

# A Near-Linear Time Exact Algorithm for the $L_1$ -Geodesic Fréchet Distance Between Two Curves on the Boundary of a Simple Polygon

Thijs van der Horst 

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

Marc van Kreveld 

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

Tim Ophelders 

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

Bettina Speckmann 

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

---

## Abstract

Let  $P$  be a polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple, interior disjoint curves on the boundary of  $P$ , with  $n$  and  $m$  vertices. We show how to compute the Fréchet distance between  $R$  and  $B$  using the geodesic  $L_1$ -distance in  $P$  in  $\mathcal{O}(k \log nm + (n + m)(\log^2 nm \log k + \log^4 nm))$  time.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Computational geometry

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

**Digital Object Identifier** 10.4230/LIPIcs.WADS.2025.37

**Related Version** *Full Version*: <https://arxiv.org/abs/2401.14815>

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

## 1 Introduction

In a seminal paper from 1992, Alt and Godau [1] introduced the *Fréchet distance* to the algorithms community as a better similarity measure for two curves than, for example, the Hausdorff distance. For two polygonal curves with  $n$  and  $m$  vertices, they addressed the algorithmic problem of computing the Fréchet distance and presented an  $\mathcal{O}(nm \log nm)$  time algorithm [2], which is near-quadratic in the input size.

Since then, research efforts have developed globally in two directions: examining variations or generalizations that can still be solved in near-quadratic time, and restrictions or approximations to beat the quadratic time upper bound. In 2014, Bringmann [3] showed that the Fréchet distance cannot be computed to within a factor 1.001 in strongly subquadratic time, unless the strong exponential time hypothesis fails. This result was improved in 2019 by Buchin, Ophelders and Speckmann [5], who showed that this conditional lower bound applies even for curves in 1-dimensional space and for approximations better than a factor 3. On the other hand, Driemel, Har-Peled and Wenk [7] showed that a near-linear time  $(1 + \varepsilon)$ -approximation exists for curves in the plane that satisfy a sparseness condition. A full overview of the results is beyond the scope of this paper.

One of the variations that has been examined concerns computing the Fréchet distance between two curves in a polygonal environment. Intuitively, the Fréchet distance is based on matching points on the two curves in a continuous and forward manner, and considering



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

19th International Symposium on Algorithms and Data Structures (WADS 2025).

Editors: Pat Morin and Eunjin Oh; Article No. 37; pp. 37:1–37:13

Leibniz International Proceedings in Informatics



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

the Euclidean distance between matched points. In a polygonal environment, the Euclidean distance is replaced by the Euclidean *geodesic* distance, the length of a shortest path in the environment. In the setting of two curves inside a simple polygon, Cook and Wenk [9] showed that the Fréchet distance can still be computed in near-quadratic time. Recently, it was shown that if the two curves lie on the boundary of a simple polygon, then a  $(1 + \varepsilon)$ -approximation exists that uses near-linear time [11]. In this paper we show that in the same setting, but using the  $L_1$ -distance to measure lengths of geodesics rather than the Euclidean distance, an exact algorithm can be given that requires near-linear time.

**Preliminaries.** Let  $P$  be a simple polygon. A *curve* on the boundary of  $P$  is a piecewise-linear function  $F: [0, 1] \rightarrow \partial P$  connecting a sequence of *vertices*. Consecutive vertices are connected by a straight-line *edge*. The curve is *simple* if the preimage of a point on  $\partial P$  is at most an interval. We denote by  $F[x, x']$  the subcurve of  $F$  over the domain  $[x, x']$ . We write  $|F|$  to denote the number of vertices of  $F$ .

A *reparameterization* of  $[0, 1]$  is a non-decreasing surjection  $f: [0, 1] \rightarrow [0, 1]$ . Two reparameterizations  $f$  and  $g$  describe a *matching*  $(f, g)$  between two curves  $R$  (red) and  $B$  (blue), where for every  $t \in [0, 1]$  the point  $R(f(t))$  is matched to  $B(g(t))$ . The matching  $(f, g)$  is said to have *cost*

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

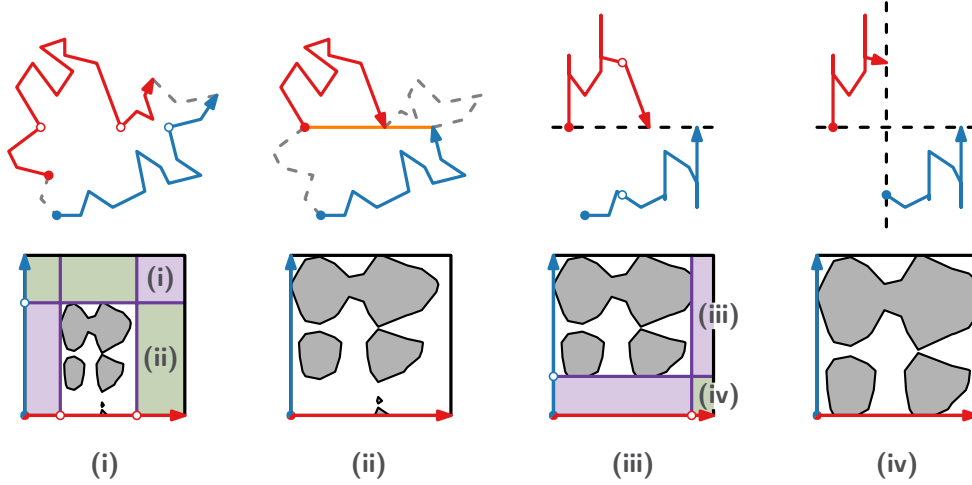
where in this work,  $\bar{d}(p, q)$  measures the length of a shortest path in  $P$  from  $p$  to  $q$  under the  $L_1$ -norm. We call such a path an  $L_1$ -*geodesic*. There are potentially uncountably many  $L_1$ -geodesics from  $p$  to  $q$ . On the other hand, under the  $L_2$ -norm,  $P$  contains a unique shortest path from  $p$  to  $q$ , which we denote by  $\pi(p, q)$ . It turns out that  $\pi(p, q)$  is also an  $L_1$ -geodesic, so we use it as a representative.

A matching with cost at most  $\delta$  is called a  $\delta$ -*matching*. The (continuous)  $L_1$ -*geodesic Fréchet distance*  $\bar{d}_F(R, B)$  between  $R$  and  $B$  is the minimum cost over all matchings. The corresponding matching is a *Fréchet matching*.

The *parameter space* of  $R$  and  $B$  is the axis-aligned rectangle  $[0, 1]^2$ . Any point  $(x, y)$  in the parameter space corresponds to the pair of points  $R(x)$  and  $B(y)$  on the two curves. A point  $(x, y)$  in the parameter space is  $\delta$ -*close* for some  $\delta \geq 0$  if  $\bar{d}(R(x), B(y)) \leq \delta$ . The  $\delta$ -*free space*  $\mathcal{F}_\delta(R, B)$  of  $R$  and  $B$  is the subset of  $[0, 1]^2$  containing all  $\delta$ -close points. A point  $q = (x', y') \in \mathcal{F}_\delta(R, B)$  is  $\delta$ -*reachable* from a point  $p = (x, y)$  if  $x \leq x'$  and  $y \leq y'$ , and there exists a bimonotone path in  $\mathcal{F}_\delta(R, B)$  from  $p$  to  $q$ . Alt and Godau [2] observe that there is a one-to-one correspondence between  $\delta$ -matchings between  $R[x, x']$  and  $B[y, y']$ , and bimonotone paths from  $p$  to  $q$  through  $\mathcal{F}_\delta(R, B)$ . We abuse terminology slightly and refer to such paths as  $\delta$ -matchings.

**Problem statement and results.** Our input consists of a simple polygon  $P$  with  $k$  vertices and curves  $R, B: [0, 1] \rightarrow \partial P$  with  $n$  and  $m$  vertices respectively, on the boundary of  $P$ . We restrict ourselves to inputs of the following type:

(i)  $R$  and  $B$  are oppositely oriented, interior-disjoint, simple curves on the boundary of  $P$ . We present an algorithm for computing the Fréchet distance  $\bar{d}_F(R, B)$  between  $R$  and  $B$ , when distances between points are measured as the minimum length of a path in  $P$ , under the  $L_1$ -norm. For our algorithm, we make two assumptions. Firstly, without loss of generality, assume that  $R$  is oriented clockwise with respect to  $P$ , and  $B$  counter-clockwise. Secondly, we assume general position: no edge of  $P$  has slope  $\pm 1$ , and no two vertices of  $P$  have the same  $y$ -coordinate. We compute the Fréchet distance between  $R$  and  $B$  under the geodesic  $L_1$ -distance in  $P$  in  $\mathcal{O}(k \log nm + (n + m)(\log^2 nm \log k + \log^4 nm))$  time.



■ **Figure 1** The four types of subproblems that arise in our divide-and-conquer scheme. The bottom row shows the parameter spaces corresponding to the top row. Between problem types, the correspondence between some subcurves and parts of the  $\delta$ -free space is illustrated.

**Algorithmic outline.** We first outline the *decision algorithm*: For an additional parameter  $\delta \geq 0$ , it reports whether  $\bar{d}_F(R, B) \leq \delta$ . Recall that a  $\delta$ -matching between  $R$  and  $B$  corresponds to a bimonotone path in  $\mathcal{F}_\delta(R, B)$  from  $(0, 0)$  to  $(1, 1)$ . We use divide-and-conquer to decide whether such a path exists.

We first explain how to divide the problem of type (i) into simpler subproblems of four types (i)–(iv), illustrated in Figure 1. In Section 3.3 we find a horizontal chord  $e^*$  of  $P$  that intersects both  $R$  and  $B$  in at most a constant number of points. We split  $R$  and  $B$  at these points, resulting in subcurves  $R_i$  of  $R$  and  $B_j$  of  $B$ . Each pair  $(R_i, B_j)$  corresponds to a simpler subproblem. Such a subproblem is either of type (i) where  $R_i$  and  $B_j$  have at most a constant fraction of the vertices of  $R$  and  $B$ , or of a new type with extra structure:

(ii)  $R_i$  and  $B_j$  are separated by a horizontal chord of  $P$ , namely  $e^*$ .

We handle subproblems of type (i) recursively.

We handle subproblems of type (ii) in Section 3.2. Suppose that  $R$  and  $B$  are separated by a horizontal chord  $e^*$  of  $P$ , such as the subproblems of type (ii). For a point  $p \in P$ , denote by  $NN(p)$  the point on  $e^*$  closest to  $p$  under the  $\bar{d}$  metric. We leverage the property that for all  $r \in R$  and  $b \in B$  there exists an  $L_1$ -geodesic from  $r$  to  $b$  that goes via  $NN(r)$  and  $NN(b)$ . Using this property, we transform the problem  $(R, B)$  of type (ii) into an equivalent problem of a new type that is no longer constrained by the polygon:

(iii) a pair of  $x$ -monotone<sup>1</sup> curves  $(\bar{R}, \bar{B})$  that are separated by a horizontal line.

We handle subproblems of type (iii) in Section 3.1. Suppose that  $R$  and  $B$  are  $x$ -monotone curves separated by a horizontal line, such as the subproblems of type (iii). We again split the problem into simpler subproblems. We find a vertical line  $\ell$  that has a constant fraction of the vertices of  $R \cup B$  on either side. We split  $R$  and  $B$  at points on  $\ell$ , resulting in subcurves  $R_1$  and  $R_2$  of  $R$ , and  $B_1$  and  $B_2$  of  $B$ . Each pair  $(R_i, B_j)$  corresponds to a simpler subproblem. Such a subproblem is either of type (iii) and handled recursively, or of a new type:

(iv)  $R_i$  and  $B_j$  are  $x$ -monotone curves separated by both a horizontal and a vertical line.

<sup>1</sup> Throughout this work, we consider monotonicity in the weak sense. That is, a curve is  $x$ -monotone if every vertical line intersects the curve in at most one subcurve.

The above scheme splits our original problem into several subproblems of type **(iv)**, each of which corresponds to an axis-aligned rectangle in the parameter space of the two input curves. Together, these rectangles partition the parameter space into interior-disjoint regions. Recall that we want to decide whether there exists a bimonotone path from  $(0, 0)$  to  $(1, 1)$  in the parameter space that lies completely in the  $\delta$ -free space. To do so, we maintain a set of points in the  $\delta$ -free space that can be reached by a bimonotone path that starts at  $(0, 0)$ . If for a subproblem of type **(iv)** we know for its corresponding rectangle a set of reachable “entry” points on the bottom and left sides, then we can compute a sufficient set of “exit” points on the top and right sides that are reachable via at least one entry point. See Section 2 for details. By processing the subproblems of type **(iv)** in the correct order, we compute whether a bimonotone path from  $(0, 0)$  to  $(1, 1)$  through the  $\delta$ -free space exists.

In Section 4.2, we use our decision algorithm to compute  $\bar{d}_F(R, B)$  exactly, by binary searching over a set of  $\tilde{O}((n + m)^2)$  candidate values and querying the decision algorithm at every step. Computing all candidate values explicitly would take too much time. Instead, we represent them implicitly as the Cartesian sum of two sorted sets  $X$  and  $Y$  of only  $O((n + m) \log^2 nm)$  values. Using algorithms for linear-time selection in such implicitly represented Cartesian sums [8, 10], the time taken to compute a candidate for  $\delta$  is dominated by the time taken to perform the decision algorithm. This allows us to compute  $\bar{d}_F(R, B)$  with only logarithmic overhead compared to the decision algorithm.

We present our decision algorithm bottom-up, starting with the most specific problem **(iv)** that arises, and ending with the general problem **(i)**, in the following sections.

## 2 Doubly-separated $x$ -monotone curves

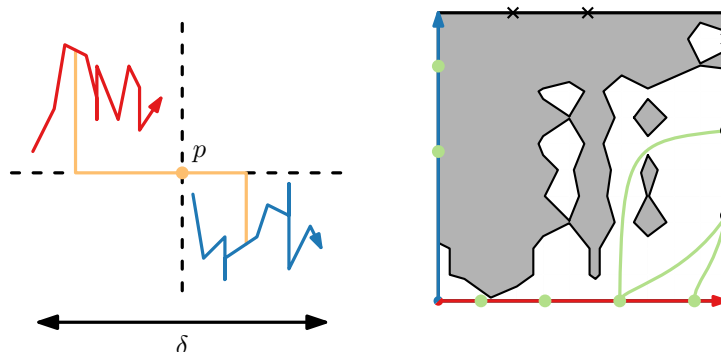
In this section, we consider subproblems of type **(iv)**, where  $R$  and  $B$  are two  $x$ -monotone curves in  $\mathbb{R}^2$  separated by both a horizontal line and a vertical line. Let  $R$  have  $n$  vertices and  $B$  have  $m$  vertices. We present an algorithm for *propagating reachability information* through the  $\delta$ -free space of  $R$  and  $B$ . We are given a parameter  $\delta \geq 0$ , a set of  $O(n + m)$  points  $S$  on the bottom and left sides of the parameter space, and a set of  $O(n + m)$  points  $T$  on the top and right sides of the parameter space. The goal is to report the points in  $T$  that are  $\delta$ -reachable from at least one point in  $S$ . See Figure 2. In this section, we measure distances under the  $L_1$ -norm  $\|\cdot\|_1$ , in which shortest paths are not restricted by any polygon.

Lemma 1 shows that the distances between  $R$  and  $B$  can be captured by two curves in one dimension that are separated by the origin, which can be constructed in linear time.

► **Lemma 1.** *Let  $R$  and  $B$  be two doubly-separated curves in  $\mathbb{R}^2$  with  $n$  and  $m$  vertices. In  $O(n + m)$  time, we can construct two curves  $\bar{R}$  and  $\bar{B}$  in  $\mathbb{R}$  that are separated by the origin, such that  $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$  for all  $\delta$ . The curves  $\bar{R}$  and  $\bar{B}$  have  $n$  and  $m$  vertices as well.*

Bringmann and Künnemann [4] and Van der Horst *et al.* [11] both give near-linear time algorithms for propagating reachability information through the free space of two separated curves in one dimension. Combining this with Lemma 1, we obtain:

► **Lemma 2.** *Let  $R$  and  $B$  be two doubly-separated  $x$ -monotone curves in  $\mathbb{R}^2$  with  $n$  and  $m$  vertices. Given a parameter  $\delta \geq 0$  and sets  $S$  and  $T$  of  $O(n + m)$  points on the bottom and left, respectively top and right, sides of the parameter space of  $R$  and  $B$ , we can report the points in  $T$  that are  $\delta$ -reachable from at least one point in  $S$  in  $O((n + m) \log nm)$  time.*



■ **Figure 2** (left) A pair of doubly-separated  $x$ -monotone curves. We may consider the curves as lying on the union of two (degenerate) “histogram” polygons, in which case  $\bar{d}$  is equal to the  $L_1$ -norm. Between any pair of red and blue points, there exists an  $L_1$ -geodesic through the point  $p$ . (right) Their  $\delta$ -free space diagram. For propagating reachability, given are a set of points  $S$  (disks) on the bottom and left sides of the parameter space, and a set  $T$  (circles and crosses) of points on the top and right sides. The filled circles are  $\delta$ -reachable, the crosses are not.

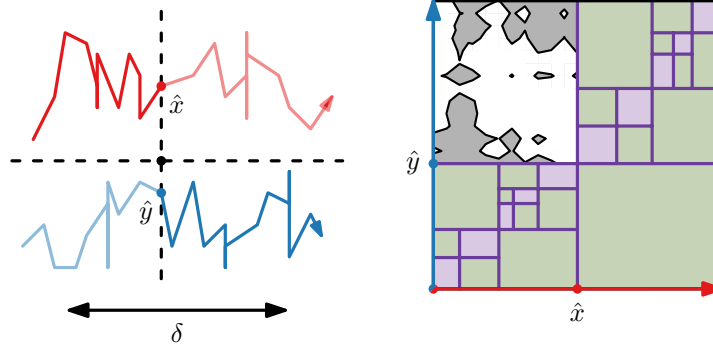
### 3 Partitioning the parameter space

In this section, we consider the main problem, of type (i), where  $R$  and  $B$  are simple, interior-disjoint curves on the boundary of a simple polygon  $P$ . We divide this problem into subproblems of type (iv) by partitioning the parameter space of  $R$  and  $B$  into axis-aligned rectangles where either the corresponding subcurves are trivial (i.e., line segments), or form a type (iv) instance, in that the subcurves are equivalent to a pair of doubly-separated  $x$ -monotone curves.

An axis-aligned rectangle  $[x, x'] \times [y, y']$  is *doubly-separated* if, for all  $\delta$ , the  $\delta$ -free space  $\mathcal{F}_\delta(R[x, x'], B[y, y'])$  inside it is equivalent to the  $\delta$ -free space of a pair of doubly-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$ , i.e.,  $\mathcal{F}_\delta(R[x, x'], B[y, y']) = \mathcal{F}_\delta(\bar{R}, \bar{B})$ . A *doubly-separated partition* of the parameter space of  $R$  and  $B$  is a partition into axis-aligned rectangles  $[x, x'] \times [y, y']$ , where either  $[x, x'] \times [y, y']$  is doubly-separated, or both  $R[x, x']$  and  $B[y, y']$  are line segments. In the latter case, we refer to such a region as *trivial*. A doubly-separated partition always exists, since we can take its regions to be only trivial regions, corresponding to the pairs of individual edges of the curves. Such a partition has high complexity however, having  $nm$  regions. Our goal is instead to compute a doubly-separated partition with few trivial regions, and where the doubly-separated regions correspond to subcurves with relatively few vertices in total, so that the total input complexity when applying the result of Lemma 2 is kept low.

To measure the “quality” of a doubly-separated partition, we define the following *costs*. We define the cost  $\text{Cost}(\mathcal{R})$  of a trivial region to be  $\mathcal{O}(1)$ , and the cost of a doubly-separated region  $\mathcal{R} = [x, x'] \times [y, y']$  as  $|R[x, x']| + |B[y, y']|$ , the total number of vertices in the two subcurves. We define the cost  $\text{Cost}(\mathcal{P})$  of a doubly-separated partition  $\mathcal{P}$  as  $\text{Cost}(\mathcal{P}) = \sum_{\mathcal{R} \in \mathcal{P}} \text{Cost}(\mathcal{R})$ . In this section we construct a doubly-separated partition of only near-linear cost, namely  $\mathcal{O}((n + m) \log^2 nm)$ .

We construct our partition in three phases. In Section 3.1 we construct a partition for subproblems of type (iii), where the curves  $R$  and  $B$  are  $x$ -monotone and separated by a horizontal line. Then, in Section 3.2, we handle subproblems of type (ii), where  $R$  and  $B$  are simple curves on the boundary of a simple polygon  $P$ , separated by a horizontal line segment in  $P$ . Finally, in Section 3.3, we handle subproblems of type (i).



■ **Figure 3** (left) A pair of  $x$ -monotone curves separated by a horizontal line, together with a vertical line with roughly half of all vertices on either side. (right) This line represents a partition of the parameter space into four regions, of which the top-left and bottom-right regions correspond to pairs of doubly-separated subcurves. The bottom-left and top-right regions are partitioned further. Green regions signify subproblems of type (iv).

### 3.1 Vertically-separated $x$ -monotone curves

In this section, we consider the setting where  $R$  and  $B$  are  $x$ -monotone curves in  $\mathbb{R}^2$ , separated by a horizontal line. We give a simple partition algorithm for the parameter space of  $R$  and  $B$  that yields a doubly-separated partition of cost  $\mathcal{O}((n+m)\log nm)$ :

► **Lemma 3.** *Let  $R$  and  $B$  be two vertically-separated  $x$ -monotone curves in  $\mathbb{R}^2$  with  $n$  and  $m$  vertices. We can compute a doubly-separated partition of the parameter space of  $R$  and  $B$  that has cost  $\mathcal{O}((n+m)\log nm)$  in  $\mathcal{O}((n+m)\log nm)$  time.*

### 3.2 Separated curves on a simple polygon

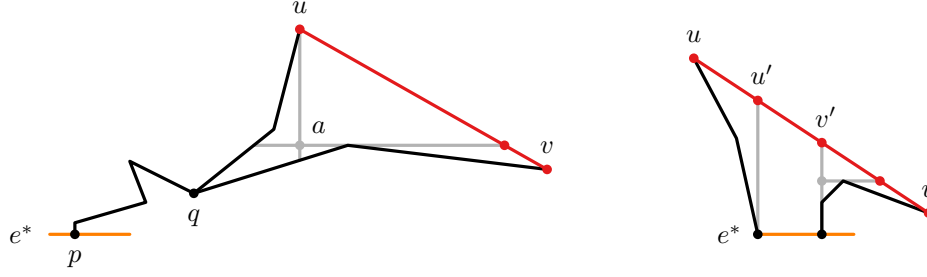
Next we reduce the following setting to that of two vertically-separated  $x$ -monotone curves: Let  $P$  be a simple polygon, and let  $R$  and  $B$  be two simple curves on the boundary of  $P$ , separated by a horizontal line segment in  $P$ . That is, there exists a horizontal line segment  $e^*$  in  $P$ , such that splitting  $P$  at  $e^*$  creates two subpolygons where  $R$  and  $B$  lie in different subpolygons.<sup>2</sup> Recall that we assume  $R$  to be oriented clockwise with respect to  $P$ , and  $B$  counter-clockwise. We construct a pair of vertically-separated  $x$ -monotone curves whose  $\delta$ -free space (under the  $L_1$ -norm) is identical to that of  $R$  and  $B$ , for all  $\delta$ .

Every  $L_1$ -geodesic between points  $r \in R$  and  $b \in B$  intersects  $e^*$  in exactly one line segment. Moreover, there exists an  $L_1$ -geodesic that is composed of the following three parts: an  $L_1$ -geodesic between  $r$  and its closest point  $u$  on  $e^*$ , an  $L_1$ -geodesic between  $b$  and its closest point  $v$  on  $e^*$ , and the line segment  $\overline{uv}$ . For two vertically-separated  $x$ -monotone curves (as in Section 3.1), there exists a similar shortest path between two points, namely one that is composed of two vertical segments and a horizontal segment on the separator. We use this connection for our reduction, which we specify next.

We refer to the “projection” of a point  $p \in P$  onto  $e^*$  (with slope other than  $\pm 1$ ) as the point on  $e^*$  closest to  $p$ , and we denote this projection by  $NN(p)$ . Above we established that between any two points  $r \in R$  and  $b \in B$ , there exists an  $L_1$ -geodesic that goes through the projections  $NN(r)$  and  $NN(b)$ . We introduce two real-valued functions, which we use to define the coordinates of the  $x$ -monotone curves we reduce the problem to.

<sup>2</sup> We consider  $e^*$  to be part of both subpolygons.





■ **Figure 4** An illustration of the two settings discussed in Lemma 5. On the left is the case where  $NN(u) = NN(v)$ . On the right is the case where  $NN(u) \neq NN(v)$ . Marked points on  $\overline{uv}$  are points where the functions  $\varphi$  and  $\psi$  change pieces.

We assume  $e^*$  is parameterized over  $[1, 2]$ . The first function,  $\varphi: P \rightarrow [1, 2]$ , indexes the projections on  $e^*$  of the points in  $P$ . That is,  $e^*(\varphi(p)) = NN(p)$ . The second function,  $\psi: P \rightarrow [1, 2]$ , represents the distance function from points to their projections, so  $\psi(p) = \bar{d}(p, NN(p))$ .

We consider the curves  $\bar{R}: [0, 1] \rightarrow \mathbb{R}^2$  and  $\bar{B}: [0, 1] \rightarrow \mathbb{R}^2$  given by  $\bar{R}(x) = (\varphi(R(x)), \psi(R(x)))$  and  $\bar{B}(y) = (\varphi(B(y)), -\psi(B(y)))$ . Note that  $\bar{d}(R(x), B(y)) = \|\bar{R}(x) - \bar{B}(y)\|_1$  for all  $x, y \in [0, 1]^2$ , and so  $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$  for all  $\delta$ . The curves are separated by the horizontal line  $(-\infty, \infty) \times \{0\}$ . Next we prove that the curves are  $x$ -monotone, making the algorithm developed in Section 3.1 applicable to them:

► **Lemma 4.** *The curves  $\bar{R}$  and  $\bar{B}$  are  $x$ -monotone.*

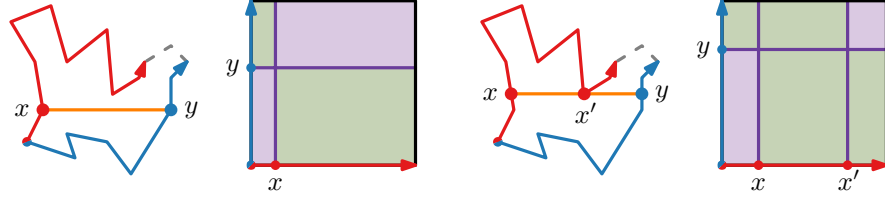
We have shown that the algorithm developed in Section 3.1 can be applied to the curves  $\bar{R}$  and  $\bar{B}$ . Next we show that  $\bar{R}$  and  $\bar{B}$  have low complexity. Let  $R$  have  $n$  vertices and  $B$  have  $m$  vertices. We show that  $\bar{R}$  has  $\mathcal{O}(n)$  vertices and  $\bar{B}$  has  $\mathcal{O}(m)$  vertices. While doing so, we develop a data structure that can construct  $\bar{R}$  and  $\bar{B}$  efficiently.

We construct pieces of  $\bar{R}$  and  $\bar{B}$  by computing the functions  $\varphi$  and  $\psi$  over individual edges of  $R$  and  $B$ . When restricted to a single line segment  $e$ , the function  $\psi$  is similar to the “hourglass function”  $H_{e,e^*}$  introduced by Cook and Wenk [9]. The hourglass function uses the  $L_2$ -norm to measure lengths of shortest paths, and can have  $\mathcal{O}(k)$  complexity [9]. Perhaps surprisingly however, we show that since we measure lengths with the  $L_1$ -norm, the function  $\psi$  has only constant complexity when applied to a single edge. Moreover, the function  $\varphi$  has constant complexity as well.

► **Lemma 5.** *Let  $\overline{uv} \subseteq P$  be a line segment interior-disjoint from  $e^*$ . The functions  $\varphi$  and  $\psi$ , restricted to  $\overline{uv}$ , are piecewise-linear with  $\mathcal{O}(1)$  pieces. After preprocessing  $P$  in  $\mathcal{O}(k)$  time, the functions can be computed over  $\overline{uv}$  in  $\mathcal{O}(\log k)$  time.*

By querying the data structure of Lemma 5 with every edge of  $R$  and  $B$  individually and combining the results, we obtain the functions  $\varphi$  and  $\psi$  over  $R$  and  $B$  in  $\mathcal{O}((n+m) \log k)$  time, after  $\mathcal{O}(k)$ -time preprocessing. From this, we extract  $\bar{R}$  and  $\bar{B}$  in  $\mathcal{O}(n+m)$  additional time. The complexities of  $\bar{R}$  and  $\bar{B}$  are  $\mathcal{O}(n)$  and  $\mathcal{O}(m)$ , respectively. Thus we obtain:

► **Lemma 6.** *Let  $P$  be a simple polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple curves with  $n$  and  $m$  vertices on the boundary of  $P$ , separated by a horizontal line segment in  $P$ . After preprocessing  $P$  in  $\mathcal{O}(k)$  time, we can construct a pair of vertically-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$  with  $\mathcal{F}_\delta(R, B) = \mathcal{F}_\delta(\bar{R}, \bar{B})$  in  $\mathcal{O}((n+m) \log k)$  time. The curves  $\bar{R}$  and  $\bar{B}$  have  $\mathcal{O}(n)$  and  $\mathcal{O}(m)$  vertices, respectively.*



■ **Figure 5** Two types of bichromatic chords and the partitions of the parameter space that they induce. All green regions correspond to separated subcurves; the purple regions are split recursively.

► **Corollary 7.** *Let  $P$  be a simple polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple curves with  $n$  and  $m$  vertices on the boundary of  $P$ , separated by a horizontal line segment in  $P$ . After preprocessing  $P$  in  $\mathcal{O}(k)$  time, we can compute a doubly-separated partition of the parameter space of  $R$  and  $B$  that has cost  $\mathcal{O}((n+m)\log nm)$  in  $\mathcal{O}((n+m)\log nm)$  time.*

### 3.3 The general case

In this section, we start out in the general setting, where  $R$  and  $B$  are simple, interior-disjoint curves on the boundary of a simple polygon  $P$ , and construct a doubly-separated partition of the parameter space of  $R$  and  $B$ .

We follow the approach of Section 3.1, in that we partition the parameter space recursively, though this time based on horizontal *chords* of  $P$ . A chord is a maximal segment that does not go outside  $P$ . By our assumptions, a chord can have at most three points in common with the boundary of  $P$ . If possible, we will use bichromatic chords, which have a point in common with both curves. Each such chord  $e^*$  splits each curve into at most three subcurves: If  $e^*$  intersects  $R$  only at an endpoint of  $R$ , the chord “trivially” splits  $R$  into the curve  $R$  itself. Otherwise,  $R$  is split into the maximal prefix  $R[0, x]$  whose intersection with  $e^*$  is only the point  $R(x)$ , the maximal suffix  $R[x', 1]$  whose intersection with  $e^*$  is only the point  $R(x')$ , and the maximal subcurve  $R[x, x']$  bounded by  $e^*$ . See Figure 5. The split of  $B$  induced by  $e^*$  is defined analogously, but if  $R$  is split into three subcurves, then  $B$  can be split into at most two subcurves (and vice versa). A chord corresponds to a partition of the parameter space into the axis-aligned rectangles whose corresponding subcurves are induced by the splits of  $R$  and  $B$ . Thus, a chord corresponds to a partition into at most six regions.

If a bichromatic chord does not exist, then  $R$  and  $B$  can be separated trivially by a horizontal chord. Our partition algorithm works recursively, like the algorithm of Lemma 3. To bound the recursive depth, we use a chord that splits  $R$  and  $B$  into at most two and three subcurves each, where the first and last subcurves of each have a total of at most  $(n+m)/c$  vertices at either side of the chord, for some constant  $c > 1$ .

We prove the existence of a horizontal chord with the desired properties, and give a construction algorithm. The result follows from a relaxed version of the “polygon-cutting theorem” of Chazelle [6].

► **Lemma 8.** *If  $R$  and  $B$  are not already separated by a horizontal chord, there exists a horizontal chord that intersects both  $R$  and  $B$ . Furthermore, if  $n+m \geq 3$ , such a chord exists that splits  $R$  and  $B$  into at most three subcurves each, where for each pair  $(R_i, B_j)$  of subcurves not separated by the chord,  $|R_i| + |B_j| \leq 2(n+m)/3$ . Such a chord can be found in  $\mathcal{O}(k)$  time.*



With the existence and construction result for good chords at hand, we make an initial partition of the parameter space. The partition is not yet a doubly-separated partition, but we refine it afterwards into one that is.

We recursively partition the parameter space as follows. If  $|R| \leq 2$  and  $|B| \leq 2$  then we stop partitioning further. Otherwise, we compute a horizontal chord  $e^*$  of  $P$  with the algorithm of Lemma 8, taking  $\mathcal{O}(k)$  time.

Assuming  $P$  is in general position, the chord intersects one curve in at most two points and the other in at most one point. Suppose that  $e^*$  intersects  $R$  in two points, splitting it into the subcurves  $R_1 = R[0, x]$ ,  $R_2 = R[x, x']$  and  $R_3 = R[x', 1]$ , and that  $e^*$  intersects  $B$  in one point, splitting it into the subcurves  $B_1 = B[0, y]$  and  $B_2 = B[y, 1]$ . These five subcurves together define six axis-parallel rectangles in the parameter space, which is the partition corresponding to  $e^*$ .

Let  $\mathcal{P}$  be this partition. The subcurve  $R_2$  is bounded by  $e^*$ , and thus is separated from all of  $B$  by  $e^*$ . Further, the subcurve  $R_1$  is separated from  $B_2$ , and the subcurve  $R_3$  from  $B_1$ . Thus the only regions in the partition whose corresponding subcurves are not separated by a horizontal chord are the bottom-left region  $[0, x] \times [0, y]$  and the top-right region  $[x', 1] \times [y, 1]$ ; we recursively partition  $\mathcal{P}$  in these regions.

We analyze the partition in the same manner as we do for doubly-separated partitions. That is, we refer to the cost of an axis-aligned rectangle  $\mathcal{R} = [x, x'] \times [y, y']$  as  $\text{Cost}(\mathcal{R}) = |R[x, x']| + |B[y, y']|$ , the same as for doubly-separated regions, and refer to the cost of the partition  $\mathcal{P}$  as  $\text{Cost}(\mathcal{P}) = \sum_{\mathcal{R} \in \mathcal{P}} \text{Cost}(\mathcal{R})$ .

► **Lemma 9.**  *$\mathcal{P}$  has cost  $\mathcal{O}((n + m) \log nm)$  and can be computed in  $\mathcal{O}(k \log nm)$  time.*

We refine  $\mathcal{P}$  into a doubly-separated partition. Some regions in  $\mathcal{P}$  already correspond to line segments on  $R$  and  $B$ , and so are trivial. The other regions correspond to subcurves of  $R$  and  $B$  that are separated by a horizontal chord of  $P$ . We partition these regions further using the result of Corollary 7.

► **Lemma 10.** *Let  $P$  be a simple polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple, interior-disjoint curves on with  $n$  and  $m$  vertices on the boundary of  $P$ . We can compute a doubly-separated partition of the parameter space of  $R$  and  $B$  that has cost  $\mathcal{O}((n + m) \log^2 nm)$  in  $\mathcal{O}(k \log nm + (n + m) \log^2 nm)$  time. Additionally, in the same time bound, we can compute, for each region  $[x, x'] \times [y, y']$  in the partition, a pair of doubly-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$  with  $\mathcal{F}_\delta(R[x, x'], B[y, y']) = \mathcal{F}_\delta(\bar{R}, \bar{B})$  for all  $\delta$ . The curves  $\bar{R}$  and  $\bar{B}$  have  $\mathcal{O}(|R[x, x']|)$  and  $\mathcal{O}(|B[y, y']|)$  vertices, respectively.*

## 4 Computing the Fréchet distance

In this section, we discuss the main problem setting of this work. Let  $P$  be a simple polygon and let  $R$  and  $B$  be two simple, interior-disjoint curves on the boundary of  $P$ . We assume  $R$  is oriented clockwise with respect to  $P$ , and  $B$  counter-clockwise. We present a near-linear time algorithm for computing  $\bar{d}_F(R, B)$ .

Our algorithm makes use of a *decision* algorithm that, given a parameter  $\delta \geq 0$ , reports whether  $\bar{d}_F(R, B) \leq \delta$  or  $\bar{d}_F(R, B) > \delta$ . Recall from the preliminaries that Alt and Godau [2] showed that  $\bar{d}_F(R, B) \leq \delta$  if and only if there exists a bimonotone path in  $\mathcal{F}_\delta(R, B)$  from  $(0, 0)$  to  $(1, 1)$ . We develop an algorithm that decides whether such a path exists.

In Section 4.1, we use the doubly-separated partition constructed in Section 3 to efficiently propagate reachability information through the various regions. Recall that there are two types of regions in a doubly-separated partition: trivial regions and doubly-separated regions.

For trivial regions, we give a simple algorithm for propagating reachability information through them, using that these regions correspond to pairs of segments on  $R$  and  $B$ . For the other regions, we use the result of Section 2 instead, to also propagate reachability information through them. This leads to our decision algorithm. In Section 4.2, we turn our decision algorithm into an optimization algorithm.

#### 4.1 The decision algorithm

We present an algorithm for deciding whether  $\bar{d}_F(R, B) \leq \delta$  for a given parameter  $\delta \geq 0$ . Our algorithm decides whether a bimonotone path in  $\mathcal{F}_\delta(R, B)$  exists from  $(0, 0)$  to  $(1, 1)$ . Let  $\mathcal{P}^*$  be a doubly-separated partition of the parameter space of  $R$  and  $B$  computed with the algorithm of Lemma 10. We propagate reachability information through each region in  $\mathcal{P}^*$  separately.

Before we do so, we need to address one issue. For the regions  $\mathcal{R} \in \mathcal{P}^*$  where we have a pair of doubly-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$  whose free space is identical to that inside  $\mathcal{R}$ , we wish to apply the result of Lemma 2. However, this algorithm is for propagating reachability from a given discrete set of points to a given discrete set of points. Hence we first construct sets  $S_{\mathcal{R}}$  and  $T_{\mathcal{R}}$  to which we apply the result of Lemma 2.

We define the sets  $S_{\mathcal{R}}$  and  $T_{\mathcal{R}}$ . For this, we say that a point  $B(x)$  is *locally closest* to  $R(y)$  if an infinitesimal perturbation of  $B(x)$  while staying on  $B$  increases its distance to  $R(y)$ . We call points on  $R$  locally closest to points on  $B$  if a symmetric condition holds. The set  $T_{\mathcal{R}}$  is the set of all points  $(x^*, y^*)$  on the top and right sides of  $\mathcal{R}$  for which at least one of  $R(x^*)$  and  $B(y^*)$  is a vertex of  $R$  or  $B$ , or locally closest to the other point.

Let  $\mathcal{R} = [x, x'] \times [y, y'] \in \mathcal{P}^*$ . The set  $T_{\mathcal{R}}$  is defined as follows. For each edge  $e$  of  $R$ , we add the point  $(x^*, y')$  to  $T_{\mathcal{R}}$  if  $x^* \in [x, x']$  and  $R(x^*)$  is either a vertex of  $R$  or the point on  $e$  closest to  $B(y')$ . Symmetrically, for each edge  $e$  of  $B$ , we add the point  $(x', y^*)$  to  $T_{\mathcal{R}}$  if  $y^* \in [y, y']$  and  $B(y^*)$  is either a vertex of  $B$  or the point on  $e$  closest to  $R(x')$ .

The set  $S_{\mathcal{R}}$  is defined as all points in  $S \cap \mathcal{R}$ , as well as all points on the bottom and left sides of  $\mathcal{R}$  that are in a set  $T_{\mathcal{R}'}$  (for an adjacent region  $\mathcal{R}'$ ) and are  $\delta$ -reachable from  $S_{\mathcal{R}'}$ . The set  $S_{\mathcal{R}}$  is defined as all points on the bottom and left sides of  $\mathcal{R}$  that are in a set  $T_{\mathcal{R}'}$  (for an adjacent region  $\mathcal{R}'$ ) and are  $\delta$ -reachable from  $S_{\mathcal{R}'}$ . If  $\mathcal{R}$  contains the point  $(0, 0)$ , we set  $S_{\mathcal{R}} = \{(0, 0)\}$  instead.

The usefulness of these sets comes from Lemma 11, from which it follows that matchings can be assumed to enter and leave a region  $\mathcal{R}$  through points in  $S_{\mathcal{R}}$  and  $T_{\mathcal{R}}$ , respectively.

► **Lemma 11.** *There exists a Fréchet matching between  $R$  and  $B$  where for every matched pair of points  $(r, b)$ , at least one of  $r$  and  $b$  is a vertex or locally closest to the other point.*

Naturally, for our decision algorithm, we need to compute these sets efficiently:

► **Lemma 12.** *We can compute the set  $T_{\mathcal{R}}$  for all  $\mathcal{R} \in \mathcal{P}^*$  in  $\mathcal{O}(k + (n + m) \log^2 nm \log k)$  time altogether.*

We are now ready to give our decision algorithm, which decides whether a bimonotone path in  $\mathcal{F}_\delta(R, B)$  from  $(0, 0)$  to  $(1, 1)$  exists, for a given  $\delta \geq 0$ . First, we compute the doubly-separated partition  $\mathcal{P}^*$  of Lemma 10, taking  $\mathcal{O}(k \log nm + (n + m) \log^2 nm)$  time. The partition has cost  $\mathcal{O}((n + m) \log^2 nm)$ , and for each region  $[x, x'] \times [y, y']$  in this partition, either  $R[x, x']$  and  $B[y, y']$  are line segments, or we have additionally computed a pair of doubly-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$  with  $\mathcal{O}(|R[x, x']|)$  and  $\mathcal{O}(|B[y, y']|)$  vertices, respectively. We compute the sets  $T_{\mathcal{R}}$  for all regions  $\mathcal{R} \in \mathcal{P}^*$ , taking  $\mathcal{O}(k + (n + m) \log^2 nm \log k)$  time. The goal is then to compute, for each set  $T_{\mathcal{R}}$ , the subset of points that are  $\delta$ -reachable from  $(0, 0)$ .

Consider a region  $\mathcal{R} = [x, x'] \times [y, y'] \in \mathcal{P}^*$  and suppose that for each region  $\mathcal{R}'$  incident to the bottom or left side of  $\mathcal{R}$  we have already computed the subset of  $T_{\mathcal{R}'}$  of points that are  $\delta$ -reachable from  $(0, 0)$ . The union of these subsets forms the set  $S_{\mathcal{R}}$  that serves as the set of “entrances” for  $\mathcal{R}$ ; a point in  $T_{\mathcal{R}}$  is  $\delta$ -reachable from  $(0, 0)$  if and only if it is  $\delta$ -reachable from a point in  $S_{\mathcal{R}}$ .

If  $R[x, x']$  or  $B[y, y']$  has more than one edge, we have available a pair of doubly-separated  $x$ -monotone curves  $\bar{R}$  and  $\bar{B}$  with  $\mathcal{F}_{\delta}(R[x, x'], B[y, y']) = \mathcal{F}_{\delta}(\bar{R}, \bar{B})$ . We apply the algorithm of Lemma 2 to these curves, using the set  $S_{\mathcal{R}}$  for the “entrances” and the set  $T_{\mathcal{R}}$  for the “potential exits.” This algorithm reports all points in  $T_{\mathcal{R}}$  that are  $\delta$ -reachable from at least one point in  $S_{\mathcal{R}}$  in  $\mathcal{O}((|\bar{R}| + |\bar{B}|) \log |\bar{R}| |\bar{B}|) = \mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R}))$  time.

If both  $R[x, x']$  and  $B[y, y']$  are line segments, then we can check in constant time whether a given point in  $T_{\mathcal{R}}$  is reachable from a given point in  $S_{\mathcal{R}}$ , so we can report the set of points in  $T_{\mathcal{R}}$  that are  $\delta$ -reachable from at least one point in  $S_{\mathcal{R}}$  in  $\mathcal{O}(|S_{\mathcal{R}}| \cdot |T_{\mathcal{R}}|) = \mathcal{O}(1)$  time.

► **Lemma 13.** *If both  $R[x, x']$  and  $B[y, y']$  are line segments, then a point  $t \in T_{\mathcal{R}}$  is  $\delta$ -reachable from a point  $s \in S_{\mathcal{R}}$  if and only if  $t \in \mathcal{F}_{\delta}(R[x, x'], B[y, y'])$  and  $t$  lies above and to the right of  $s$ .*

We now have an algorithm that, given the sets  $S_{\mathcal{R}}$  and  $T_{\mathcal{R}}$ , reports the points in  $T_{\mathcal{R}}$  that are  $\delta$ -reachable in  $\mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R}))$  time. Applying this algorithm to all regions in  $\mathcal{P}^*$ , in some topological order, we eventually decide whether  $(1, 1)$  is  $\delta$ -reachable from  $(0, 0)$ , since it is included in a set  $T_{\mathcal{R}}$ . This takes

$$\sum_{\mathcal{R} \in \mathcal{P}^*} \mathcal{O}(\text{Cost}(\mathcal{R}) \log \text{Cost}(\mathcal{R})) = \mathcal{O}(\text{Cost}(\mathcal{P}^*) \log nm) = \mathcal{O}((n + m) \log^3 nm)$$

time. The computations of  $\mathcal{P}^*$  and the sets  $T_{\mathcal{R}}$ , which took  $\mathcal{O}(k \log nm + (n + m) \log^2 nm \log k)$  time in total, dominate the running time. However, these computations are independent of the decision parameter  $\delta$ , and so we view these computations as preprocessing. This gives the following result:

► **Lemma 14.** *Let  $P$  be a polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple, interior disjoint curves on the boundary of  $P$ , with  $n$  and  $m$  vertices. After preprocessing  $P$ ,  $R$  and  $B$  in  $\mathcal{O}(k \log nm + (n + m) \log^2 nm \log k)$  time, we can decide, given  $\delta \geq 0$ , whether  $\bar{d}_F(R, B) \leq \delta$  in  $\mathcal{O}((n + m) \log^3 nm)$  time.*

## 4.2 Obtaining an optimization algorithm

In this section, we turn the decision algorithm of the previous section into an algorithm that computes  $\bar{d}_F(R, B)$ , at the cost of a logarithmic factor in the running time when compared to the decision algorithm. For this, we derive a polynomial number of candidate distances, one of which is the actual Fréchet distance  $\bar{d}_F(R, B)$ . Specifically, for two sets of real numbers  $X$  and  $Y$ , their *Cartesian sum*, denoted  $X \oplus Y$ , is the multiset  $X \oplus Y = \{\{x + y \mid x \in X, y \in Y\}\}$ . We will show that we can compute two sets  $X$  and  $Y$  of  $\mathcal{O}((n + m) \log^2 nm)$  real numbers each, such that  $\bar{d}_F(R, B) \in X \oplus Y$ . After computing  $X$  and  $Y$  in near-linear time, we can use existing techniques [8, 10] for selection in Cartesian sums to binary search over the Cartesian sum  $X \oplus Y$  without explicitly computing its  $|X| \cdot |Y|$  many elements.

We again make use of the doubly-separated partition  $\mathcal{P}^*$  computed with the algorithm of Lemma 10. Additionally, we again use the sets  $T_{\mathcal{R}}$  for regions  $\mathcal{R} \in \mathcal{P}^*$ . Recall that there exists a Fréchet matching between  $R$  and  $B$  that corresponds to a bimonotone path in the parameter space that, for each region  $\mathcal{R} \in \mathcal{P}^*$  that it intersects, goes through a point in  $T_{\mathcal{R}}$ .

Thus, there exists a pair of points  $s \in T_{\mathcal{R}'}$  and  $t \in T_{\mathcal{R}}$ , for a region  $\mathcal{R}'$  and region  $\mathcal{R}$  incident to the top or right side of  $\mathcal{R}'$ , such that the minimum parameter  $\delta$  for which there exists a  $\delta$ -matching from  $s$  to  $t$  is the Fréchet distance.

Take regions  $\mathcal{R}'$  and  $\mathcal{R}$ , with  $\mathcal{R}$  incident to the top or right side of  $\mathcal{R}'$ . Let  $s \in T_{\mathcal{R}'}$  and let  $t \in T_{\mathcal{R}}$  be above and to the right of  $s$ . Let  $\delta^*$  be the minimum parameter for which there exists a  $\delta^*$ -matching from  $s$  to  $t$ . We construct sets  $X_{\mathcal{R}}$  and  $Y_{\mathcal{R}}$  such that  $\delta^* \in X_{\mathcal{R}} \oplus Y_{\mathcal{R}}$ .

If the subcurves corresponding to  $\mathcal{R}$  both are line segments, then from Lemma 13 we obtain that  $t$  is  $\delta$ -reachable from  $s$ , for some  $\delta \geq 0$ , if and only if both points lie in  $\mathcal{F}_{\delta}(R, B)$ . We set  $X_{\mathcal{R}}$  to be the set minimum values  $\delta$  for which points  $t \in T_{\mathcal{R}}$  lie in  $\mathcal{F}_{\delta}(R, B)$ . The set  $Y_{\mathcal{R}}$  is simply set to  $\{0\}$ . Taken over all regions in  $\mathcal{P}^*$ , the sets have a total cardinality of  $\mathcal{O}(\text{Cost}(\mathcal{P}^*)) = \mathcal{O}((n+m) \log^2 nm)$ .

If one of the subcurves corresponding to  $\mathcal{R}$  has more than one edge, then we have access to a pair of doubly-separated  $x$ -monotone curves whose  $\delta$ -free space is identical to the  $\delta$ -free space of  $R$  and  $B$  inside  $\mathcal{R}$ . Moreover, by Lemma 1, we can transform these curves into a pair of curves in  $\mathbb{R}$  that are separated by the point 0, such that the  $\delta$ -free space remains the same. Let  $\bar{R}$  and  $\bar{B}$  be these curves. Alt and Godau [2] observe that  $\delta^*$  must be one of the following values:

- the minimum value for which  $s$  and  $t$  lie in  $\mathcal{F}_{\delta^*}(\bar{R}, \bar{B})$ , or
- the distance between a vertex of  $\bar{R}$  (resp.  $\bar{B}$ ) and an edge of  $\bar{B}$  (resp.  $\bar{R}$ ), or
- the distance between two vertices of  $\bar{R}$  (resp.  $\bar{B}$ ) to an edge of  $\bar{B}$  (resp.  $\bar{R}$ ), if the vertices lie at equal distance to the edge.

Since  $\bar{R}$  and  $\bar{B}$  are curves in  $\mathbb{R}$  that are separated by a point, the candidates for  $\delta^*$  of second and third type coincide. Moreover, the distance from a vertex  $p$  of one curve to an edge  $e$  of the other curve is also the distance from  $p$  to the endpoint of  $e$  closest to  $p$ , which must be a vertex. Thus, the latter two types of candidates for  $\delta^*$  can be summarized as all pairwise distances between vertices of  $\bar{R}$  and vertices of  $\bar{B}$ . Exploiting the separation of  $\bar{R}$  and  $\bar{B}$  further, the pairwise distances between vertices of  $\bar{R}$  and vertices of  $\bar{B}$  can be represented by the Cartesian sum of two sets  $\bar{X}$  and  $\bar{Y}$ , containing the absolute values of the vertex values of  $\bar{R}$  and  $\bar{B}$ , respectively. We set  $X_{\mathcal{R}} \leftarrow \bar{X}$  and  $Y_{\mathcal{R}} \leftarrow \bar{Y}$ . Taken over all regions in  $\mathcal{P}^*$ , the sets have a total cardinality of  $\mathcal{O}(\text{Cost}(\mathcal{P}^*)) = \mathcal{O}((n+m) \log^2 nm)$ .

Let  $X = \bigcup_{\mathcal{R} \in \mathcal{P}^*} X_{\mathcal{R}}$  and  $Y = \bigcup_{\mathcal{R} \in \mathcal{P}^*} Y_{\mathcal{R}}$ . These sets have a total cardinality of  $\mathcal{O}((n+m) \log^2 nm)$ , and contain values  $x^* \in X$  and  $y^* \in Y$  such that  $\bar{d}_F(R, B) = x^* + y^*$ . Next we search over  $X \oplus Y$  to find the exact value of  $\bar{d}_F(R, B)$ .

After sorting  $X$  and  $Y$ , we can compute the  $i^{\text{th}}$  smallest value in  $X \oplus Y$ , for any given  $i$ , in  $\mathcal{O}(|X| + |Y|) = \mathcal{O}((n+m) \log^2 nm)$  time [8, 10]. There are  $|X| \cdot |Y|$  values in  $X \oplus Y$ . We binary search over the integers  $1, \dots, |X| \cdot |Y|$ , and at each considered integer  $i$ , we compute the  $i^{\text{th}}$  smallest value  $\delta$  in  $X \oplus Y$ . Then we use our decision algorithm to decide whether  $\bar{d}_F(R, B) \leq \delta$  and to guide the search to the value  $\delta^* \in X \oplus Y$  with  $\delta^* = \bar{d}_F(R, B)$ .

In the above search procedure, each step takes  $\mathcal{O}((n+m) \log^3 nm)$  time after preprocessing  $P$  and the curves  $R$  and  $B$  for decision queries, which takes  $\mathcal{O}(k \log nm + (n+m) \log^2 nm \log k)$  time (Lemma 14). We perform  $\mathcal{O}(\log(|X| \cdot |Y|)) = \mathcal{O}(\log nm)$  steps. Thus we obtain our main result:

► **Theorem 15.** *Let  $P$  be a polygon with  $k$  vertices. Let  $R$  and  $B$  be two simple, interior disjoint curves on the boundary of  $P$ , with  $n$  and  $m$  vertices. We can compute  $\bar{d}_F(R, B)$  in  $\mathcal{O}(k \log nm + (n+m)(\log^2 nm \log k + \log^4 nm))$  time.*

---

References

---

- 1 Helmut Alt and Michael Godau. Measuring the resemblance of polygonal curves. In *Proc. Eighth Annual Symposium on Computational Geometry (SoCG)*, pages 102–109. ACM, 1992. doi:10.1145/142675.142699.
- 2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995. doi:10.1142/S0218195995000064.
- 3 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly sub-quadratic algorithms unless SETH fails. In *Proc. 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 661–670, 2014. doi:10.1109/FOCS.2014.76.
- 4 Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds. *International Journal of Computational Geometry & Applications*, 27(1-2):85–120, 2017. doi:10.1142/S0218195917600056.
- 5 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. SETH says: Weak Fréchet distance is faster, but only if it is continuous and in one dimension. In *Proc. 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2887–2901, 2019. doi:10.1137/1.9781611975482.179.
- 6 Bernard Chazelle. A theorem on polygon cutting with applications. In *proc. 23rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 339–349. IEEE Computer Society, 1982. doi:10.1109/SFCS.1982.58.
- 7 Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012. doi:10.1007/s00454-012-9402-z.
- 8 Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM Journal of Computing*, 13(1):14–30, 1984. doi:10.1137/0213002.
- 9 Atlas F. Cook IV and Carola Wenk. Geodesic Fréchet distance inside a simple polygon. *ACM Transactions on Algorithms*, 7(1):9:1–9:19, 2010. doi:10.1145/1868237.1868247.
- 10 Andranik Mirzaian and Eshrat Arjomandi. Selection in X+Y and matrices with sorted rows and columns. *Information Processing Letters*, 20(1):13–17, 1985. doi:10.1016/0020-0190(85)90123-1.
- 11 Thijs van der Horst, Marc van Kreveld, Tim Ophelders, and Bettina Speckmann. The geodesic Fréchet distance between two curves bounding a simple polygon. *CoRR*, abs/2501.03834, 2025. doi:10.48550/arXiv.2501.03834.