The Complexity of Realizing Free Spaces

Hugo A. Akitaya* Maike Buchin† Majid Mirzanezhad‡ Leonie Ryvkin§
Carola Wenk¶

Abstract

The free space diagram is a popular tool to analyze and compute the well-known Fréchet distance, as the Fréchet distance is used in many application areas, and hence many variants have been established to cover the specific needs of the various applications. When exploring distance measures related to the Fréchet distance, one again works with free space diagrams. Often the question arises, whether or not a certain pattern in the free space diagram is *realizable* or not, i.e., whether there exists a pair of polygonal chains that correspond to the pattern in question. We show that the problem is $\exists\mathbb{R}$-complete.

1 Introduction

The Fréchet distance is an important distance measure for curves used in many applications, including computer aided design, geographic data analysis and the comparison of protein chains. A popular tool for computing the Fréchet distance of two curves is the *free space diagram*, which is the cross-product of the parameter spaces of the curves partitioned into *free space* and its complement. Free space is the sublevel set of the distance function for a given $\varepsilon > 0$. For two piecewise linear curves comprised of m and n line segments parameterized by their natural arc-length parameterizations, it is well-known that the free space diagram consists of mn cells, and the free space in each cell is a cropped ellipse [3]. The Fréchet distance is at most ε iff there exists a xy-monotone path in the free space diagram that covers the parameter spaces of both curves. Hence, to compute the Fréchet distance, one searches for such a path in the free space diagram. For different applications, many variants of the Fréchet distance have been developed. These are typically also computed using the free space diagram, which needs to be analyzed for this. Runtimes of the resulting algorithms usually depend directly on the complexity of the free space diagram. It is known that the Fréchet distance cannot be computed in subquadratic time unless SETH fails [4].

In this paper we study the inverse problem: Given a free space diagram, do there exist a pair of curves that generate this free space diagram? The problem was introduced by Buchin, Ryvkin and Wenk [6]. They give algorithms for a restricted family of inputs and show that maximizing the number of realizing cells is NP-hard. Here, we show that the problem is indeed $\exists\mathbb{R}$-complete.

2 Definitions and Preliminaries

We use some notation from Buchin et al. [5]. We define a polygonal curve P as the image of a continuous piecewise linear function $\gamma_P : [0, \ell_P] \to \mathbb{R}^d$ where ℓ_P is the geometric length of P. The endpoints of segments in the image of P are called *vertices*, and we represent a curve by its sequence of vertices $P = \langle p_0, \ldots, p_n \rangle$. Let $|p_i| = \sum_{j=1}^{i} \|p_{j-1}p_j\|$ denote the length up to p_i, where $\|.\|$ denotes the Euclidian norm. Note that $|p_0| = 0$ and $|p_n| = \ell_P$. Without loss of generality, we consider that $\gamma_P(|p_i|) = p_i$, otherwise we can change

*Department of Computer Science, University of Massachusetts Lowell, USA
†Department of Computer Science, Ruhr University Bochum, Germany
‡Transportation Research Institute, College of Engineering, University of Michigan – Ann Arbor, USA
§Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
¶Department of Computer Science, Tulane University, Louisiana, USA. Supported by NSF grant CCF-2107434.

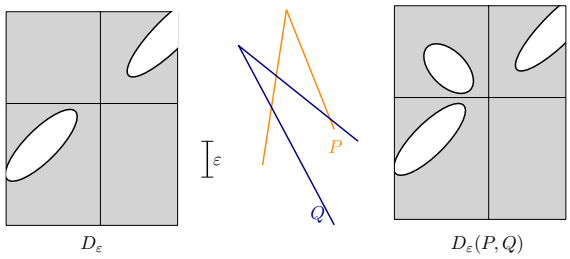

Figure 1: A given diagram D_ε, the curves as computed from the partially full cells, and their free space diagram $D_\varepsilon(P,Q)$.

the parametrization of P. For two polygonal curves $P = \langle p_0, \ldots, p_n \rangle$ and $Q = \langle q_0, \ldots, q_m \rangle$ in $\mathbb{R}^d$, and a real number $\varepsilon > 0$, the free space is defined as $F_\varepsilon(P,Q) = \{(r,t) \mid \|P(r) - Q(t)\| \leq \varepsilon\}$. The free space diagram puts the free space information in an $\ell_P \times \ell_Q$ rectangle divided into an $n \times m$ grid. More precisely, we define $D_\varepsilon(P,Q)$ as the colored rectangle $R = [0, \ell_P] \times [0, \ell_Q] \subset \mathbb{R}^2$, where a point $(p,q) \in R$ is colored white if and only if $(p,q) \in F_\varepsilon(P,Q)$. The grid formed by the set of segments $\{|p_i| \times [0, \ell_Q] \mid i \in \{0, \ldots, n\}\} \cup \{[0, \ell_P] \times |q_j| \mid j \in \{0, \ldots, m\}\}$, subdivides R into $n \times m$ cells $C_{i,j}$.

We define further important terms concerning the free space diagram: A cell $C_{i,j}$ is called *empty* (or *gray*) if $C_{i,j} \cap F_\varepsilon = \emptyset$, and a cell is called *full* (or *white*) if $C_{i,j} \cap F_\varepsilon = C_{i,j}$. In the intermediate case, where $\emptyset \neq C_{i,j} \cap F_\varepsilon \neq C_{i,j}$, the cell $C_{i,j}$ is called *partially full*. We use the following known result stated in [6]:

Lemma 1. *Given a partially full free space cell $C_{i,j}$, the distance ε and four points on the boundary of the ellipse within the cell, none of which are mirror images of another one with respect to the ellipse's major and minor axes, we can compute the corresponding segments' relative placement.*

A given diagram D_ε is called *realizable* if there exist curves P and Q such that $D_\varepsilon(P,Q) = D_\varepsilon$. In Figure 1, the given diagram on the left-hand side is not realizable: the curves obtained using Lemma 1 from the two components in cells $C_{1,1}$ and $C_{2,2}$ correspond to the free space diagram on the right-hand side of the figure, which features an additional component in cell $C_{2,1}$.

3 $\exists\mathbb{R}$-completeness.

Containment in $\exists\mathbb{R}$ can be shown by expressing the problem using real inequalities. We omit the details due to space restrictions. We reduce from the problem of deciding whether a linkage has a planar realization which was shown $\exists\mathbb{R}$-hard by Abel et al. [1, 2]. A *mechanical linkage* is a mechanism made of rigid bars connected at hinges usually used to convert between two movements. It can be modeled as a weighted graph $G = (V(G), E(G), \ell_G)$ with a function $\Pi : W \to \mathbb{R}^2$, where $W \subseteq V(G)$, that represents vertices whose positions are *pinned*. A *configuration* C of a linkage $\mathcal{L} = (G, \Pi)$ is a straight-line drawing of G where the length of each edge $e \in E(G)$ is $\ell_G(e)$ and the position of each vertex $w \in W$ is $\Pi(w)$. A configuration C is *noncrossing* if C is a plane graph.

Abel et al. [1, 2] showed that the linkage realization problem remains hard for a series of restrictions on the input linkage $\mathcal{L}$. We restate a simplified form of the main theorem in [2] summarizing the restrictions.

Theorem 2. *[Simplified from Theorem 2.2.13 in [2]] Given a linkage $\mathcal{L} = (G, \Pi)$ and a combinatorial embedding (clockwise circular order of edges around each vertex) σ of G, deciding whether there exists a planar realization of $\mathcal{L}$ is $\exists\mathbb{R}$-hard even if the following constraints are enforced:*

1. *G is connected and the length of every edge is integer.*
2. *A set of edge disjoint subgraphs H of G can be assigned* rigid*, meaning that each angle between consecutive (in σ) incident edges in H is prescribed from $\{90°, 180°, 270°, 360°\}$. Each subgraph H is a tree, and an edge in $E(G) \setminus E(H)$ incident to H must be incident to a leaf of H.*

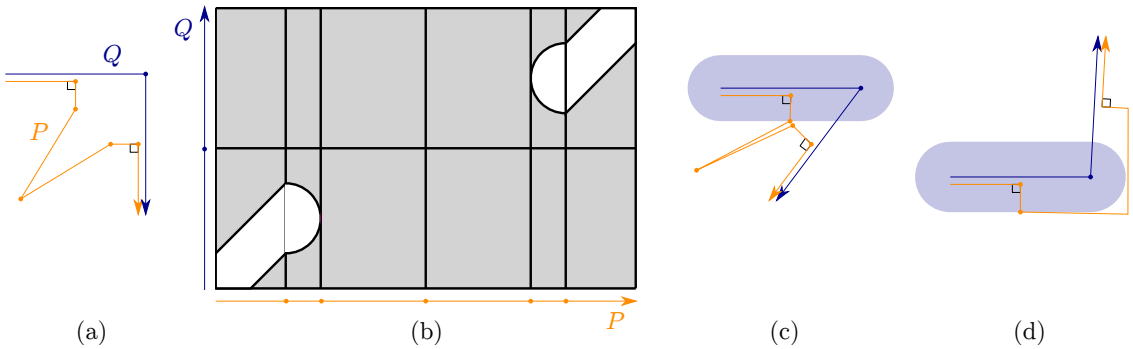

Figure 2: The angle gadget. (a) The 90° configuration and (b) its free-space diagram. (c) and (d) shows the extremal configurations of the gadget with angles $2 \cdot \tan^{-1}(1/2) \approx 53.13°$ and 270°, respectively.

3. *Only three vertices are pinned ($|\Pi| = 3$), all three belong to the same rigid subgraph H (described in constraint (2)), and they are not collinear.*

4. *For every noncrossing configuration C of $\mathcal{L}$ that satisfies constraints (1–3), we are promised that:*

   (a) *C agrees with σ.*

   (b) *The angles that are not prescribed by constraint (2) lie strictly between 60° and 240°.*

   (c) *The feature size (minimum distance from a vertex to a nonincident edge) is at least a constant ϕ.*

We call a vertex *rigid* if it is incident to at least two edges of the same rigid subgraph H. Else, we call the vertex *nonrigid*. By constraint (2), every angle incident to a rigid vertex is prescribed while no angle in a nonrigid vertex is prescribed (which by (4b) can only vary in the interval $(60, 240)$).

Reduction. Given $\mathcal{L}$ and σ satisfying the constraints in Theorem 2, we construct an instance D_ε as follows. While there is a cycle in G, split one edge in a cycle by placing a new vertex in its midpoint and performing a vertex split, creating two copies of the new vertex, each attached to half of the original edge. We end up with a tree T. Let T' be the multigraph obtained by doubling each edge of T. Intuitively, D_ε will force curves P and Q to roughly trace a planar Eulerian circuit of T' using the combinatorial embedding σ. (Up to a reflection and translation since D_ε can only specify the relative placement of P and Q.) More precisely, Q will be exactly a planar Eulerian circuit of T' while P will trace the same circuit but avoid an ε-neighborhood of each nonrigid vertex using our *angle gadget* (described later) which will allow these angles to lie freely between 60° and 240°. Note that there are two possible ways to choose a planar Eulerian circuit of T' agreeing with σ: circling the "outline" of the embedding σ clockwise or counterclockwise. Both P and Q trace the "outline" σ *counterclockwise*.Without loss of generality, we assume $\phi < 6$, scaling the linkage by a constant factor if necessary. We chose $\varepsilon = 1$ so that edges of P and Q that correspond to an edge e of G will be close to each other and far from other edges. Recall that every partially full cell determines the relative position of the corresponding pair of edges. Thus, the four edges (two from P and two from Q) that correspond to the traversal of e will be fixed relative to one another and lie on top of each other. They then simulate edge e. The angle gadget guarantees flexibility so that the angle between incident edges can vary accordingly. We add free space components to make the newly introduced vertices rigid: Their relative position is locked by Lemma 1 forming a 180° angle.

We now describe the *angle gadget*, shown in Figure 2. The gadget is represented by the 12 free-space cells shown in Figure 2(b). As explained before, the gadget is located at a small neighborhood of a vertex v of Q and the figure only shows the portion of the free space relative to this neighborhood. We refer to this angle gadget as $\mathsf{AngGadget}(v)$. Note that v is a degree-2 copy of a vertex v^* of G. For clarity, we refer to all the copies of v^* in Q with different labels. (By construction, there are $\deg(v^*)$ copies of each $v^* \in V(G)$, except for the starting vertex of the Eulerian circuit which will have an extra copy.) Let $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$ be the

two edges of Q incident to v, and let $\overleftarrow{e_1}$ and $\overleftarrow{e_2}$ be the corresponding copies going in the opposite direction in Q, respectively. Locally, P has two edges $\overrightarrow{e_1}'$ and $\overrightarrow{e_2}'$ that overlap with $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$, respectively. We enforce the overlap by making all free-space cells relative to $\overrightarrow{e_1}'$ (resp., $\overrightarrow{e_2}'$) empty except for the ones relative to $\overrightarrow{e_1}$ and $\overleftarrow{e_1}$ (resp., $\overrightarrow{e_2}$ and $\overleftarrow{e_2}$) which are partially full, containing an upward and downward 45° full strip. The distance between v and the endpoints of $\overrightarrow{e_1}'$ and $\overrightarrow{e_2}'$ closest to v is 2 by Lemma 1. We place four edges $(\overrightarrow{e_{1,a}}, \overrightarrow{e_{1,b}}, \overrightarrow{e_{1,c}}, \overrightarrow{e_{1,d}})$ between $\overrightarrow{e_1}'$ and $\overrightarrow{e_2}'$ of lengths 1, 3, 3, and 1 in this order. Only the edges of length 1 have corresponding partially full cells: $C_{\overrightarrow{e_{1,a}},\overrightarrow{e_1}}$ and $C_{\overrightarrow{e_{1,a}},\overleftarrow{e_1}}$ contain half of a disk of radius 1.

Lemma 3. *Given a realization of P and Q, assume that $(\overrightarrow{e_{1,a}}, \overrightarrow{e_{1,b}}, \overrightarrow{e_{1,c}}, \overrightarrow{e_{1,d}})$ lie to the right of $(\overrightarrow{e_1}, \overrightarrow{e_2})$. Then, $\overrightarrow{e_1}$ and $\overleftarrow{e_1}$ (resp., $\overrightarrow{e_1}$ and $\overleftarrow{e_1}$) lie exactly on top of each other, and the angle to the right of $(\overrightarrow{e_1}, \overrightarrow{e_2})$ is strictly between $2 \cdot \tan^{-1}(1/2) \approx 53.13°$ and 270°.*

Proof. The fact that $\overrightarrow{e_1}$ and $\overleftarrow{e_1}$ lie exactly on top of each other is a consequence of applying Lemma 1 to $\overrightarrow{e_1}$ and $\overrightarrow{e_1}'$, and to $\overrightarrow{e_1}'$ and $\overleftarrow{e_1}$. We now focus on the angle constraint. Note that by Lemma 1, the relative positions of $\overrightarrow{e_1}$ and $\overrightarrow{e_{1,a}}$ (resp., $\overrightarrow{e_2}$ and $\overrightarrow{e_{1,d}}$) is fixed. If we fix the positions of $\overrightarrow{e_{1,a}}$ and $\overrightarrow{e_{1,d}}$, then the positions of $\overrightarrow{e_{1,b}}$ and $\overrightarrow{e_{1,c}}$ are completely determined: There are two points whose distance is 3 from the endpoints of $\overrightarrow{e_{1,a}}$ and $\overrightarrow{e_{1,d}}$; one of them causes $\overrightarrow{e_{1,b}}$ and $\overrightarrow{e_{1,c}}$ to intersect with Q which can't happen since their free-space cells are empty. If the angle is $2 \cdot \tan^{-1}(1/2)$ or smaller, the common endpoint of $\overrightarrow{e_{1,c}}$ and $\overrightarrow{e_{1,d}}$ would lie in the closed ε-neighborhood of $\overrightarrow{e_1}$ and $C_{\overrightarrow{e_1},\overrightarrow{e_{1,c}}}$ would not be empty (Figure 2(c)), a contradiction. If the angle is 270° or greater, a portion of $\overrightarrow{e_{1,b}}$ would lie in the closed ε-neighborhood of $\overrightarrow{e_1}$ and $C_{\overrightarrow{e_1},\overrightarrow{e_{1,b}}}$ would not be empty (Figure 2(d)), a contradiction. For all values in between there is a placement for $\overrightarrow{e_{1,b}}$ and $\overrightarrow{e_{1,c}}$ away from $\overrightarrow{e_1}$ and $\overrightarrow{e_2}$, making the section of the free-space diagram exactly as required. □

Theorem 4. *It is $\exists\mathbb{R}$-complete to decide whether a given a free space diagram D_ε is realizable in 2D.*

Proof sketch. Given a positive instance of linkage realization, Theorem 2(4) and Lemma 3 guarantee that we can find a placement of P and Q realizing D_ε. The other direction is a little more subtle. D_ε forces Q to trace σ exactly. If there is a valid placement of P and Q one can find a noncrossing configuration of $\mathcal{L}$ obtained by the image of Q. If such a configuration does not satisfy Theorem 2(4), that would contradict Theorem 2. Thus the promise in Theorem 2(4) must also be fulfilled by the Fréchet realization instance and the angles in each angle gadget would indeed be between 60° and 240°. □

References

[1] Zachary Abel, Erik D Demaine, Martin L Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B Schardl. Who needs crossings? Hardness of plane graph rigidity. In *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51, pages 3:1–3:15, 2016.

[2] Zachary Ryan Abel. *On folding and unfolding with linkages and origami.* PhD thesis, Massachusetts Institute of Technology, 2016.

[3] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1-2):75–91, 1995.

[4] Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless seth fails. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 661–670, 2014.

[5] Kevin Buchin, Jinhee Chun, Maarten Löffler, Aleksandar Markovic, Wouter Meulemans, Yoshio Okamoto, and Taichi Shiitada. Folding free-space diagrams: Computing the Fréchet distance between 1-dimensional curves (multimedia contribution). In *33rd International Symposium on Computational Geometry, (SoCG'17)*, volume 77 of *LIPIcs*, pages 64:1–64:5, 2017.

[6] Maike Buchin, Leonie Ryvkin, and Carola Wenk. On the realizability of free space diagrams. In *37th European Workshop on Computational Geometry (EuroCG)*, 2021.