^b) = height(\mathcal{I}_k^a)$ or a value of s such that $height(\mathcal{H}_j^a) = height(\mathcal{H}_k^b)$, and a floating monotonicity event occurs at pairs s, t such that $height(\mathcal{I}_j^b) = height(\mathcal{H}_k^b)$ or $height(\mathcal{H}_j^a) = height(\mathcal{I}_k^a)$. From that section we also know the functions for s and t in terms of these heights. Specifically, $height(\mathcal{I}_k^a)$ is determined by the distance from σ_k, and so the function for t in terms of the height $h = height(\mathcal{I}_k^a)$ is given by $t = \sqrt{\chi^2 + (\gamma - h)^2}$, where χ, γ are the coordinates of vertex σ_k (in the $\pi_{i-1}, \pi_i, \sigma_k$ plane). Note this is also the function for t in terms of $h = height(\mathcal{I}_k^b)$ (i.e. the function is symmetric with one side representing $\mathcal{I}_k^a$ and the other $\mathcal{I}_k^b$). $height(\mathcal{H}_j^a)$ on the other hand is determined by the distance from the edge $\sigma_{j-1}\sigma_j$, and so the function for s in terms of the height $h = height(\mathcal{H}_j^a)$ has three cases, based on whether it is a distance to one of two edge endpoints or an interior point. For the endpoint cases again we have functions of the form $s = \sqrt{\alpha^2 + (\beta - h)^2}$, for some constants α, β, and just as before these functions also describe the endpoint cases for $height(\mathcal{H}_j^b)$. For the interior case the function is of the form $s = c \cdot h + d$ for both $height(\mathcal{H}_j^a)$ and $height(\mathcal{H}_j^b)$, although the constants in the linear functions for $height(\mathcal{H}_j^a)$ and $height(\mathcal{H}_j^b)$ can differ. (Slopes may have opposite sign when $dist(\pi_{i-1}\pi_i, \sigma_{j-1}\sigma_j)$ is realized at the interior of both edges, a case not shown in Figure 4.2.)

Fix a row of the free space, and for all pairs of column indices $j \leq k$ plot the above described function for t and all the functions for s (endpoints and interior functions). Specifically, in the plot of these functions the horizontal axis is h and the vertical axis is both s and t. Consider a standard monotonicity event. This happens at a value of t (resp. s) such that $h = height(\mathcal{I}_j^b) = height(\mathcal{I}_k^a)$ (resp. $h = height(\mathcal{H}_j^a) = height(\mathcal{H}_k^b)$), i.e. at an intersection point of two of the plotted t (resp. s) functions. Floating monotonicity events as always are a bit trickier (actually the easier standard monotonicity case was already described in [19], though in a different way). Specifically, while any floating event still occurs at a single h value, such events in general do not occur at intersections of the s and t functions since s and t may differ in value. The key observation however is that now we are only concerned with projected points, i.e. points such that $d_o - s = t - d_o$. So instead plot all functions of the form $d_o - s$, $t - d_o$ where s and t are any of the functions described above. In this new plot the horizontal coordinate is again h but the vertical coordinate is now the radius D (i.e. distance from d_o), and projected points now occur at the intersections of these transformed s and t functions, see Figure 6.2. (Note the standard events still occur at intersections.)

▶ **Definition 21.** For a fixed row i of the free space, consider the arrangement of all functions of the form $d_o - s$ and $t - d_o$ in the 2-dimensional D, h space, where s and t are given by the above described functions of h for any pair of columns $j \leq k$. Define $\mathcal{Z}_i^r$ to be the set of radius D values of all intersection points in this arrangement for row i, and similarly define $\mathcal{Z}_i^c$ for column i. The set of all intersection radii over all rows and columns is denoted $\mathcal{Z}$.

It would appear from the above discussion that $\mathcal{Z}$ is a superset of $\mathcal{D}_{mono}$, though there is one subtlety. Consider the line in the s, t parametric space defined by the equation $d_o - s = t - d_o$, i.e. the line defining all projected points. For a standard monotonicity event the boundary of $\mathcal{P}_\imath$ is a single horizontal (or vertical) line, which intersects $d_o - s = t - d_o$ and thus the minimum projected point in the region lies on this line boundary[4]. On the other hand, if $\mathcal{P}_\imath$ is from a floating monotonicity event then the boundary is more complicated, as show in Figure 4.3. In this figure, only the blue curve relates to an intersection point from the D, h space, and thus we also must consider when the minimum projected point is not on this curve, i.e. when $d_o - s = t - d_o$ intersects a different part of the boundary. There are three cases. The first two case are when the minimum projected point lies on either the horizontal line at t_1 or the vertical line at s_2, however, as mentioned in Section 4.1.1 these boundaries correspond to connectivity events and hence are already covered by in the set $\mathcal{D}_{conn}$. The third case is when the minimum point lies on the vertical line at $s^l = s_1$ (i.e. the left part of Figure 4.3). Recall $s_1 = dist(\sigma_{j-1}\sigma_j, \pi_{i-1}\pi_i)$, that is even though there are $O(n^2)$ functions $f_{i,j,k}$ in the ith free space row, there are only $O(n)$ distinct s_1 values, indexed by edges of σ. Thus we find all intersections of the line $d_o - s = t - d_o$ with the lines $s = s_1$ for all possible s_1. Let $\mathcal{B}$ be the corresponding radii of all such intersections over all rows, and note that $|\mathcal{B}| = O(n^2)$. Thus from the above discussion, $\mathcal{D}_{conn} \cup \mathcal{Z} \cup \mathcal{B}$ is a superset of $\mathcal{D}$ and so we have the following.

▶ **Lemma 22.** *For any values $\alpha \leq \beta$, there is an algorithm* **sweep**(α, β) *which in $O((n^2 + |W|) \log n)$ time outputs a set $W \supseteq (\mathcal{D} \cap [\alpha, \beta])$, such that $|W| = O(\max\{n^2, |\mathcal{Z} \cap [\alpha, \beta]|\})$, where $\mathcal{Z}$ is as defined in Definition 21.*

Proof. Above it was argued that $\mathcal{D} \subseteq (\mathcal{D}_{conn} \cup \mathcal{Z} \cup \mathcal{B})$. Thus for $W = \mathcal{D}_{conn} \cup (\mathcal{Z} \cap [\alpha, \beta]) \cup \mathcal{B}$, it holds that $W \supseteq (\mathcal{D} \cap [\alpha, \beta])$. As mentioned above, $|\mathcal{D}_{conn}|, |\mathcal{B}| = O(n^2)$, thus $|W| = O(\max\{n^2, |\mathcal{Z} \cap [\alpha, \beta]|\})$. Moreover, $\mathcal{D}_{conn}$ and $\mathcal{B}$ are easily computable in $O(n^2)$ time.

What remains is how to compute $\mathcal{Z} \cap [\alpha, \beta]$, which is done by using standard sweeping. So for some row i consider the set $\mathcal{Z}_i^r$. As discussed above this is the set of D values of the intersection points of a set of n constant complexity monotone functions in the D, h space. Thus $\mathcal{Z}_i^r \cap [\alpha, \beta]$ is determined by the set of all such intersections in a given horizontal strip $[\alpha, \beta]$, which just as in Lemma 10, can be found using a standard horizontal sweep line [8], which given a set of n monotone constant-complexity curves, finds all intersections in the interval $[\alpha, \beta]$ in $O((n + |\mathcal{Z}_i^r \cap [\alpha, \beta]|) \log n)$ time. Thus doing this sweeping for every row and column finds the set $\mathcal{Z} \cap [\alpha, \beta]$ in $O((n^2 + |\mathcal{Z} \cap [\alpha, \beta]|) \log n)$ time. ◀

6.3.2 The Algorithm

The full algorithm is shown in Algorithm 2, which the reader may observe has the same high level structure as Algorithm 1.

The only yet unspecified step of this algorithm is the first line where we call the subroutine **sampler**, which samples values from the set $\mathcal{Z}$ of Definition 21. Any value in $\mathcal{Z}$ is determined by an intersection point of a $d_o - s$ and a $t - d_o$ function, which in turn are specified by:

[4] Technically, here we ignore irrelevant, but possible, constraints where the horizontal line lies below d_o.

(i) a triple of indices i, j, k, (ii) whether it is a row or column of the free space, and (iii) which of the $O(1)$ types of intersection points it is (i.e. whether it is standard or floating, if standard whether it is a t or s function intersection, also which type of the three possible s functions). Note alternatively, at the cost of increasing the sample size by a constant factor, after sampling i, j, k, one can include all $O(1)$ intersections this triple defines. Thus a value in $\mathcal{Z}$ can be sampled in constant time by sampling these pieces of information and thus **sampler**(γn^2) runs in $O(n^2)$ time.

```
R = sampler(γn²)                                   // γ a sufficiently large constant
Sort I = {[d_o − D, d_o + D] | D ∈ R} by the values in R
Binary search over I, using deciderPoint, for interval [D_α, D_β] such that
  deciderPoint(d_o − D_α, d_o + D_α) false and deciderPoint(d_o − D_β, d_o + D_β) true
                                          // Set initial values D_α = 0, D_β = ∞
S = sweep(D_α, D_β)
Sort I′ = {[d_o − D, d_o + D] | D ∈ S} by the values in S
Binary search over I′ using deciderPoint and return the smallest value D̂ ∈ S
  such that [d_o − D̂, d_o + D̂] is feasible.
```

Algorithm 2: Sampling to compute a fast 2-approximation.

Let $[D_\alpha, D_\beta]$ be as defined in Algorithm 2. The number of values from $\mathcal{Z}$ which fall in this interval affects the running time **sweep** (see Lemma 22). We now prove that with high probability this set has quadratic size. The proof is nearly identical to Lemma 11 (and a proof in [19]), and is included for the sake of completeness.

▶ **Lemma 23.** *Let $[D_\alpha, D_\beta]$ be as defined in Algorithm 2. Then with exponentially high probability, **sweep**(D_α, D_β) returns a set of size $O(n^2)$.*

Proof. **sampler**(γn^2) takes a γn^2 sized sample $\mathcal{R}$ from the set $\mathcal{Z}$, which is a set of values of size ζn^3 for some constant ζ. Consider the sorted placement of the values in $\mathcal{Z}$ along the real line, and let x be any point on the real line. Let U^+ (resp. U^-) be the closest n^2 values from $\mathcal{Z}$ larger (resp. smaller) than x. Suppose $|U^+| = n^2$ (note it may happen that $|U^+| < n^2$, if x is large enough). Then the probability it does not contain a value in $\mathcal{R}$ is

$$\left(1 - \frac{|U^+|}{\zeta n^3}\right)^{\gamma n^2} = \left(1 - \frac{n^2}{\zeta n^3}\right)^{\gamma n^2} \leq \exp\left(\frac{-\gamma n^4}{\zeta n^3}\right) = \exp(-(\gamma/\zeta)n) \leq e^{-\hat{c}n}$$

for any constant $\hat{c}$, by choosing γ to be sufficiently large. A similar statement holds for $|U^-|$, and taking the union bound, a similar statement holds simultaneously for both U^- and U^+.

Note there are only $O(n^3)$ possibilities for each value in $\mathcal{R}$. Thus by setting x to each one of these values, and taking the union bound, it holds that between any two adjacent values in (the sorted set) $\mathcal{R}$, or in the unbounded end intervals, with high probability there are at most $O(n^2)$ values of $\mathcal{Z}$. Thus the lemma statement holds, as **sweep**(D_α, D_β) returns such a set of $\mathcal{Z}$ values, together with all of $\mathcal{D}_{corr}$ and $\mathcal{B}$ which are each of size $O(n^2)$. ◀

▶ **Theorem 24.** *Given polygonal curves π and σ, each of length at most n, one can $(1+\varepsilon)$-approximate the Fréchet gap distance in $O(n^2(\log n + \frac{1}{\varepsilon}\log\frac{1}{\varepsilon}))$ time.*

Proof. As discussed above, we only require Algorithm 2 to return a 2-approximation, as such an approximation can then be transformed into $(1+\varepsilon)$-approximation in additive $O(\frac{n^2}{\varepsilon}\log\frac{1}{\varepsilon})$ time, using Lemma 19. Now by Lemma 15, there is some radius $D' \in \mathcal{D}$ whose corresponding projected point gives a 2-approximation. Recall from Observation 14, that sorting projected points by increasing radius D, induces a linear ordering of feasibility. Thus for the interval $[D_\alpha, D_\beta]$ found by the algorithm, it must be that $D' \geq D_\alpha$ since D_α is infeasible (or zero). Now if $D' \leq D_\beta$ then $D' \in [D_\alpha, D_\beta]$, and since **sweep** finds a superset of all $\mathcal{D}$ in $[D_\alpha, D_\beta]$, when we search over the returned values we must find at least a 2-approximation. Otherwise $D' > D_\beta$, and hence the feasible D_β or something even better will be found and returned from the sweeping, again getting at least a 2-approximation.

As for the running time, sampling $O(n^2)$ values and sorting takes $O(n^2 \log n)$ time. Binary searching with **deciderPoint** also takes $O(n^2 \log n)$ time as **deciderPoint** takes $O(n^2)$ time. By Lemma 23, with exponentially high probability the number of values from $\mathcal{Z}$ in the found interval is $O(n^2)$, and thus by Lemma 22 **sweep** takes $O(n^2 \log n)$ time with exponentially high probability. Finally we do another round of sorting and searching over the $O(n^2)$ values returned from **sweep**, which again takes $O(n^2 \log n)$ time. Adding the time to convert the result to a $(1+\varepsilon)$-approximation gives $O(n^2(\log n + \frac{1}{\varepsilon} \log \frac{1}{\varepsilon}))$ time overall. ◀

References

1. P. Agarwal, R. Avraham, H. Kaplan, and M. Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014.
2. H. Alt and M. Buchin. Can we compute the similarity between surfaces? *Discrete & Computational Geometry*, 43(1):78–99, 2010.
3. H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. In *Proc. of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 589–598, 2003.
4. H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.
5. H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Annu. Symp. on Theo. Aspects of Comp. Sci. (STACS)*, pages 63–74, 2001.
6. R. Avraham, O. Filtser, H. Kaplan, M. Katz, and M. Sharir. The discrete and semicontinuous Fréchet distance with shortcuts via approximate distance counting and selection. *ACM Trans. Algorithms*, 11(4):29, 2015.
7. R. Avraham, H. Kaplan, and M. Sharir. A faster algorithm for the discrete Fréchet distance under translation. *CoRR*, abs/1501.03724, 2015.
8. J. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Computers*, 28(9):643–647, 1979.
9. S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proc. 31st VLDB Conference*, pages 853–864, 2005.
10. K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless seth fails. In *Symp. on Found. of Comp. Sci. (FOCS)*, pages 661–670. IEEE, 2014.
11. K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. *Int. J. Comput. Geom. Appl.*, 21(3):253–282, 2011.
12. K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog - with an application to alt's conjecture. In *Proc. of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1399–1413, 2014.
13. K. Buchin, M. Buchin, and C. Wenk. Computing the Fréchet distance between simple polygons in polynomial time. In *22nd Annu. Sympos. Comput. Geom.*, pages 80–87, 2006.
14. M. Buchin, A. Driemel, and B. Speckmann. Computing the Fréchet distance with shortcuts is np-hard. In *30th Annu. Sympos. Comput. Geom. (SoCG)*, page 367, 2014.
15. A. Driemel and S. Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

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

17 T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical report, 1994.

18 O. Filtser and M. Katz. The discrete Fréchet distance gap. *arXiv:1506.04861*, 2015.

19 S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms (TALG)*, 10(1):3, 2014.

20 M. Kim, S. Kim, and M. Shin. Optimization of subsequence matching under time warping in time-series databases. In *Proc. ACM Symp. on Applied Computing*, pages 581–586, 2005.

21 G. Rote. Computing the Fréchet distance between piecewise smooth curves. *Computational Geometry*, 37(3):162–174, 2007.

22 J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identifica. *Audio, Speech & Lang. Proc.*, 16(6):1138–1151, 2008.

23 C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing global curve-matching algorithms. In *Sci. Statis. Database Manag.*, pages 879–888, 2006.