Faster Fréchet Distance under Transformations

Kevin Buchin
Technical University of Dortmund, Germany

Maike Buchin
Ruhr University Bochum, Germany

Zijin Huang
University of Sydney, Australia

André Nusser
Université Côte d'Azur, CNRS, Inria, France

Sampson Wong
BARC, University of Copenhagen, Denmark

Abstract

We study the problem of computing the Fréchet distance between two polygonal curves under transformations. First, we consider translations in the Euclidean plane. Given two curves π and σ of total complexity n and a threshold $\delta \geq 0$, we present an $\tilde{\mathcal{O}}(n^{7+\frac{1}{3}})$ time algorithm to determine whether there exists a translation $t \in \mathbb{R}^2$ such that the Fréchet distance between π and $\sigma + t$ is at most δ. This improves on the previous best result, which is an $\mathcal{O}(n^8)$ time algorithm.

We then generalize this result to any class of rationally parameterized transformations, which includes translation, rotation, scaling, and arbitrary affine transformations. For a class $\mathcal{T}$ of rationally parametrized transformations with k degrees of freedom, we show that one can determine whether there is a transformation $\tau \in \mathcal{T}$ such that the Fréchet distance between π and $\tau(\sigma)$ is at most δ in $\tilde{\mathcal{O}}(n^{3k+\frac{4}{3}})$ time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Fréchet distance, curve similarity, shape matching

Funding *André Nusser*: This work was supported by the French government through the France 2030 investment plan managed by the National Research Agency (ANR), as part of the Initiative of Excellence of Université Côte d'Azur under reference number ANR-15-IDEX-01.

1 Introduction

Many applications require determining the similarity of two geometric shapes, disregarding their location or orientation. More specifically, *shape matching* asks for the distance between two shapes if we allow the shapes to be transformed to minimize their distance [1, 3, 53]. Transformations may include translations, rotations, scaling, or a combination thereof. In this paper, we focus on polygonal curves under the Fréchet distance. Curves occur in many applications and need to be matched whenever they describe a local pattern, for example to recognize handwritten characters [47] or trademarks [5]. The Fréchet distance is arguably the most popular distance measure for curves in computational geometry. There has been significant algorithmic progress on the Fréchet distance, for instance, most recently on approximation [30, 50, 51], data structures [6, 12, 17, 26, 36, 37, 41, 40], algorithm engineering [8, 18, 33, 15, 13], simplification [11, 27, 49, 52], clustering [16, 19, 25, 42, 46, 20], Fréchet variants [21, 22, 23, 32, 35, 38, 39], and its complexity in general [10, 24, 29, 28].

The Fréchet distance under translation is defined as the minimum Fréchet distance for any translation of the curves. Computing the Fréchet distance under translations was first studied by Efrat, Indyk and Venkatasubramanian [34] and by Alt, Knauer and Wenk [4, 45, 54].

For two curves of total complexity n in $\mathbb{R}^2$, an $\tilde{\mathcal{O}}(n^{10})$ time algorithm[1] for the Fréchet distance under translation was presented in [34], and an $\tilde{\mathcal{O}}(n^8)$ time algorithm was presented in [4]. Wenk [54] generalized the approach in [4] to higher dimensions and a wide range of other transformations, e.g., for rotations or scalings in 2D in $\tilde{\mathcal{O}}(n^5)$ time, or for affine transformations in d dimensions in $\tilde{\mathcal{O}}(n^{3(d^2+d)+2})$ time.

Despite significant algorithmic progress on various aspects of the Fréchet distance, no faster algorithms for computing the (continuous) Fréchet distance under transformations have been developed until now. In contrast, for computing the discrete Fréchet distance under translation, first an $\tilde{\mathcal{O}}(n^6)$-time algorithm [43] was developed, then an $\tilde{\mathcal{O}}(n^5)$-time algorithm [7], and more lately an $\tilde{\mathcal{O}}(n^{4+\frac{2}{3}})$-time algorithm [14].

Our Contribution

In this paper, we present the first progress on the Fréchet distance under transformations since its introduction [4, 34, 45, 54]. Similar to the algorithm in [4, 54], our algorithm can be used for a wide range of classes of transformations. Specifically, given two curves π and σ of total complexity n and a threshold $\delta \geq 0$, we want to determine whether there is a transformation τ from a given class of transformations, such that the Fréchet distance between π and $\tau(\sigma)$ is at most δ. Our algorithm improves the running time for translations in two dimensions from $\mathcal{O}(n^8)$ to $\tilde{\mathcal{O}}(n^{7+\frac{1}{3}})$. More generally, we improve the running time for the various classes of transformations (and dimensions) given by Wenk [4] by roughly a factor of $n^{2/3}$. For example, for rotations or scalings in $\mathbb{R}^2$, our algorithm runs in $\tilde{\mathcal{O}}(n^{4+\frac{1}{3}})$ time and for affine transformations in d dimensions in $\tilde{\mathcal{O}}(n^{3(d^2+d)+\frac{1}{3}})$ time.

We first present our approach for the special case of translations in two dimensions. Then we generalize our approach to other transformations and higher dimensions. Similarly to the approach in [4], we compute an arrangement in the space of transformations, which in the case of 2D translations has complexity $\mathcal{O}(n^6)$. In [4], the Fréchet distance is computed for every vertex of this arrangement in $\mathcal{O}(n^2)$ time per vertex by using the *free space diagram*, which results in an overall running time of $\mathcal{O}(n^8)$. In our approach, we instead traverse the arrangement, making use of the fact that the free space diagram for adjacent faces in the arrangement is similar.

The challenge with using this approach is that it requires a data structure for dynamic directed graph reachability. This is a challenging problem on its own that to the best of our knowledge mostly saw progress on planar graphs [31, 44, 48] or more specifically grid graphs [7, 14]. In [14], a data structure with efficient updates and queries for reachability in a dynamic directed grid graph is presented. However, while the discrete Fréchet distance naturally reduces to a reachability problem on such a grid, this is not the case for the (continuous) Fréchet distance. An open question is whether there is a dynamic graph reachability data structure more suitable for the case of the continuous Fréchet distance under transformations.

We note that in an independent work, progress was made on the one-dimensional Fréchet distance under translation or scaling [9]. In this work, the authors present an $\tilde{\mathcal{O}}(n^{8/3})$ algorithm for both problems, making use of the offline dynamic data structure of [14].

[1] By $\tilde{\mathcal{O}}(\cdot)$ we hide (poly-)logarithmic factors in n.

Structure

We provide an outline for the remainder of this paper. In Section 2, we cover the preliminaries for our paper. In Section 3, we first present a detailed analysis of the changes in the free space diagram when traversing the arrangement in the space of translations. We refer to these changes as *events*. The free space diagram induces a directed graph, which we refer to as *free space graph* and in which we need to perform a reachability query. We analyze how the graph changes for the various types of events. In Section 4, we then show how these events can be modeled by a small number of changes to a suitable grid graph, which allows us to utilize the data structure in [14]. The challenge is that at an event rows or columns in the free space graph may swap, or rows and columns may appear or disappear, which are operations that cannot be modeled directly in the grid graph. In Section 5, we show how to generalize our algorithm to other classes of transformations.

2 Preliminaries

A d-dimensional polygonal curve π is a piecewise linear curve represented as a continuous mapping from $[1, n]$ to $\mathbb{R}^d$. For integer i, the point $\pi_i = \pi[i]$ is a vertex, and $\overline{\pi_i} = \pi_i\pi_{i+1} = \pi[i, i+1]$ is a segment.

The Fréchet distance is a popular measure of the similarity between two polygonal curves. An *orientation-preserving reparameterization* is a continuous and bijective function $f : [0, 1] \to [0, 1]$ such that $f(0) = 0$, and $f(1) = 1$. The $\mathbf{width}_{f,g}(\pi, \sigma)$ between two curves π and σ with respect to the reparameterizations f and g, is defined as follows.

$$\mathbf{width}_{f,g}(\pi, \sigma) = \max_{t\in[0,1]} \|\pi(f(t)) - \sigma(g(t))\|$$

Consider the scenario where a person is walking their dog with a leash connecting them: the person needs to stay on π while walking according to f, and the dog needs to stay on σ while walking according to g. The maximum leash length is the width between π and σ with respect to the reparameterizations f and g. The standard Fréchet distance $d_{\mathcal{F}}(\pi, \sigma)$ is the minimum leash length required over all possible walks (defined by reparameterizations f and g).

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

Problems relating to the Fréchet distance are commonly solved in a configuration space called the *freespace diagram*.

The *freespace* $\mathcal{F}_\delta(\pi, \sigma)$ is a collection of points in $\mathbb{R}^2$ in the range $[1, n] \times [1, n]$. A point (x, y) is in the freespace if the Euclidean norm (distance) $\|\pi[x] - \sigma[y]\|$ between $\pi[x]$ and $\sigma[y]$ is at most δ. As opposed to the free space, we will call $[1, n] \times [1, n] \setminus \mathcal{F}_\delta(\pi, \sigma)$ the *forbidden space*.

$$\mathcal{F}_\delta(\pi, \sigma) = \{(x, y) \in [1, n] \times [1, n] \mid \|\pi(x) - \sigma(y)\| \leq \delta\}$$

The *freespace diagram* $\mathcal{D} = \mathcal{D}_\delta(\pi, \sigma)$ is the partition of the freespace into $n \times n$ cells. A cell $C_{i,j}$ in $\mathcal{D}$ contains the freespace in the range $[i, i+1] \times [j, j+1]$. Alt and Godau [2] made the critical observation that the freespace within a cell $C_{i,j}$ is an intersection between an ellipse $E_{i,j}$ and the square $S_{i,j} = [i, i+1] \times [j, j+1]$; it is convex with constant description complexity. They also showed that $d_{\mathcal{F}}(\pi, \sigma) \leq \delta$ if and only if there is a bi-monotone path in $\mathcal{D}_\delta(\pi, \sigma)$ from $(1, 1)$ to (n, n) through the freespace.

An intersection between the boundaries $\partial E_{i,j}$ and $\partial S_{i,j} \setminus \{(i,j),(i+1,j),(i,j+1),(i+1,j+1)\}$ is called a *critical point*. A point $(i,j) \in \mathbb{N} \times \mathbb{N}$ is a *corner point*. We say the line $\pi[i] \times [1,n]$ is the *freespace diagram boundary* defined by π_i (and analogous for σ_i). We say the strip $[i,i+1] \times [1,n]$ is the ith *column*, and the strip $[1,n] \times [i,i+1]$ is the ith *row*. We say a critical point p is a row (resp. column) critical point if p lies on a vertical (resp. horizontal) boundary.

3 From freespace reachability to graph reachability

In this and the next section, we describe our ideas using an intuitive class of transformations: translation in $\mathbb{R}^2$. Specifically, we determine whether there exists a translation vector $t = \lambda\vec{v}$ such that $d_{\mathcal{F}}(\pi, \sigma + t) \leq \delta$, where $\lambda \in \mathbb{R}_{\geq 0}$ and $\vec{v}$ is a fixed directional vector. For this, we first describe how to transform the freespace reachability problem into a graph reachability problem.

Using the freespace diagram $\mathcal{D} = \mathcal{D}_\delta(\pi,\sigma)$, we construct a *refined freespace diagram (refined FSD or FSD for short)* (see Figure 1, left). Let $l(\pi_i)$ (resp. $l(\sigma_j)$) be the vertical (resp. horizontal) boundary of $\mathcal{D}$ defined by π_i (resp. σ_j). For every critical point p on the boundaries, we draw a perpendicular *grid line* $l(p)$ through p. Note that by definition, a critical point does not coincide with a corner point of $\mathcal{D}$. If p lies on the horizontal boundary, $l(p)$ is a vertical line; otherwise, $l(p)$ is a horizontal line. The refined FSD includes all grid lines, the freespace diagram boundaries, and their intersections. For simplicity, we redefine $\mathcal{D}_\delta(\pi,\sigma)$ to denote the refined FSD. Let the intersection between a grid line and a FSD boundary be a *propagated critical point*.

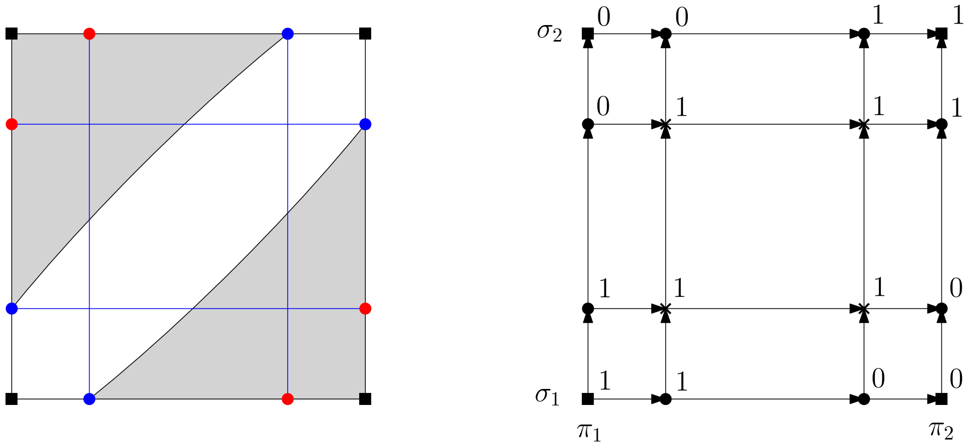

Figure 1 From the refined freespace diagram to the refined freespace graph. On the left freespace diagram (FSD), the corner points, critical points, and propagated critical points are marked by squares, blue circles, and red circles, respectively. On the right freespace diagram graph (FSG), the corner, boundary, and interior vertices are marked by squares, circles, and crosses, respectively.

Using the refined FSD $\mathcal{D}_\delta(\pi,\sigma)$, we construct a *refined freespace graph (refined FSG or FSG for short)* $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi,\sigma)$ as follows (see Figure 1, right). For every intersection between a grid line and a FSD boundary, add a *boundary vertex*. For every intersection between two grid lines, add an *interior vertex*. For every intersection between two freespace boundaries, add a *corner vertex*. Note that each vertex in $\mathcal{G}^f$ is uniquely defined by an ordered pair (p,q), where $l(p)$ is vertical and $l(q)$ is horizontal; p (resp. q) is either a vertex of π (resp. σ) or a critical point in $\mathcal{D}$ — let $\mathbf{v}(p,q)$ be such vertex in $\mathcal{G}^f$. Each vertex is assigned a weight that is either 1 or 0. For every vertex $u = \mathbf{v}(p, q)$, if $a = l(p) \cap l(q)$ lies on a boundary of the freespace diagram, we set $\mathbf{w}(u) = 1$ if a lies in the freespace or $\mathbf{w}(u) = 0$ if otherwise. We set the weights of the rest of the vertices, the interior vertices, to 1. We say a vertex u is *activated* if $\mathbf{w}(u) = 1$ or *deactivated* if $\mathbf{w}(u) = 0$. We say a vertex $\mathbf{v}(\pi_i, \cdot)$ (resp. $\mathbf{v}(\cdot, \sigma_j)$) lies on the freespace graph boundary defined by π_i (resp. σ_j).

To construct edges, we use the geometric positions of the grid lines and the FSG boundaries. For vertices in $\mathcal{G}^f$ defined by every grid line or FSG boundary $l(p)$, add a directed edge $(\mathbf{v}(p, q), \mathbf{v}(p, q'))$ to $\mathcal{G}^f$ if $l(q)$ is immediately below $l(q')$. If $l(q)$ and $l(q')$ overlap, break ties arbitrarily. Analogously, add a directed edge $(\mathbf{v}(p, q), \mathbf{v}(p', q))$ if $l(p)$ is immediately to the left of $l(p')$. A path $P \subseteq \mathcal{G}^f$ is an ordered subset $\{(a_1, b_1), ..., (a_m, b_m)\}$ of edges where $b_i = a_{i+1}$, $\forall 1 \leq i < m$. We say P is a *feasible path* if $a_1 = \mathbf{v}(\pi_1, \sigma_1)$, $b_m = \mathbf{v}(\pi_n, \sigma_n)$, and $\mathbf{w}(a_i) = \mathbf{w}(b_i) = 1$ for all values of i. When a_1 and b_m are specified, P is a feasible path if P uses exclusively activated vertices. We say a FSG $\mathcal{G}^f$ is *st-reachable* if there exists a feasible path in $\mathcal{G}^f$ from $\mathbf{v}(\pi_1, \sigma_1)$ to $\mathbf{v}(\sigma_n, \pi_n)$. The following lemma naturally is derived from the properties of the freespace diagram.

▶ **Lemma 1.** *The FSG $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi, \sigma)$ is st-reachable if and only if $d_{\mathcal{F}}(\pi, \sigma) \leq \delta$.*

Proof. TOPROVE 0 ◀

Consider translating σ along $\vec{v}$. As σ translates, the FSG also changes, and we would like to know how these changes affect the *st*-reachability of $\mathcal{G}^f$. To do this, we track the following freespace events. See Figure 2 for visualizations of these row freespace events.

▶ **Definition 2.** *We define the following* row freespace events*. Column freespace events are defined analogously.*

1. *A* Vertex-edge (VE) event *is defined by a tuple $(p, \overline{\sigma_w})$, where p is either a vertex of π or a row critical point in row w.*
   - **a.** *Entering/leaving event: the corner point p enters or leaves the freespace.*
   - **b.** *Appearing/disappearing event: the grid line $l(p)$ appears or disappears as the critical point p appears or disappears.*
2. *A* Vertex-vertex-edge (VVE) event *is defined by a triplet $(p, q, \overline{\sigma_w})$, where p and q are row critical points in row w.*
   - **a.** *Overlapping event: two grid lines $l(p)$ and $l(q)$ overlap.*
   - **b.** *Separating event: two overlapping grid lines $l(p)$ and $l(q)$ no longer overlap.*

Let $\sigma + t$ be the curve by applying translation t to σ. For now, we assume that we know the ordered set $T = \{t_1, t_2, ..., t_m\}$ of freespace events (translations), where $t_i = \lambda_i \vec{v}$, $\lambda_i \in \mathbb{R}_{\geq 0}$, and $\lambda_i \leq \lambda_{i+1}$. We further assume that T is *complete*, which is defined as follows.

▶ **Definition 3.** *We say the ordered set $T = \{t_1, ..., t_m\}$ of freespace events is* complete *if $\mathcal{G}^f_\delta(\pi, \sigma + t_i)$ and $\mathcal{G}^f_\delta(\pi, \sigma + t_{i+1})$ differ by exactly one freespace event, for all $1 \leq i < m$.*

To update the freespace graph to reflect the state of the freespace diagram, we define the following freespace graph operations. Let $R(p)$ be the set of vertices defined by $l(p)$, and let $R(p)[i]$ denote the ith vertex, where $i \geq 1$.

▶ **Definition 4.** *We define the following* freespace graph operations*.*

- *Vertex operation: either activate or deactivate a vertex of $\mathcal{G}^f$.*
- *Row operation: either insert or delete a row of $\mathcal{G}^f$. Column operations are defined analogously.*

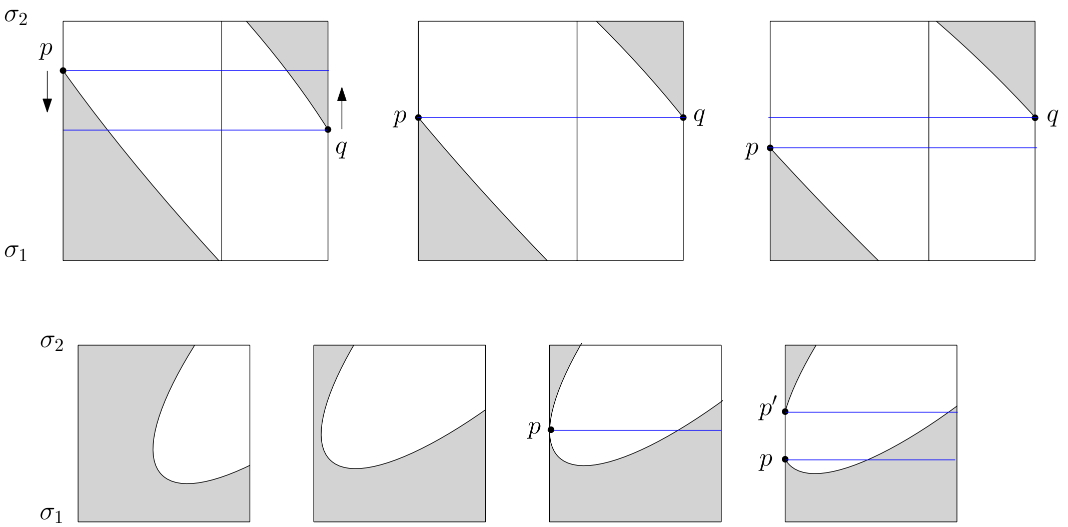

Figure 2 In the top row of figures, as the segment $\sigma_1\sigma_2$ translates, the critical points p and q moves down and up, respectively. As $l(p)$ and $l(q)$ (colored in blue) move, they overlap and then separate. In the bottom row, as $\sigma_1\sigma_2$ translates, a new critical point p appears, and then a new critical point p' appears.

- *To insert a row $R(p)$ of vertices between adjacent rows $R(p_a)$ and $R(p_b)$, add a vertex $\mathbf{v}(q, p)$ for every q, where q is either a critical point on a horizontal FSG boundary or a vertex of π. For every $1 \leq i < |R(p_a)|$,*
  1. *remove the edge $(R(p_b)[i], R(p_a)[i])$,*
  2. *add horizontal edge $(R(p)[i], R(p)[i + 1])$, and*
  3. *add vertical edges $(R(p_b)[i], R(p)[i])$ and $(R(p)[i], R(p_a)[i])$.*
- *To remove the row $R(p)$ of vertices, remove $R(p)$ and their adjacent edges. For every $1 \leq i \leq |R(p_a)|$, add edges $(R(p_b)[i], R(p_a)[i])$.*

We show that we can update the FSG $\mathcal{G}^f_\delta(\pi, \sigma + t_{i-1})$ to $\mathcal{G}^f_\delta(\pi, \sigma + t_i)$ using a constant number of freespace graph operations, plus processing time. For these updates, we will distinguish between a corner vertex operation and a boundary vertex operation.

▶ **Lemma 5.** *Let $\mathcal{G}^f_i = \mathcal{G}^f_\delta(\pi, \sigma + t_i)$. Given $\mathcal{G}^f_{i-1}$, to compute $\mathcal{G}^f_i$, it takes*

- *$\mathcal{O}(1)$ time and $\mathcal{O}(1)$ corner vertex operations if t_i is an entering/leaving event,*
- *$\mathcal{O}(1)$ time and $\mathcal{O}(1)$ boundary vertex operations if t_i is an overlapping or separating event,*
- *$\mathcal{O}(n)$ time plus $\mathcal{O}(1)$ row operations, if t_i is an appearing/disappearing event.*

Proof. TOPROVE 1 ◀

Using a naive implementation, in the worst case, a row operation can take $\Omega(n^2)$ time since there are $\Omega(n^2)$ vertical grid lines. Determining st-reachability takes $\Omega(n^4)$ time, since there are $\Omega(n^4)$ vertices in the FSG. Hence, using the FSG directly would be infeasible. In the next section, we fix these issues by defining an "equivalent" grid graph in which updates and queries can be done more efficiently than the naive implementation.

4 From freespace graph reachability to grid graph reachability

In this section, using the refined FSG $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi, \sigma)$, we define a *grid graph* $\mathcal{G}^g = \mathcal{G}^g_\delta(\pi, \sigma)$. We then show that $\mathcal{G}^f$ is st-reachable if and only if $\mathcal{G}^g$ is st-reachable. An advantage of the newly defined grid graph is that its number of vertices does not change under updates. We show that the structural changes implied by the FSG operations can be simulated by simply modifying the weights in the grid graph, without needing to add or remove vertices. These features of the grid graph allow us to use the offline dynamic grid reachability result in [14] to obtain a faster update and query time.

The grid graph $\mathcal{G}^g$ contains all vertices of the the freespace graph $\mathcal{G}^f$. Additionally, to maintain the same number of vertices in the grid graph under updates, we require a set of placeholder vertices, which we define as follows. Refer to Figure 3 for an illustration. Let m_i^r (resp. m_i^c) be the number of row critical points in the ith row (resp. column) of $\mathcal{D} = \mathcal{D}_\delta(\pi, \sigma)$. We define a family of ordered sets $\{H_1^r, ..., H_{n-1}^r\}$ of row placeholder points. Each set H_i^r contains exactly $2n - m_i^r$ points, and let $H_i^r[j]$ denote the jth point for $1 \le j \le 2n - m_i^r$. The family of ordered sets of column placeholder points $\{H_1^c, ..., H_{n-1}^c\}$ are analogously defined. For a row (resp. column) placeholder point h, the *placeholder line* $l(h)$ is horizontal (resp. vertical). Note that the placeholder points and lines are abstractly defined as they do not exist in $\mathcal{D}$.

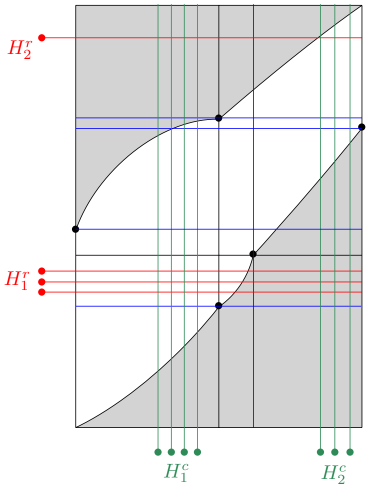

Figure 3 A visual illustration of the placeholder lines and their relative positions among the grid lines and the freespace diagram boundaries. The row (resp. column) placeholder points and lines are colored in red (resp. green). The grid lines are colored in blue.

In addition to vertices in $\mathcal{G}^f$, add a *placeholder vertex* $\mathbf{v}(p, q)$ to $\mathcal{G}^g$ where p (resp. q) is either a vertex of π (resp. σ), a critical point, or a placeholder point. The weight of a placeholder vertex on a boundary matches the weight of the adjacent corner vertex either directly above or to its right. Specifically, for every h in every H_i^r and every point p in the union of the vertices of π and row critical points. For $h \in H_j^r$, we set $\mathbf{w}(\mathbf{v}(p, h)) = \mathbf{w}(\mathbf{v}(\pi_i, \sigma_{j+1}))$ if $p = \pi_i$. Otherwise, we set $\mathbf{w}(\mathbf{v}(p, h)) = 1$. Analogously, for every $h \in H_i^c$, we set $\mathbf{w}(\mathbf{v}(h, q)) = \mathbf{w}(\mathbf{v}(\pi_{i+1}, \sigma_j))$ if $q = \sigma_j$. Otherwise, we set $\mathbf{w}(\mathbf{v}(h, q)) = 1$.

To define the edges in $\mathcal{G}^g$, we define the ordering of the placeholder lines among the grid lines and the FSD boundaries. The lowest placeholder line $l(H_i^r[1])$ lies above all grid lines $l(p)$, where p is a row critical point on the ith row. The placeholder line $l(H_i^r[j+1])$ is above $l(H_i^r[j])$. The FSD boundary $l(\sigma_{i+1})$ is above the highest placeholder line $l(H_i^r[2n - m_i^r])$. The ordering involving vertical placeholder lines is analogously defined. The set of placeholder lines defined by H_i^c are between $l(\pi_{i+1})$ and the rightmost grid line in the ith column.

For vertices in $\mathcal{G}^g$ defined by $l(p)$, add a directed edge $(\mathbf{v}(p, q), \mathbf{v}(p, q'))$ to $\mathcal{G}^g$ if $l(q)$ is immediately below $l(q')$ and add a directed edge $(\mathbf{v}(q,p), \mathbf{v}(q',p))$ to $\mathcal{G}^g$ if $l(q)$ is immediately to the left of $l(q')$. If $l(q)$ and $l(q')$ overlap, then break ties using the same ordering as in $\mathcal{G}^f$. Add additional diagonal directed edge $(\mathbf{v}(p,q), \mathbf{v}(p',q'))$ if p is immediately to the left of p' and q is immediately below q'. We say a vertex defined by two placeholder points is a placeholder vertex as well as an interior vertex.

Next, we check that our construction fits the grid graph definition of [14]. An $N \times N$ grid graph consists of vertices numbered from $(1,1)$ to (N,N) and edges from vertex (i,j) to each of vertices $(i+1,j)$, $(i,j+1)$, and $(i+1,j+1)$, where the weight of a vertex is either 1 or 0. With $\mathcal{G}^g_\delta(\pi,\sigma)$ defined, we prove the following. By construction, every of $n-1$ rows in $\mathcal{D}_\delta(\pi,\sigma)$ contains exactly $2n$ critical points and placeholder points combined. Together with n horizontal boundaries, there are $N = 2n \cdot (n-1) + n = 2n^2 - n$ horizontal lines. For the same reasons, there are N vertical lines. Clearly, $\mathcal{G}^g_\delta(\pi,\sigma)$ is an $N \times N$ grid graph.

Before proving that $\mathcal{G}^f_\delta(\pi,\sigma)$ and $\mathcal{G}^g_\delta(\pi,\sigma)$ are equivalent in terms of st-reachability, we first show that a feasible path $P \subseteq \mathcal{G}^g$ has several desired properties. By the monotonicity of a feasible path and the convexity of the freespace in a cell, we observe the following.

▶ **Observation 6.** *Let P be a feasible path in $\mathcal{G}^g$. If P contains a corner vertex $\mathbf{v}(\pi_i,\sigma_j)$, then let $l(p_l)$ and $l(p_r)$ be the first grid lines that are not placeholder lines to the left and right of $l(\pi_i)$, respectively. Similarly, let $l(p_a)$ and $l(p_b)$ be the first grid line above and below $l(\sigma_j)$ respectively. Then, we have either $\mathbf{w}(\mathbf{v}(p_l,\sigma_j)) = 1$ or $\mathbf{w}(\mathbf{v}(\pi_i,p_b)) = 1$. Analogously, we have either $\mathbf{w}(\mathbf{v}(\pi_i,p_a)) = 1$ or $\mathbf{w}(\mathbf{v}(p_r,\sigma_j)) = 1$.*

Proof. TOPROVE 2 ◀

Due to the properties of $\mathcal{G}^g$, a diagonal edge in a path P can be replaced by using the rectilinear edges in the same "cube". Furthermore, if the final vertex b of a subpath $P_j \subseteq P$ is a placeholder vertex, we can transform P_j such that it ends at a non-placeholder vertex. We have the following lemma.

▶ **Lemma 7.** *If there is a feasible path P in a grid graph $\mathcal{G}^g$, there is a feasible path P' in $\mathcal{G}^g$ with the following properties.*

1. *P' does not contain diagonal edges.*
2. *For every $1 \leq j \leq n$, the first vertex of P' partly defined by σ_j is not a placeholder vertex.*

Proof. TOPROVE 3 ◀

We next observe that if there is a path in $\mathcal{G}^g$ that does not use a placeholder vertex, the path also exists in $\mathcal{G}^f$. Indeed, excluding the placeholder lines, $\mathcal{G}^g$ and $\mathcal{G}^f$ use the same set of grid lines and FSG boundaries, and the same ordering.

▶ **Observation 8.** *If there is a path P in $\mathcal{G}^g = \mathcal{G}^g_\delta(\pi,\sigma)$ such that P does not use any placeholder vertices or diagonal edges, then P also exists in $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi,\sigma)$.*

To demonstrate that $\mathcal{G}^f$ and $\mathcal{G}^g$ are equivalent with respect to st-reachability, we first note that any feasible path in $\mathcal{G}^g$ corresponds to a feasible path in $\mathcal{G}^f$. Specifically, given a subpath $P \subseteq \mathcal{G}^g$ that traverses the "strip" defined by a single set of placeholder points, we can always construct a corresponding subpath $Q \subseteq \mathcal{G}^f$ such that Q starts and ends at the same vertices as P. We have Lemma 9 and 10.

▶ **Lemma 9.** *Let P be a path in $\mathcal{G}^g = \mathcal{G}^g_\delta(\pi,\sigma)$. Let P start at a non-placeholder vertex $\mathbf{v}(p,\sigma_j)$. Let $(\mathbf{v}(p',q'), \mathbf{v}(p',H^r_j[1]))$ be the last edge of P, where $q' \neq H^r_j[1]$. There exists a path Q from $\mathbf{v}(p,\sigma_j)$ to $\mathbf{v}(p',q')$ in $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi,\sigma)$.*

Proof. TOPROVE 4 ◀

▶ **Lemma 10.** *Let P be a path in $\mathcal{G}^g = \mathcal{G}^g_\delta(\pi, \sigma)$. Let $(\mathbf{v}(p, q), \mathbf{v}(p, H^r_j[1]))$ be the first edge of P, and let P ends at a non-placeholder vertex $\mathbf{v}(p', \sigma_{j+1})$. There exists a path Q from $\mathbf{v}(p, q)$ to $\mathbf{v}(p', \sigma_{j+1})$ in $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi, \sigma)$.*

Proof. TOPROVE 5 ◀

We are finally ready to show that $\mathcal{G}^f$ and $\mathcal{G}^g$ are equivalent in terms of st-reachability.

▶ **Lemma 11.** *There exists a feasible path P^f in the freespace graph $\mathcal{G}^f_\delta(\pi, \sigma)$ if and only if there exists a feasible path P^g in the grid graph $\mathcal{G}^g_\delta(\pi, \sigma)$.*

Proof. TOPROVE 6 ◀

Now, we show that freespace graph operations can be implemented in the grid graph efficiently.

▶ **Lemma 12.** *Let $\mathcal{G}^g_i$ be the associated grid graph of the freespace graph $\mathcal{G}^f_i$. Let u be a freespace graph operation that updates $\mathcal{G}^f_1$ to $\mathcal{G}^f_2$. To update $\mathcal{G}^g_1$ to $\mathcal{G}^g_2$, it is sufficient to update the weights of at most:*

- *$\mathcal{O}(n)$ vertices if u is a corner vertex operation,*
- *$\mathcal{O}(1)$ vertices if u is a boundary vertex operation, or*
- *$\mathcal{O}(n)$ vertices if u is a row operation.*

Proof. TOPROVE 7 ◀

Given $\mathcal{G}^g_{i-1} = \mathcal{G}^g_\delta(\pi, \sigma + t_{i-1})$ and event t_i, we can now transform $\mathcal{G}^g_{i-1}$ to $\mathcal{G}^g_i$. Specifically, in Lemma 5, we have shown that if t_i is a VE event, it takes $\mathcal{O}(n)$ time plus $\mathcal{O}(1)$ corner vertex operations or row operations. If t_i is a VVE event, it takes $\mathcal{O}(1)$ time plus $\mathcal{O}(1)$ boundary vertex operations. By combining Lemma 5 and Lemma 12, we can bound the number of vertex weight changes for each freespace event type.

▶ **Lemma 13.** *Given $\mathcal{G}^g_\delta(\pi, \sigma + t_{i-1})$ and the next event t_i, to compute $\mathcal{G}^g_\delta(\pi, \sigma + t_i)$, it takes*

- *$\mathcal{O}(n)$ vertex weight changes if t_i is a VE event, or*
- *$\mathcal{O}(1)$ vertex weight changes if t_i is a VVE event.*

We can now summarize Sections 3 and 4 and state the main lemma of this section. In Lemma 1, we have shown that the Fréchet distance $d_{\mathcal{F}}(\pi, \sigma)$ is at most δ if and only if the refined freespace graph $\mathcal{G}^f = \mathcal{G}^f_\delta(\pi, \sigma)$ is st-reachable. In Section 4, for each $\mathcal{G}^f$, we have defined an associate grid graph $\mathcal{G}^g = \mathcal{G}^g_\delta(\pi, \sigma)$. In Lemma 11, we have shown that $\mathcal{G}^g$ is st-reachable if and only if $\mathcal{G}^f$ is st-reachable. Combining the above with Lemma 13, we have the following.

▶ **Lemma 14.** *Let $T = \{t_1, ..., t_m\}$ be a complete set of freespace events containing exclusively m_{vve} VVE events and m_{ve} VE events. Let $N = 2n^2 - n$, and let $T_u(N)$ (resp. $T_q(N)$) be the time complexity to update (resp. query st-reachability) in an $N \times N$ grid graph. It takes*

$$\mathcal{O}(N^2 + (m_{ve} \cdot n + m_{vve}) \cdot T_u(N) + m \cdot T_q(N))$$

time to determine if there exists $t_i \in T$ such that $d_{\mathcal{F}}(\pi, \sigma + t_i) \leq \delta$.

Using the results by Alt, Knauer and Wenk [4], we can build an arrangement in the translation space. Using this arrangement, we can compute a set of complete events (translations) T containing exclusively $\mathcal{O}(n^6)$ VVE events and $\mathcal{O}(n^5)$ VE events. They have shown that it is sufficient to consider only translations in T to determine if there is a translation t such that $d_{\mathcal{F}}(\pi, \sigma + t) \leq \delta$.

We analyze the running time. In total, we require $\mathcal{O}(n^6)$ vertex updates for $\mathcal{G}^g$. Using the result of Bringmann, Künnemann and Nusser [14], we can update a vertex and perform st-reachability queries in amortized $\mathcal{O}(N^{2/3} \cdot \log^2 N) = \mathcal{O}(n^{4/3} \cdot \log^2 n)$ time. We obtain the following theorem. We defer the proof of Theorem 15, and its generalization, to Section 5.

▶ **Theorem 15.** *The Fréchet distance under translation in $\mathbb{R}^2$ can be decided in $\mathcal{O}(n^{7+\frac{1}{3}} \log^2 n)$ time.*

5 Fréchet distance under transformation

In this section, we consider a class $\mathcal{T}$ of transformations that is *rationally parameterized* or *rationally represented with k degrees of freedom* as defined by Wenk [54].

▶ **Definition 16** ([54, Definition 25])**.** *Let $1 \leq k \leq d^2+d$, and let $p_1, \ldots, p_{d^2+d}, q_1, \ldots, q_{d^2+d} \in \mathbb{R}[X_1, \ldots, X_k]$ be $2(d^2+d)$ polynomials of constant degree in k variables, such that $q_i(x) \neq 0$ for all $1 \leq i \leq d^2+d$ and for all $x \in \mathbb{R}^k$. Let $r_i := p_i/q_i$ for all $1 \leq i \leq d^2+d$, such that $r_i(x) := p_i(x)/q_i(x)$ for all $x \in \mathbb{R}^k$. If*

$$\mathcal{T} = \left\{ \left(\begin{pmatrix} r_1(x) & \ldots & r_d(x) \\ r_{d+1}(x) & \ldots & r_{2d}(x) \\ \vdots & \ddots & \vdots \\ r_{d^2-d+1}(x) & \ldots & r_{d^2}(x) \end{pmatrix}, \begin{pmatrix} r_{d^2+1}(x) \\ r_{d^2+2}(x) \\ \vdots \\ r_{d^2+d}(x) \end{pmatrix} \right) \;\middle|\; x \in \mathbb{R}^k \right\},$$

then we call $\mathcal{T}$ rationally parameterized or rationally represented with k degrees of freedom (dof). $\mathbb{R}^k$ is called the parameter space of $\mathcal{T}$.

Let $\mathbb{R}^k$ be the parameter space of $\mathcal{T}$. For $x \in \mathbb{R}^k$, let τ_x denote the transformation defined by the k-tuple x of parameters. Let π and σ be two d-dimensional polygonal curves. Let $\tau_x(\sigma)$ be the resulting curve by applying the transformation τ_x to σ.

In $\mathbb{R}^k$, a *vertex-vertex-edge (VVE) critical transformation* $\mathcal{T}_\delta^{vve}(\pi_i, \pi_j, \overline{\sigma_w})$ is the union of every point $x \in \mathbb{R}^k$ such that the segment $\tau_x(\overline{\sigma_w})$ lies on the intersection of the boundary of the d-spheres centered at π_i and π_j, respectively, and it is formally defined as follows.

$$\mathcal{T}_\delta^{vve}(\pi_i, \pi_j, \overline{\sigma_w}) = \{x \in \mathbb{R}^k \mid \exists z \in \overline{\sigma_w}, \|\tau_x(z) - \pi_i\| = \|\tau_x(z) - \pi_j\| = \delta\}$$

Analogously, a *vertex-edge (VE) critical transformation* $\mathcal{T}_\delta^{ve}(\pi_i, \overline{\sigma_w})$ is the union of every point $x \in \mathbb{R}^k$ such that the segment $\tau_x(\overline{\sigma_w})$ lie on the boundary of the d-ball centered at π_i, and it is formally defined as follows.

$$\mathcal{T}_\delta^{ve}(\pi_i, \overline{\sigma_w}) = \{x \in \mathbb{R}^k \mid \exists z \in \overline{\sigma_w}, \|\tau_x(z) - \pi_i\| = \delta\}$$

Every critical transformation is a semi-algebraic set. Using $\mathcal{O}(n^3)$ VVE critical transformations and $\mathcal{O}(n^2)$ VE critical transformations, Wenk [54, Proof of Theorem 8] showed that they can build an arrangement $\mathcal{A}_\delta$ in $\mathcal{O}(n^{3k})$ time using $\mathcal{O}(n^{3k})$ space. Furthermore, $\mathcal{A}_\delta$ contains at most $\mathcal{O}(n^{3k})$ k'-dimensional faces for $0 \leq k' \leq k-1$. Let $\tau_F = \tau_x$ be the transformation that is represented by an arbitrary parameter $x \in F$. In [54, Lemma 24], Wenk showed that if there exists some transformation τ such that $d_{\mathcal{F}}(\pi, \tau(\sigma)) < \delta$, then there exists some k'-dimensional face $F \in \mathcal{A}_\delta$ such that for any τ_F, $d_{\mathcal{F}}(\pi, \tau_F(\sigma)) \leq \delta$. Their results are summarized as follows.

▶ **Fact 17.** *Given a pair of polygonal curve π and σ, a real number $\delta \geq 0$, and a class $\mathcal{T}$ of transformations that is rationally represented with k degrees of freedom, one can build an arrangement $\mathcal{A}_\delta = \mathcal{A}_\delta(\pi, \sigma, \mathcal{T})$ using at most $\mathcal{O}(n^3)$ VVE critical transformations and $\mathcal{O}(n^2)$ VE critical transformations. The arrangement $\mathcal{A}_\delta$ has in total $\mathcal{O}(n^{3k})$ complexity, and it can be constructed in $\mathcal{O}(n^{3k})$ time using $\mathcal{O}(n^{3k})$ space. To determine if there exists a transformation $\tau \in \mathcal{T}$ such that $d_{\mathcal{F}}(\pi, \tau(\sigma)) \leq \delta$, it is sufficient to check exactly one transformation τ_F for every k'-dimensional face $F \in \mathcal{A}_\delta$, where $0 \leq k' \leq k - 1$.*

Once the arrangement $\mathcal{A}_\delta$ is constructed, the remainder of the previous algorithm in Wenk [54] is straightforward. For every face $F \in \mathcal{A}_\delta$, sample a point $x \in F$, and determine if $d_{\mathcal{F}}(\pi, \tau_x(\sigma)) \leq \delta$ using classic algorithms (Alt and Godau [2] for example). In total, this takes $\tilde{\mathcal{O}}(n^{3k} \cdot n^2)$ time.

To obtain a running time improvement, we use a similar approach to previous sections. We generate a complete set of events as follows. Initialize an empty graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For every face $F \in \mathcal{A}_\delta$, add a vertex v_F to $\mathcal{V}$. For every two adjacent faces F and F', add an edge $(v_F, v_{F'})$ to $\mathcal{E}$. For each vertex v_F, record a transformation τ_F. Next, we compute a complete set of events using $\mathcal{G}$. Initialize an empty set of events T. Then, use a DFS to compute a spanning tree of $\mathcal{G}$, and perform a Euler tour over the spanning tree starting from an arbitrary vertex. For each directed edge $e = (v_{F'}, v_F)$ in the tour, we add an event only if we enter or leave a critical transformation. More specifically, let $B(F)$ be the set of critical transformations adjacent to face F. If $B(F) \setminus B(F') = \{\mathcal{T}_c\}$, we say e *traverses onto* the critical transformation $\mathcal{T}_c$ via F. If $B(F') \setminus B(F) = \{\mathcal{T}_c\}$, we say e *traverses out of* the critical transformation $\mathcal{T}_c$ via F.

Let $D(p)$ be the d-sphere of radius δ centered at p. Depending on the cases where e traverse onto or out of a VVE or VE critical transformation, we add the respective freespace events defined in Definition 2.

1. If $\mathcal{T}_c$ is a VVE critical transformation $\mathcal{T}_\delta^{vve} = \mathcal{T}_\delta^{vve}(\pi_i, \pi_j, \overline{\sigma_w})$, we compute $p = D(\pi_i) \cap \tau_F(\overline{\sigma_w})$ and $q = D(\pi_j) \cap \tau_F(\overline{\sigma_w})$, and append an event t defined by $(p, q, \overline{\sigma_w})$ to T. If e traverses onto $\mathcal{T}_c$, t is an overlapping event. If e traverses out of $\mathcal{T}_\delta^{vve}$, t is a separating event.
2. If $\mathcal{T}_c$ is a VE critical transformation $\mathcal{T}_\delta^{ve} = \mathcal{T}_\delta^{ve}(\pi_i, \overline{\sigma_w})$, we compute $p = D(\pi_i) \cap \overline{\sigma_w}$, and we append an event t represented by $(p, \overline{\sigma_w})$ to T. If e traverses onto $\mathcal{T}_c$, t is an entering event if $p = \pi_i$, or an appearing event if $p \neq \pi_i$. Analogously, if e traverses out of $\mathcal{T}_c$, t_e is a leaving event if $p = \pi_i$, or a disappearing event if $p \neq \pi_i$.

We say an event t is associated with the critical transformation $\mathcal{T}_c$ and the edge e, if t is computed from an edge e traversing onto or out of $\mathcal{T}_c$. We show that T has two desired property.

▶ **Lemma 18.** *Given the arrangement $\mathcal{A}_\delta = \mathcal{A}_\delta(\pi, \sigma, \mathcal{T})$, one can compute a complete set $T = \{t_1, ..., t_{\mathcal{O}(n^{3k})}\}$ of freespace events in $\mathcal{O}(n^{3k})$ time with the following properties.*

1. *Every face in $\mathcal{A}_\delta$ is associated with at least one event in T.*
2. *For each event t_i associated with edge $(v_F, v_{F'}) \in \mathcal{E}$, $\mathcal{G}_\delta^f(\pi, \tau_F(\sigma))$ and $\mathcal{G}_\delta^f(\pi, \tau_{F'}(\sigma))$ differ by exactly one freespace event.*

Proof. TOPROVE 8 ◀

We also observe that fewer faces are adjacent to VE critical transformations than to VVE critical transformations.

▶ **Observation 19.** *In the arrangement $\mathcal{A}_\delta = \mathcal{A}_\delta(\pi, \sigma, \mathcal{T})$, there are at most $\mathcal{O}(n^{3k})$ faces adjacent to VVE critical transformations, and $\mathcal{O}(n^{3k-1})$ faces adjacent to VE critical transformations.*

Proof. TOPROVE 9 ◀

To obtain fast updates and queries in the grid graph, we use the offline dynamic grid reachability result by [14]. The problem is defined as follows. We start from a directed $N \times N$-grid graph, and we are given a set $\{u_1, ..., u_U\}$ of updates such that each update u_i is to either activate or deactivate a vertex. For $1 \leq i \leq U$, the goal is to compute after each update u_i whether there is a feasible path from vertex $\mathbf{v}(1,1)$ to $\mathbf{v}(N,N)$. Their result is as follows.

▶ **Fact 20** (Frechet distance under translation[14, Theorem 3.4]). *Offline dynamic grid reachability can be solved in time $\mathcal{O}(N^2 + UN^{2/3}\log^2 N)$.*

We summarize the results in this paper, which we will then combine to obtain a faster algorithm for computing the Fréchet distance under rationally parameterized transformations. We take as inputs a real number $\delta \geq 0$, a class $\mathcal{T}$ of transformations rationally represented with k degrees of freedom, and a pair of polygonal curves π and σ, each with n vertices. Our goal is to determine if there exists a transformation $\tau \in \mathcal{T}$ such that $d_{\mathcal{F}}(\pi, \tau(\sigma)) \leq \delta$.

To do this, Wenk [54] has shown that it is sufficient to construct an arrangement $\mathcal{A}_\delta$ in the parameter space $\mathbb{R}^k$ and sample a transformation from each face. In Lemma 18, we have shown that we can traverse this arrangement and generate a complete set of $\mathcal{O}(n^{3k})$ freespace events in $\mathcal{O}(n^{3k})$ time. In Section 3 and 4, we explained our ideas using translation. However, all results revolve around handling a complete set of freespace events, and hence apply to more general settings of transformations. We can therefore plug in the number of updates to Lemma 14. Our algorithm takes $\mathcal{O}(n^4 + n^{3k} \cdot (T_u(n^2) + T_q(n^2))$ time, where $T_u(n^2)$ (resp. $T_q(n^2)$) is the time complexity to update a vertex (resp. query st-reachability) in an $\mathcal{O}(n^2) \times \mathcal{O}(n^2)$-grid graph. By Fact 20, $T_u(n^2) + T_q(n^2)$ takes amortized $\mathcal{O}(n^{4/3}\log^2 n)$ time. For classes of rationally parameterized transformations with at least one degree of freedom, $k \geq 1$. We state our final result.

▶ **Theorem 21.** *The Fréchet distance under transformations rationally represented with k degrees of freedom can be decided in $\mathcal{O}(n^{3k+4/3}\log^2 n)$ time.*

6 Conclusion

Our algorithm provides the first progress in over 20 years for computing the (continuous) Fréchet distance under transformations. The running time comes from traversing an arrangement in the space of transformations, performing an update to a dynamic grid graph data structure in each step.

We conclude with open questions. Can the update time be reduced? Improving the update time for the data structure of [14] would directly improve our running time. But it may also be possible to tailor the data structure for our setting. For instance, we could prune the lower levels of the data structure, since reachability in the inside of free space cells does not carry any information.

Can we prove a non-trivial conditional lower bound for computing the Fréchet distance under translations? The complexity of the arrangement, $\Omega(n^6)$, would seem like a natural lower bound. However, even transferring the $n^{4-o(1)}$ conditional lower bound for the discrete Fréchet distance under translation [14] to the (continuous) Fréchet distance seems difficult, since the lower bound construction crucially relies on the fact that the discrete Fréchet traversal can only stay on the vertices and not on the edges.

References

1 Helmut Alt. The computational geometry of comparing shapes. In Susanne Albers, Helmut Alt, and Stefan Näher, editors, *Efficient Algorithms, Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*, volume 5760 of *Lecture Notes in Computer Science*, pages 235–248. Springer, 2009. doi:10.1007/978-3-642-03456-5_16.

2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 05:75–91, 1995. URL: https://www.worldscientific.com/doi/abs/10.1142/S0218195995000064, doi:10.1142/S0218195995000064.

3 Helmut Alt and Leonidas J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, chapter 3, pages 121–153. North Holland / Elsevier, 2000. doi:10.1016/B978-044482537-7/50004-8.

4 Helmut Alt, Christian Knauer, and Carola Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 63–74. Springer, 2001. doi:10.1007/3-540-44693-1_6.

5 Helmut Alt, Ludmila Scharf, and Sven Scholz. Probabilistic matching and resemblance evaluation of shapes in trademark images. In *Proc. 6th ACM International Conference on Image and Video Retrieval (CIVR)*, pages 533–540. ACM, 2007. doi:10.1145/1282280.1282357.

6 Boris Aronov, Tsuri Farhana, Matthew J. Katz, and Indu Ramesh. Discrete Fréchet distance oracles. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.SoCG.2024.10, doi:10.4230/LIPICS.SOCG.2024.10.

7 Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. A faster algorithm for the discrete Fréchet distance under translation. *CoRR*, abs/1501.03724, 2015. URL: http://arxiv.org/abs/1501.03724, arXiv:1501.03724.

8 Julian Baldus and Karl Bringmann. A fast implementation of near neighbors queries for Fréchet distance (GIS Cup). In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 99:1–99:4, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3139958.3140062, doi:10.1145/3139958.3140062.

9 Lotte Blank, Jacobus Conradi, Anne Driemel, Benedikt Kolbe, André Nusser, and Marena Richter. Transforming dogs on the line: On the Fréchet distance under translation or scaling in 1D. Under submission, January 2025.

10 Lotte Blank and Anne Driemel. A faster algorithm for the Fréchet distance in 1d for the imbalanced case. In *Proc. 32nd Annual European Symposium on Algorithms (ESA)*, volume 308 of *LIPIcs*, pages 28:1–28:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.ESA.2024.28, doi:10.4230/LIPICS.ESA.2024.28.

11 Karl Bringmann and Bhaskar Ray Chaudhury. Polyline simplification has cubic complexity. *J. Comput. Geom.*, 11(2):94–130, 2020. URL: https://doi.org/10.20382/jocg.v11i2a5, doi:10.20382/JOCG.V11I2A5.

12 Karl Bringmann, Anne Driemel, André Nusser, and Ioannis Psarros. Tight bounds for approximate near neighbor searching for time series under the Fréchet distance. In *Proc.*

ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 517–550. SIAM, 2022. doi:10.1137/1.9781611977073.25.

13 Karl Bringmann, Marvin Künnemann, and André Nusser. When lipschitz walks your dog: Algorithm engineering of the discrete fréchet distance under translation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 25:1–25:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. URL: https://doi.org/10.4230/LIPIcs.ESA.2020.25, doi:10.4230/LIPICS.ESA.2020.25.

14 Karl Bringmann, Marvin Künnemann, and André Nusser. Discrete Fréchet distance under translation: Conditional hardness and an improved algorithm. *ACM Trans. Algorithms*, 17(3):25:1–25:42, 2021. doi:10.1145/3460656.

15 Karl Bringmann, Marvin Künnemann, and André Nusser. Walking the dog fast in practice: Algorithm engineering of the fréchet distance. *J. Comput. Geom.*, 12(1):70–108, 2021. URL: https://doi.org/10.20382/jocg.v12i1a4, doi:10.20382/JOCG.V12I1A4.

16 Frederik Brüning, Jacobus Conradi, and Anne Driemel. Faster approximate covering of subcurves under the Fréchet distance. In *Proc. 30th Annual European Symposium on Algorithms (ESA)*, volume 244 of *LIPIcs*, pages 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.ESA.2022.28, doi:10.4230/LIPICS.ESA.2022.28.

17 Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Aleksandr Popov, and Sampson Wong. Map-matching queries under Fréchet distance on low-density spanners. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.SoCG.2024.27, doi:10.4230/LIPICS.SOCG.2024.27.

18 Kevin Buchin, Yago Diez, Tom van Diggelen, and Wouter Meulemans. Efficient trajectory queries under the Fréchet distance (GIS Cup). In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 101:1–101:4, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3139958.3140064, doi:10.1145/3139958.3140064.

19 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *Proc. 30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2922–2938. SIAM, 2019. doi:10.1137/1.9781611975482.181.

20 Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. klcluster: Center-based clustering of trajectories. In Farnoush Banaei Kashani, Goce Trajcevski, Ralf Hartmut Güting, Lars Kulik, and Shawn D. Newsam, editors, *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019, Chicago, IL, USA, November 5-8, 2019*, pages 496–499. ACM, 2019. doi:10.1145/3347146.3359111.

21 Kevin Buchin, Chenglin Fan, Maarten Löffler, Aleksandr Popov, Benjamin Raichel, and Marcel Roeloffzen. Fréchet distance for uncertain curves. *ACM Trans. Algorithms*, 19(3):29:1–29:47, 2023. doi:10.1145/3597640.

22 Kevin Buchin, Brittany Terese Fasy, Erfan Hosseini Sereshgi, and Carola Wenk. On length-sensitive Fréchet similarity. In *Proc. 18th International Symposium on Algorithms and Data Structures (WADS)*, volume 14079 of *Lecture Notes in Computer Science*, pages 208–231. Springer, 2023. doi:10.1007/978-3-031-38906-1_15.

23 Kevin Buchin, Maarten Löffler, Tim Ophelders, Aleksandr Popov, Jérôme Urhausen, and Kevin Verbeek. Computing the Fréchet distance between uncertain curves in one dimension. *Comput. Geom.*, 109:101923, 2023. URL: https://doi.org/10.1016/j.comgeo.2022.101923, doi:10.1016/J.COMGEO.2022.101923.

24 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 ACM-SIAM*

Symposium on Discrete Algorithms (SODA), pages 2887–2901. SIAM, 2019. doi:10.1137/1.9781611975482.179.

25 Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k, ℓ)-median clustering for polygonal curves. *ACM Trans. Algorithms*, 19(1):4:1–4:32, 2023. doi:10.1145/3559764.

26 Maike Buchin, Ivor van der Hoog, Tim Ophelders, Lena Schlipf, Rodrigo I. Silveira, and Frank Staals. Efficient Fréchet distance queries for segments. In *Proc. 30th Annual European Symposium on Algorithms (ESA)*, volume 244 of *LIPIcs*, pages 29:1–29:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.ESA.2022.29, doi:10.4230/LIPICS.ESA.2022.29.

27 Siu-Wing Cheng and Haoqiang Huang. Curve simplification and clustering under Fréchet distance. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1414–1432. SIAM, 2023. URL: https://doi.org/10.1137/1.9781611977554.ch51, doi:10.1137/1.9781611977554.CH51.

28 Siu-Wing Cheng and Haoqiang Huang. Fréchet distance in subquadratic time. *CoRR*, abs/2407.05231, 2024. To appear at SODA 2025. URL: https://doi.org/10.48550/arXiv.2407.05231, arXiv:2407.05231, doi:10.48550/ARXIV.2407.05231.

29 Siu-Wing Cheng and Haoqiang Huang. Solving Fréchet distance problems by algebraic geometric methods. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4502–4513. SIAM, 2024. doi:10.1137/1.9781611977912.158.

30 Connor Colombe and Kyle Fox. Approximating the (continuous) Fréchet distance. In *Proc. 37th International Symposium on Computational Geometry (SoCG)*, volume 189 of *LIPIcs*, pages 26:1–26:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: https://doi.org/10.4230/LIPIcs.SoCG.2021.26, doi:10.4230/LIPICS.SOCG.2021.26.

31 Krzysztof Diks and Piotr Sankowski. Dynamic plane transitive closure. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, volume 4698 of *Lecture Notes in Computer Science*, pages 594–604. Springer, 2007. doi:10.1007/978-3-540-75520-3_53.

32 Anne Driemel, Ivor van der Hoog, and Eva Rotenberg. On the discrete Fréchet distance in a graph. In *Proc. 38th International Symposium on Computational Geometry, (SoCG)*, volume 224 of *LIPIcs*, pages 36:1–36:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.SoCG.2022.36, doi:10.4230/LIPICS.SOCG.2022.36.

33 Fabian Dütsch and Jan Vahrenhold. A filter-and-refinement-algorithm for range queries based on the Fréchet distance (GIS Cup). In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 100:1–100:4, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3139958.3140063, doi:10.1145/3139958.3140063.

34 Alon Efrat, Piotr Indyk, and Suresh Venkatasubramanian. Pattern matching for sets of segments. *Algorithmica*, 40(3):147–160, 2004. URL: https://doi.org/10.1007/s00453-004-1089-y, doi:10.1007/S00453-004-1089-Y.

35 Chenglin Fan and Benjamin Raichel. Computing the Fréchet gap distance. *Discrete & Computational Geometry*, 65(4):1244–1274, June 2021. doi:10.1007/s00454-020-00224-w.

36 Arnold Filtser and Omrit Filtser. Static and streaming data structures for Fréchet distance queries. *ACM Trans. Algorithms*, 19(4):39:1–39:36, 2023. doi:10.1145/3610227.

37 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate nearest neighbor for curves: Simple, efficient, and deterministic. *Algorithmica*, 85(5):1490–1519, 2023. URL: https://doi.org/10.1007/s00453-022-01080-1, doi:10.1007/S00453-022-01080-1.

38 Omrit Filtser, Mayank Goswami, Joseph S. B. Mitchell, and Valentin Polishchuk. On flipping the Fréchet distance. In *Proc. 14th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 251 of *LIPIcs*, pages 51:1–51:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. URL: https://doi.org/10.4230/LIPIcs.ITCS.2023.51, doi:10.4230/LIPICS.ITCS.2023.51.

39 Emily Fox, Amir Nayyeri, Jonathan James Perry, and Benjamin Raichel. Fréchet edit distance. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: `https://doi.org/10.4230/LIPIcs.SoCG.2024.58`, `doi:10.4230/LIPICS.SOCG.2024.58`.

40 Joachim Gudmundsson, Martin P. Seybold, and Sampson Wong. Map matching queries on realistic input graphs under the Fréchet distance. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1464–1492. SIAM, 2023. URL: `https://doi.org/10.1137/1.9781611977554.ch53`, `doi:10.1137/1.9781611977554.CH53`.

41 Joachim Gudmundsson, André van Renssen, Zeinab Saeidi, and Sampson Wong. Translation invariant Fréchet distance queries. *Algorithmica*, 83(11):3514–3533, 2021. URL: `https://doi.org/10.1007/s00453-021-00865-0`, `doi:10.1007/S00453-021-00865-0`.

42 Joachim Gudmundsson and Sampson Wong. Cubic upper and lower bounds for subtrajectory clustering under the continuous Fréchet distance. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 173–189. SIAM, 2022. `doi:10.1137/1.9781611977073.9`.

43 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure-structure alignment with discrete Fréchet distance. In *Proc. 5th Asia-Pacific Bioinformatics Conference (APBC)*, volume 5 of *Advances in Bioinformatics and Computational Biology*, pages 131–141. Imperial College Press, 2007. URL: `http://www.comp.nus.edu.sg/%7Ewongls/psZ/apbc2007/apbc162a.pdf`.

44 Adam Karczmarz and Marcin Smulewicz. Fully Dynamic Strongly Connected Components in Planar Digraphs. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 95:1–95:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2024.95`, `doi:10.4230/LIPIcs.ICALP.2024.95`.

45 Christian Knauer. *Algorithms for Comparing Geometric Patterns*. PhD thesis, Free University Berlin, 2002. URL: `http://dx.doi.org/10.17169/refubium-16312`.

46 Abhinandan Nath and Erin Taylor. k-median clustering under discrete Fréchet and hausdorff distances. In *Proc. 36th International Symposium on Computational Geometry (SoCG)*, volume 164 of *LIPIcs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. URL: `https://doi.org/10.4230/LIPIcs.SoCG.2020.58`, `doi:10.4230/LIPICS.SOCG.2020.58`.

47 E. Sriraghavendra, K. Karthik, and Chiranjib Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Proc. 9th International Conference on Document Analysis and Recognition (ICDAR)*, pages 461–465. IEEE Computer Society, 2007. `doi:10.1109/ICDAR.2007.4378752`.

48 Sairam Subramanian. A fully dynamic data structure for reachability in planar digraphs. In Thomas Lengauer, editor, *Algorithms - ESA '93, First Annual European Symposium, Bad Honnef, Germany, September 30 - October 2, 1993, Proceedings*, volume 726 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 1993. `doi:10.1007/3-540-57273-2\_72`.

49 Mees van de Kerkhof, Irina Kostitsyna, Maarten Löffler, Majid Mirzanezhad, and Carola Wenk. Global curve simplification. In *Proc. 27th Annual European Symposium on Algorithms (ESA)*, volume 144 of *LIPIcs*, pages 67:1–67:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: `https://doi.org/10.4230/LIPIcs.ESA.2019.67`, `doi:10.4230/LIPICS.ESA.2019.67`.

50 Thijs van der Horst and Tim Ophelders. Faster Fréchet distance approximation through truncated smoothing. In *Proc. 40th International Symposium on Computational Geometry (SoCG)*, volume 293 of *LIPIcs*, pages 63:1–63:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: `https://doi.org/10.4230/LIPIcs.SoCG.2024.63`, `doi:10.4230/LIPICS.SOCG.2024.63`.

51 Thijs van der Horst, Marc J. van Kreveld, Tim Ophelders, and Bettina Speckmann. A subquadratic n^ϵ-approximation for the continuous Fréchet distance. In *Proc. ACM-SIAM*

Symposium on Discrete Algorithms (SODA), pages 1759–1776. SIAM, 2023. URL: https://doi.org/10.1137/1.9781611977554.ch67, doi:10.1137/1.9781611977554.CH67.

52 Marc J. van Kreveld, Maarten Löffler, and Lionov Wiratma. On optimal polyline simplification using the Hausdorff and Fréchet distance. *J. Comput. Geom.*, 11(1):1–25, 2020. URL: https://doi.org/10.20382/jocg.v11i1a1, doi:10.20382/JOCG.V11I1A1.

53 Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proc. International Conference on Shape Modeling and Applications (SMI)*, page 188. IEEE Computer Society, 2001. doi:10.1109/SMA.2001.923389.

54 Carola Wenk. *Shape Matching in Higher Dimensions*. PhD thesis, Free University Berlin, 2003. URL: http://dx.doi.org/10.17169/refubium-8310.