

Dynamic Time Warping Under Translation: Approximation Guided by Space-Filling Curves

Karl Bringmann ✉

Universität des Saarlandes, Saarbrücken, Germany

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Sándor Kisfaludi-Bak ✉

Aalto University, Espoo, Finland

Marvin Künnemann

Institute for Theoretical Studies, ETH Zürich, Switzerland

Dániel Marx ✉ 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

André Nusser ✉ 

BARC, University of Copenhagen, Denmark

Abstract

The Dynamic Time Warping (DTW) distance is a popular measure of similarity for a variety of sequence data. For comparing polygonal curves π, σ in \mathbb{R}^d , it provides a robust, outlier-insensitive alternative to the Fréchet distance. However, like the Fréchet distance, the DTW distance is not invariant under translations. Can we efficiently optimize the DTW distance of π and σ under arbitrary translations, to compare the curves' *shape* irrespective of their absolute location?

There are surprisingly few works in this direction, which may be due to its computational intricacy: For the Euclidean norm, this problem contains as a special case the geometric median problem, which provably admits no exact algebraic algorithm (that is, no algorithm using only addition, multiplication, and k -th roots). We thus investigate exact algorithms for non-Euclidean norms as well as approximation algorithms for the Euclidean norm.

For the L_1 norm in \mathbb{R}^d , we provide an $\mathcal{O}(n^{2(d+1)})$ -time algorithm, i.e., an exact polynomial-time algorithm for constant d . Here and below, n bounds the curves' complexities. For the Euclidean norm in \mathbb{R}^2 , we show that a simple problem-specific insight leads to a $(1 + \varepsilon)$ -approximation in time $\mathcal{O}(n^3/\varepsilon^2)$. We then show how to obtain a subcubic $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$ time algorithm with significant new ideas; this time comes close to the well-known quadratic time barrier for computing DTW for fixed translations. Technically, the algorithm is obtained by speeding up repeated DTW distance estimations using a dynamic data structure for maintaining shortest paths in weighted planar digraphs. Crucially, we show how to traverse a candidate set of translations using space-filling curves in a way that incurs only few updates to the data structure.

We hope that our results will facilitate the use of DTW under translation both in theory and practice, and inspire similar algorithmic approaches for related geometric optimization problems.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Dynamic Time Warping, Sequence Similarity Measures

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.20

Related Version *Full Version*: <https://arxiv.org/abs/2203.07898>

Funding *Karl Bringmann*: This work is part of the project TIPEA that has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 850979).

Sándor Kisfaludi-Bak: Part of this research was conducted while the author was at the Max Planck Institute for Informatics, and part of it while he was at the Institute for Theoretical Studies, ETH Zürich.



© Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, Dániel Marx, and André Nusser;

licensed under Creative Commons License CC-BY 4.0

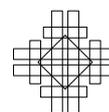
38th International Symposium on Computational Geometry (SoCG 2022).

Editors: Xavier Goaoc and Michael Kerber; Article No. 20; pp. 20:1–20:17

Leibniz International Proceedings in Informatics



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



Marvin Künnemann: Research supported by Dr. Max Rössler, by the Walter Haefner Foundation, and by the ETH Zürich Foundation.

Dániel Marx: Research supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH.

André Nusser: Part of this research was conducted while the author was at Saarbrücken Graduate School of Computer Science and Max Planck Institute for Informatics. The author is supported by the VILLUM Foundation grant 16582.

Acknowledgements We thank Christian Wulff-Nilsen for making us aware of an improved offline dynamic shortest paths data structure.

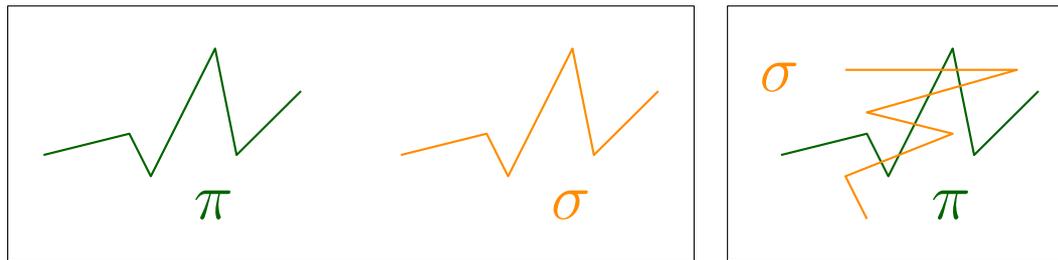
1 Introduction

Fast algorithms for computing similarity measures for sequence data enable a number of applications such as signature/handwriting recognition [39, 18], map matching [8, 37], analysis of GPS tracking data [9] and many more. For polygonal curves in \mathbb{R}^d , a popular measure is the Fréchet distance [4, 20] – we refer to [25] for an overview over the extensive literature. Unfortunately, the Fréchet distance is very sensitive to outliers, as the distance value may easily be dominated by erroneous samplings of the curves. Consequently, some contexts would profit from a measure that is more robust to outliers, such as the average/integral Fréchet distance (see [14, 31]) or the well-known dynamic time warping (DTW) distance. The DTW distance is particularly popular for audio sequences (such as speech recognition) and other domains, but has seen an increasing number of uses for geometric curves [33, 39, 18, 38, 9].

Given two polygonal curves $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$ in \mathbb{R}^d , their DTW distance $d_{\text{DTW}}(\pi, \sigma)$ can be defined as follows: We imagine a dog walking on π and its owner walking on σ . Both owner and dog start at the beginning of their curves, and in each step independently decide to either stay in place or jump to the next vertex, until both of them have reached the end of their curves. Formally, this yields a traversal $T = ((i_1, j_1), \dots, (i_t, j_t))$ where $i_1 = j_1 = 1$, $i_t = n$, $j_t = m$ and $(i_{\ell+1}, j_{\ell+1}) \in \{(i_\ell + 1, j_\ell), (i_\ell, j_\ell + 1), (i_\ell + 1, j_\ell + 1)\}$. We define the cost of this traversal as the sum of distances of dog and owner during the traversal, i.e., $\sum_{\ell=1}^t \|\pi_{i_\ell} - \sigma_{j_\ell}\|$. The corresponding DTW distance $d_{\text{DTW}}(\pi, \sigma)$ is defined as the minimum cost of such a traversal.¹ Note that this measure depends on the metric space we use for our curves π, σ . For any metric that we can evaluate in constant time, a simple dynamic programming approach computes $d_{\text{DTW}}(\pi, \sigma)$ in time $O(nm)$, i.e., time $O(n^2)$ when both curves have at most n vertices. While one can achieve mild improvements over this running time [23], one can rule out $O(n^{2-\varepsilon})$ -time algorithms under the Strong Exponential Time Hypothesis, already for curves in \mathbb{R} [1, 11]. Even for constant-factor approximations, no strongly subquadratic algorithms are known, see [30] for (sub)polynomial approximation guarantees and [3, 38] for approximation algorithms on restricted input models.

Unfortunately, the DTW distance is not *translation-invariant*: Distant copies of the same curve may have a much larger distance than differently shaped curves that stay close to each other, see Figure 1. For certain curve similarity applications such as signature recognition, it is thus frequently argued (sometimes implicitly) that a translation-invariant measure is desirable, see e.g. [33, 19, 35, 39, 17].

¹ For comparison, to obtain the discrete Fréchet distance of π and σ , we would minimize, over all traversals T , the *maximum* distance of the dog and its owner during T – one may think of the smallest leash length required to connect dog and owner while traversing their curves.



■ **Figure 1** Curves with similar shape but large DTW distance (left) and different shape but small DTW distance (right).

Arguably the most natural way to make any curve distance measure translation-invariant is to take its minimum under translations of the curves: correspondingly, DTW under translation is defined as $d_{\text{DTW}}^T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^d} d_{\text{DTW}}(\pi, \sigma + \tau)$. Unfortunately, for computing this translation-invariant measure, much less is known than, e.g., for the Fréchet distance under translation. This state of the art, which we review below, is the starting point for our work.

Translation-invariant curve similarity measures. For the continuous Fréchet distance, the earliest algorithmic work studying its translation-invariant version dates back to 2001 [19, 5], with algorithms running in time $\tilde{O}(n^{10})$ and $\tilde{O}(n^8)$, respectively. For the discrete Fréchet distance under translation, algorithms have been improved from $\tilde{O}(n^6)$ [26], via $\tilde{O}(n^5)$ [7], to $\tilde{O}(n^{4.667})$ [13], with a conditional lower bound of $n^{4-o(1)}$ based on the Strong Exponential Time Hypothesis [13]. These theoretical results have been complemented by an algorithm engineering study [12]. Approximation algorithms have been given by [19, 5], including a $(1 + \varepsilon)$ -approximation in time $O(n^2/\varepsilon^2)$. Other works study related settings, such as more general transformations than translations [36, 32], or data structure variants [24].

Unfortunately, we are not aware of algorithmic works with rigorous analyses for DTW under translation, but only heuristic approaches or works on related but different measures. Qiao and Yasuhara [39] experimentally evaluate an iterative method for DTW distance under transformations including translation, rotation and scaling, but provide no theoretical guarantees. Vlachos, Kollios, and Gunopulos [35] study a closely related measure, a variation of the Longest Common Subsequence distance for geometric curves that is translation-invariant. This measure is similar to the DTW distance under translation using a binary distance metric with $d(x, y) = 0$ if $\|x - y\|_\infty \leq \varepsilon$ and $d(x, y) = 1$ otherwise. For their measure, Vlachos et al. provide both exact and approximation algorithms. Munich and Perona [33] define another translation-invariant measure that roughly speaking minimizes differences in direction and velocity changes over traversals of the curves. Efrat, Fan, and Venkatasubramanian [18] study further variants of this measure.

One of the reasons for this lack of rigorous algorithmic work for DTW under translation may be its computational intricacy: Already when $\pi = (\pi_1, \dots, \pi_n)$ is a polygonal curve in \mathbb{R}^d and $\sigma = (\sigma_1)$ consists of a single point $\sigma_1 \in \mathbb{R}^d$, we obtain the geometric median problem as a special case. Specifically, the task simplifies to finding a point $x \in \mathbb{R}^d$ such that $\sum_{i=1}^n \|\pi_i - x\|$ is minimized. This problem provably has no exact algebraic algorithm already for $n = 5$ and $d = 2$ [6] (that is, no algorithm using only addition, multiplication, and k -th roots). We refer to [15] for a recent near-linear time approximation algorithm and an overview of the literature on geometric median. By this lack of an exact, efficient algorithm for geometric median, we can thus hardly expect to solve DTW under translation in Euclidean spaces *exactly*. This motivates to study the problem for norms other than Euclidean, as well as to study approximation algorithms for the Euclidean norm.

1.1 Our results

Exact algorithms for non-Euclidean norms. For the L_1 norm in \mathbb{R}^d , we give a polynomial-time exact algorithm whenever d is constant.

► **Theorem 1.** *For the L_1 -norm in \mathbb{R}^d we can solve DTW under translation in time $\mathcal{O}(n^{2(d+1)})$.*

Since in \mathbb{R}^2 we can transform the L_∞ norm to the L_1 norm by rotating the input by $\frac{\pi}{2}$ and scaling by $1/\sqrt{2}$, this also yields an $\mathcal{O}(n^6)$ time algorithm for L_∞ in \mathbb{R}^2 . We prove the result in the full version.

Approximation algorithms for the Euclidean norm. The main focus in this paper is DTW under translation in the Euclidean plane. Since there is no exact algebraic algorithm due to the special case of geometric median, we focus on developing an *approximation* algorithm.

As a first baseline, we observe that DTW under translation is at least as hard to compute as DTW for a fixed translation, even for approximation (we prove this in the full version). Since exactly computing DTW for a fixed translation requires time $n^{2-o(1)}$ under the Strong Exponential Time Hypothesis [1, 11], and no subquadratic-time constant-factor approximation algorithm is known, the best we could hope for with current techniques would be a $f(1/\varepsilon)n^2$ -time algorithm. Can we reach this baseline or does optimizing over translations in \mathbb{R}^2 increase the problem’s complexity (and if so, by how much)?

For the discrete Fréchet distance, optimizing over a translation increases the time complexity from $n^{2\pm o(1)}$ [20, 10] to at least $n^{4-o(1)}$ and at most $\mathcal{O}(n^{4.667})$ [13] (where the lower bounds are based on the Strong Exponential Time Hypothesis). For $(1 + \varepsilon)$ -approximations, a simple algorithm indeed manages to match the baseline of $\mathcal{O}(n^2/\varepsilon^2)$, see [5]. Does the same hold true for the DTW distance?

Similar arguments to [5] only achieve an $\tilde{\mathcal{O}}(n^4/\varepsilon^2)$ time bound for DTW under translation. Using an insight specific to the nature of the DTW distance, we present a surprisingly simple $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ time algorithm. We describe both approaches in Section 1.2. Our most important contribution is to obtain a *subcubic* $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$ time bound via a sophisticated approach that exploits geometric arguments (specifically, a traversal via space-filling curves) to reduce our problem to maintaining shortest paths in a dynamically changing directed grid graph.

► **Theorem 2.** *For the Euclidean norm in \mathbb{R}^2 , we can solve $(1 + \varepsilon)$ -approximate DTW under translation in time $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$.*

Our techniques strengthen the paradigm of using dynamic algorithms for geometric optimization problems, for which we see a growing number of applications (besides classical examples such as [34], see, e.g., recent work for the Fréchet distance under translation [7, 13] or polygon placement [29]). Finally, only a sublinear factor of $\tilde{\mathcal{O}}(\sqrt{n})$ to the baseline of $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ remains, which one might hope to decrease by further developing our ideas.

1.2 Technical overview

In this section, we describe the main ideas for our approximation algorithm for DTW under translation. To keep this exposition as simple as possible, we assume that both curves have the same complexity; let these curves be denoted by $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_n)$ throughout this section. The proof in Section 3 gives the slightly more detailed arguments for possibly different complexities of the curves. We start off with a simple algorithm that achieves a rather modest approximation guarantee: Let $\tau_{\text{start}} := \pi_1 - \sigma_1$ denote the

translation of σ that aligns the first points of π and $\sigma + \tau$. It is straightforward to prove that the resulting DTW distance $\delta_{\text{start}} := d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}})$ yields a $2n$ -approximation to DTW under translation, i.e., $d_{\text{DTW}}^T(\pi, \sigma) \in [\delta_{\text{start}}/(2n), \delta_{\text{start}}]$. This follows from the fact that $d_{\text{DTW}}(\pi, \sigma + \tau)$ is $(2n - 1)$ -Lipschitz with respect to τ , and that $\tau^* := \operatorname{argmin}_{\tau} d_{\text{DTW}}(\pi, \sigma + \tau)$ satisfies $\|\tau_{\text{start}} - \tau^*\| \leq d_{\text{DTW}}^T(\pi, \sigma)$, see Lemma 4. (Analogous arguments are known to give a 2-approximation for the Fréchet distance under translation [5, 12].)

With this rough approximation, the main task for approximating DTW under translation is to design an approximate decider with the following guarantee: Given the polygonal curves π, σ , a threshold $\delta > 0$ and approximation parameter $\varepsilon > 0$, output a verdict “ $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ ” or “ $d_{\text{DTW}}^T(\pi, \sigma) > \delta$ ” in time $T(n, \varepsilon)$. In any case, the returned verdict has to be correct, i.e., if $\delta < d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ any output is admissible, otherwise it is uniquely determined. Given such an approximate decider, it is straightforward to obtain a $(1 + \varepsilon)$ -approximation algorithm with running time $\mathcal{O}(T(n, \varepsilon/3) \log(n/\varepsilon))$ via binary search in the interval $[\delta_{\text{start}}/(2n), \delta_{\text{start}}]$, see Theorem 11. We thus focus on the approximate decider for the remainder of this section.

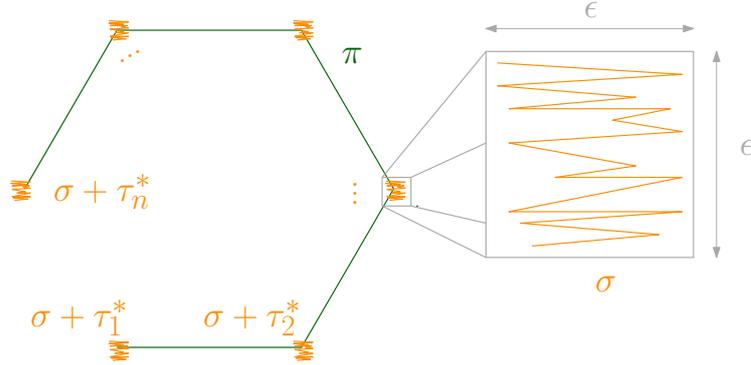
A simple $\mathcal{O}(n^4/\varepsilon^2)$ solution. Let B be the square of side length 2δ centered at $\tau_{\text{start}} = \pi_1 - \sigma_1$. To approximately decide whether $d_{\text{DTW}}(\pi, \sigma) \leq \delta$, we only need to consider translations in B , as any other translation τ incurs a DTW distance larger than δ by $d_{\text{DTW}}(\pi, \sigma + \tau) \geq \|\pi_1 - (\sigma_1 + \tau)\| = \|\tau_{\text{start}} - \tau\| > \delta$. Note that we can discretize this bounding box by a set Q of $\mathcal{O}((n/\varepsilon)^2)$ translations such that for each translation $\tau^* \in B$, there is a close translation $\tau \in Q$ with $\|\tau^* - \tau\| \leq \frac{\varepsilon\delta}{2n}$. Thus, if there is a translation τ^* with $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then by $(2n - 1)$ -Lipschitzness (Lemma 3), there is a translation $\tau \in Q$ with $d_{\text{DTW}}(\pi, \sigma + \tau) \leq d_{\text{DTW}}(\pi, \sigma + \tau^*) + (2n - 1) \cdot \|\tau^* - \tau\| \leq (1 + \varepsilon)\delta$. Consequently, by deciding $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$ for all $\tau \in Q$ using the exact $\mathcal{O}(n^2)$ -time algorithm, we obtain an approximate decider with running time $T(n, \varepsilon) = \mathcal{O}(n^4/\varepsilon^2)$.

Note that the above arguments simplify the problem as follows: Find a set Q of translations such that if there is some *witness translation* τ^* , i.e., $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then there is some $\tau \in Q$ with $\|\tau - \tau^*\| \leq \frac{\varepsilon\delta}{2n}$. By computing $d_{\text{DTW}}(\pi, \sigma + \tau)$ for all $\tau \in Q$, we can then approximately decide whether $d_{\text{DTW}}^T(\pi, \sigma) \leq \delta$.

A more careful $\mathcal{O}(n^3/\varepsilon^2)$ solution. It turns out that we can significantly reduce the size of the set Q by analyzing the properties of good DTW traversals more closely. Consider a DTW traversal $((i_1, j_1), \dots, (i_t, j_t))$ of π and $\sigma + \tau^*$, with traversal cost $\sum_{\ell=1}^t \|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \leq \delta$. Then, by a simple Markov argument, there can be at most $n/2$ pairs $\pi_{i_\ell}, \sigma_{j_\ell}$ with $\|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \geq 2\delta/n$, since otherwise already these pairs would lead to a traversal cost of more than δ . Since the traversal has $t \geq n$ steps, it follows that there are at least $t - n/2 \geq n/2$ pairs $\pi_{i_\ell}, \sigma_{j_\ell}$ with $\|\pi_{i_\ell} - (\sigma_{j_\ell} + \tau^*)\| \leq 2\delta/n$. Since $\pi_\ell - (\sigma_\ell + \tau^*) = (\pi_\ell - \sigma_\ell) - \tau^*$, this yields an important restriction on τ^* :

For any τ^ such that π and $\sigma + \tau^*$ have DTW distance at most δ ,
there exist at least $n/2$ pairs π_i, σ_j with $\|(\pi_i - \sigma_j) - \tau^*\| \leq 2\delta/n$.*

This property immediately gives a simple randomized $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ algorithm: Simply draw a pair (i, j) uniformly at random from $[n]^2$ and test all translations τ given by $\mathcal{O}(1/\varepsilon^2)$ equally-spaced points in a $[-\frac{2\delta}{n}, \frac{2\delta}{n}]^2$ -box $C_{i,j}$ centered at $\pi_i - \sigma_j$. If there is some $\tau^* \in C_{i,j}$ with $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, one of the checked translations τ achieves $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$. By the above property, we have that $\tau^* \in C_{i,j}$ with probability at least $(n/2)/n^2 = 1/(2n)$. Thus, it suffices to repeat this process $\tilde{\mathcal{O}}(n)$ times to find a good translation with high probability, if one exists. This yields a total running time of $\tilde{\mathcal{O}}(n \cdot \frac{1}{\varepsilon^2} \cdot n^2) = \tilde{\mathcal{O}}(n^3/\varepsilon^2)$.



■ **Figure 2** If π is given by a regular n -gon and σ is a $3n$ -vertex curve in a small $[0, \varepsilon]^2$ area, where ε is small, then the DTW under translation distance can have $\Omega(n)$ local optima, each of which is near-optimal. These local optima correspond to translating σ towards each vertex of π .

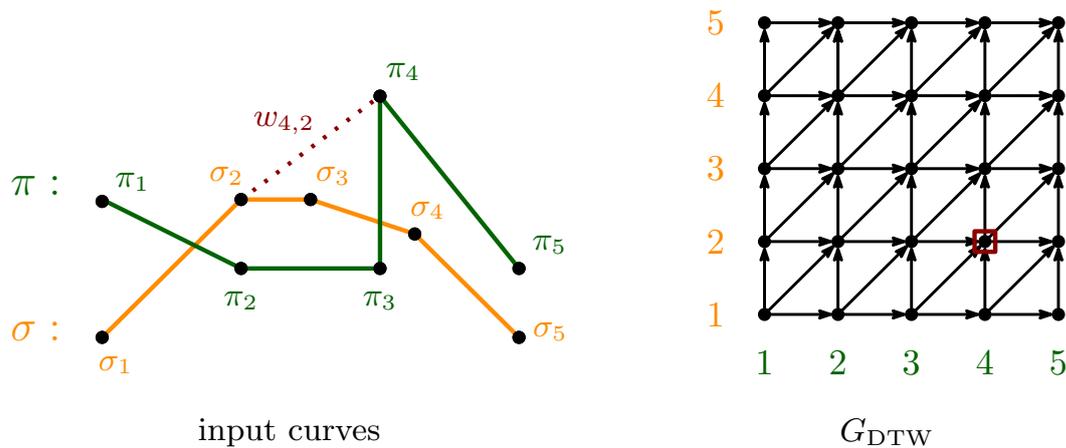
In order to leverage this property *deterministically*, define the multiset $P := \{\pi_i - \sigma_j \mid i, j \in [n]\}$ of n^2 points. Recall that B is the square of side length 2δ centered at $\tau_{\text{start}} = \pi_1 - \sigma_1$. We impose a grid on the bounding box B where each grid cell has side length $2\delta/n$. Consider a translation τ^* in some grid cell C such that π and $\sigma + \tau^*$ have DTW distance at most δ . Then there must be $n/2$ points $p \in P$ with $\|p - \tau^*\| \leq 2\delta/n$ – these points are distributed among C and at most 3 neighboring cells of C .² Thus, for any witness translation τ^* , there must be a neighboring (including itself) grid cell containing at least $n/8$ points from P – we call such a cell *dense*. Thus, we only need to check for translations that are inside a dense cell or neighboring a dense cell. Since $|P| = n^2$, there can be at most $|P|/(n/8) = 8n$ dense cells, resulting in $\mathcal{O}(n)$ cells to check for a good translation.

Since each grid cell has side length $\mathcal{O}(\delta/n)$, we can discretize each relevant cell C by $\mathcal{O}(1/\varepsilon^2)$ many translations Q_C such that if any $\tau^* \in C$ achieves $d_{\text{DTW}}(\pi, \sigma + \tau^*) \leq \delta$, then there is a $\tau \in Q_C$ with $\|\tau^* - \tau\| \leq \varepsilon\delta/(2n)$ and thus $d_{\text{DTW}}(\pi, \sigma + \tau) \leq (1 + \varepsilon)\delta$. Thus, by letting Q be the union of Q_C for all $\mathcal{O}(n)$ cells C that we need to check, we obtain $|Q| = \mathcal{O}(n/\varepsilon^2)$, significantly improving over the previous bound of $\mathcal{O}(n^2/\varepsilon^2)$. Computing the DTW distance for each translation in Q , we obtain a *deterministic* $\mathcal{O}(n^3/\varepsilon^2)$ -time algorithm.

Beating $\mathcal{O}(n^3/\varepsilon^2)$. Can we improve over the previous algorithm? A first idea would be to try to reduce the size of Q even further, below $\Theta(n \cdot \text{poly}(1/\varepsilon))$. However, there is evidence that this route is rather difficult: One can construct instances with $\Omega(n)$ many near-optimal local optima that are well-separated from each other, see Figure 2. It thus appears quite challenging to avoid a check of $\Omega(n)$ regions of translations.

A different route is to speed up the computation of DTW distances $d_{\text{DTW}}(\pi, \sigma + \tau)$ over all $\tau \in Q$, avoiding the naive time bound of $\mathcal{O}(|Q| \cdot n^2)$. Such approaches have been proven successful for related geometric optimization problems, such as Fréchet distance under translation [7, 13] or polygon placement [29]. Crucially, one needs to exploit that the $|Q|$ distance computations are related (for solving $|Q|$ independent instances, a conditional lower bound of $(|Q|n^2)^{1-o(1)}$ can be shown based on the quadratic-time hardness for DTW [1, 11]). To this end, we open up the black-box $\mathcal{O}(n^2)$ -algorithm for DTW.

² Here, we say that two cells are neighboring if they share a common vertex.



■ **Figure 3** An example of two curves π, σ and their dynamic time warping graph G_{DTW} .

Given $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_n)$, let G_{DTW} denote the node-weighted directed grid graph with vertex set $V = \{(i, j) \mid i, j \in [n]\}$ and edge set E consisting of horizontal edges from (i, j) to $(i+1, j)$, vertical edges from (i, j) to $(i, j+1)$ and diagonal edges from (i, j) to $(i+1, j+1)$. Each node (i, j) receives the weight $w_{i,j} = \|\pi_i - \sigma_j\|$. Then it is not difficult to see that $d_{DTW}(\pi, \sigma)$ is equal to the distance from $(1, 1)$ to (n, n) in G_{DTW} . As such, we can exploit algorithmic results on maintaining shortest paths in weighted planar digraphs under weight updates (here, one usually considers edge-weighted graphs, which subsumes the node-weighted setting). Unfortunately, when translating σ by τ , $\Omega(n^2)$ weights may change in G_{DTW} so that even constant-time updates would lead to an $\Omega(n^3/\varepsilon^2)$ time solution. In contrast, work on the Fréchet distance under translation [7, 13] considers translations in an order that incurs only $\mathcal{O}(1)$ updates per translation.

Surprisingly, one can indeed reduce the number of weight updates below $\mathcal{O}(n^2)$ when we resort to *approximating* each weight $w_{i,j}$ by an estimate $\|\pi_i - (\sigma_j + \tau)\|/(1 + \varepsilon) \leq w_{i,j} \leq (1 + \varepsilon) \cdot \|\pi_i - (\sigma_j + \tau)\|$. Specifically, we show how to traverse the $\mathcal{O}(n/\varepsilon^2)$ translations in Q in an order specified by a *space-filling curve* such that we only need to update $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ weights in total to maintain approximate weights. This statement and its analysis is one of the most interesting technical contributions of this paper and is proven in Section 3.2. It remains to report the shortest distance from $(1, 1)$ to (n, n) in the directed grid graph G_{DTW} for $\mathcal{O}(n/\varepsilon^2)$ queries and $\tilde{\mathcal{O}}(n^2/\varepsilon^2)$ weight updates. For this task, we use the data structure due to Das et al. [16] whose parameters can be set to give query time $\tilde{\mathcal{O}}(N^{3/4})$ and update time $\tilde{\mathcal{O}}(N^{1/4})$ for weighted planar digraphs with N vertices. Since $N = n^2$, we obtain a total running time of $\tilde{\mathcal{O}}(n^{2.5}/\varepsilon^2)$, which improves polynomially over the previous $\tilde{\mathcal{O}}(n^3/\varepsilon^2)$ solution. We believe that our approach of maintaining approximate weights efficiently using a space-filling curve traversal may turn out useful for further improvements in similar contexts of geometric optimization problems.

2 Preliminaries & notation

To denote index sets we use the notation $[n] := \{1, \dots, n\}$. Let $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ and $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_m)$ be two sequences of points in \mathbb{R}^d . We assume $n \geq m$ without loss of generality. To define the Dynamic Time Warping distance (DTW), we first introduce traversals. A sequence of index pairs $T = ((i_1, j_1), (i_2, j_2), \dots, (i_L, j_L))$ is a traversal of

20:8 (Approximation) Algorithms for Dynamic Time Warping Under Translation

two curves of complexity n and m if $(i_1, j_1) = (1, 1)$, $(i_L, j_L) = (n, m)$, and $(i_{\ell+1}, j_{\ell+1}) \in \{(i_\ell + 1, j_\ell), (i_\ell, j_\ell + 1), (i_\ell + 1, j_\ell + 1)\}$ for each $\ell \in [L - 1]$. We call L the number of steps of the traversal T . Let $\mathcal{T}_{n,m}$ be the set of all traversals of curves of length n and m . The Dynamic Time Warping distance between π and σ is then defined as

$$d_{\text{DTW}}(\pi, \sigma) := \min_{T \in \mathcal{T}} \sum_{(i,j) \in T} d(\pi_i, \sigma_j),$$

where for the metric $d(\cdot, \cdot)$, we use the L_p -norm $d(x, y) = \|x - y\|_p$ throughout this paper. In the remainder, we omit the p as it is either clear from the context, or the statement holds for all $p \in [1, \infty)$. Furthermore, we often use bounds on the number of steps of the traversal. To that end, note that for $m \leq n$, any traversal in $\mathcal{T}_{n,m}$ consists of at least n and at most $n + m - 1$ steps.

For a sequence $\pi = (\pi_1, \dots, \pi_n)$ with $\pi_i \in \mathbb{R}^d$ and a translation $\tau \in \mathbb{R}^d$, we define the translated sequence as $\pi + \tau := (\pi_1 + \tau, \pi_2 + \tau, \dots, \pi_n + \tau)$. Dynamic Time Warping Under Translation is then defined as $d_{\text{DTW}}^T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^d} d_{\text{DTW}}(\pi, \sigma + \tau)$. Recall that a function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ is called L -Lipschitz (with respect to norm $\|\cdot\|$) if for any $\tau, \tau' \in \mathbb{R}^d$ we have $|f(\tau) - f(\tau')| \leq L \cdot \|\tau - \tau'\|$. We prove the following lemma in the full version.

► **Lemma 3.** $d_{\text{DTW}}(\pi, \sigma + \tau)$ is $(n + m - 1)$ -Lipschitz in τ .

The following lemma gives a simple $(n + m)$ -approximation for DTW under translation and is a straightforward adaption of a corresponding 2-approximation for the Fréchet distance under translation [12, Observation 2]. Note that one can create simple examples where this approximation ratio is almost tight. Again, we defer the proof to the full version.

► **Lemma 4.** Let $\tau_{\text{start}} = \pi_1 - \sigma_1$. Then $d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}}) \leq (n + m) \cdot d_{\text{DTW}}^T(\pi, \sigma)$.

As discussed in Section 1, DTW corresponds to a grid graph problem. We now formally define this. Given a DTW instance with curves $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$, we define a directed graph $G_{\text{DTW}} = (V, E, w)$ on a node-weighted grid (including certain diagonals) with node set $V := \{(i, j) \mid i \in [n], j \in [m]\}$, edge set

$$E := \{((i, j), (i + 1, j)) \mid i \in [n - 1], j \in [m]\} \cup \{((i, j), (i, j + 1)) \mid i \in [n], j \in [m - 1]\} \\ \cup \{((i, j), (i + 1, j + 1)) \mid i \in [n - 1], j \in [m - 1]\},$$

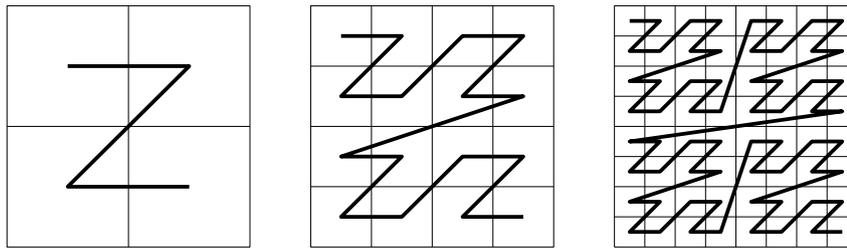
and weights $w: V \rightarrow \mathbb{R}$ with $w((i, j)) := \|\pi_i - \sigma_j\|$. To simplify notation, we write $w_{i,j}$ instead of $w((i, j))$ to denote the weight of node (i, j) . Note that finding a shortest path in this graph from $(1, 1)$ to (n, m) is equivalent to finding the minimum cost traversal.

In order to define the order of the updates and queries in the dynamic graph problem that we introduce in Section 3, we use a space-filling curve on a grid. Let

$$\mathcal{G}_R := \{i \cdot R \mid i \in \mathbb{Z}\} \times \{j \cdot R \mid j \in \mathbb{Z}\}.$$

be an infinite grid with resolution $R \in \mathbb{R}$. For our purpose, a space-filling curve is a hierarchical traversal of a finite grid: we partition this grid into four parts and, in some fixed order of the parts, recursively traverse each subgrid exhaustively before traversing the next one. More precisely, we define the curve on the $2^k \times 2^k$ grid $C_{0,0}^k := \mathcal{G}_R \cap [0, (2^k - 1)R]^2$ for some R , and we recursively split $C_{i,j}^\ell$ into the boxes $C_{2i,2j}^{\ell-1}, C_{2i,2j+1}^{\ell-1}, C_{2i+1,2j}^{\ell-1}, C_{2i+1,2j+1}^{\ell-1}$ until they only contain a single grid point, i.e., until $\ell = 0$. This leads to the following definition:

$$C_{i,j}^\ell := \{(i2^\ell + s)R \mid s \in \{0, \dots, 2^\ell - 1\}\} \times \{(j2^\ell + s)R \mid s \in \{0, \dots, 2^\ell - 1\}\}.$$



■ **Figure 4** The traversal of the z-curve for $k = 1$, $k = 2$, and $k = 3$.

For each cell $C_{i,j}^\ell$ with $\ell > 0$, the space-filling curve then traverses the points of the children in a way such that for each child all points are traversed in a continuous piece. For example, we first traverse all points of $C_{2i,2j+1}^{\ell-1}$, then $C_{2i+1,2j+1}^{\ell-1}$, then $C_{2i,2j}^{\ell-1}$, and finally $C_{2i+1,2j}^{\ell-1}$. Recursively applying this leads to a sequence $z_1, \dots, z_{2^{2^k}}$ of all points in $\mathcal{G}_R \cap [0, (2^k - 1)R]^2$. This sequence is called the z-curve, see Figure 4. (However, any other order to traverse the children also works for our purpose.)

To argue about the space-filling curve traversals, it is sometimes useful to view the grid of points \mathcal{G}_R equivalently as a grid of cells, i.e., as a set of squares partitioning \mathbb{R}^2 . To switch between these views, build the Voronoi diagram of the point grid to obtain the cell grid, and conversely, use the center of each cell to obtain the point grid. We will freely use whichever view is most convenient in any context.

3 Approximating DTW under translation in L_p

In this section we present an $\tilde{O}(n^{2.5}/\varepsilon^2)$ algorithm for the problem of $(1 + \varepsilon)$ -approximating DTW under translation in the Euclidean plane. The algorithm that we present consists of two parts. First, we reduce to a dynamic shortest path problem on a grid graph. Second, we show that with the resulting number of updates and queries, we can use an existing dynamic graph algorithm to then obtain a subcubic algorithm for the problem at hand.

Recall that we consider the approximate decision problem: Given sequences $\pi = (\pi_1, \dots, \pi_n)$ and $\sigma = (\sigma_1, \dots, \sigma_m)$ with $\pi_i, \sigma_j \in \mathbb{R}^2$, a distance $\delta \in \mathbb{R}$, and an approximation parameter $\varepsilon > 0$, either decide that $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$ or that $d_{\text{DTW}}^T(\pi, \sigma) > \delta$. Recall that we assume $n \geq m$. We first present a basic cubic algorithm that already captures some important properties of the subcubic algorithm that we subsequently present.

3.1 Cubic algorithm

We now present the cubic algorithm that was already outlined in Section 1.2. First, if $\delta = 0$, we make a precise decision by testing for $d_{\text{DTW}}(\pi, \sigma + \tau_{\text{start}}) = 0$ with $\tau_{\text{start}} = \pi_1 - \sigma_1$. To facilitate the presentation, we furthermore assume that ε is given such that $\frac{n}{\varepsilon} = 2^k$ for some $k \in \mathbb{N}$. We can easily achieve this by rounding the input ε down to the largest value that fulfils this constraint, which changes the value of ε by at most a factor of 2.

In another preprocessing step, we round the coordinates of the points of π and σ to the closest multiple of $\frac{\delta}{4n}\varepsilon$. This is feasible as it changes the DTW distance by less than

$$(n + m) \cdot \frac{\delta}{4n}\varepsilon \leq \delta \frac{\varepsilon}{2}.$$

The multiset of translations from any point in σ to any point in π is then defined as

$$P := \{\pi_i - \sigma_j \mid i \in [n] \text{ and } j \in [m]\}.$$

20:10 (Approximation) Algorithms for Dynamic Time Warping Under Translation

Note that by construction also all coordinates of all points in P are multiples of $\frac{\delta}{4n}\varepsilon$. Furthermore, as P is a multiset, we have $|P| = nm$. We now define a set of boxes that enables us to find dense regions. Consider the square $B := [-\delta, \delta]^2 + \tau_{\text{start}}$. Partition B into n^2 boxes B_1, \dots, B_{n^2} of size $\frac{2\delta}{n} \times \frac{2\delta}{n}$ (note that their boundaries might intersect). We now formally define the notion of a dense box already introduced intuitively in Section 1.2.

► **Definition 5** (Dense Box). *A box B_i is dense if at least $\frac{n}{18}$ points of P are contained in B_i .*

As $|P| = nm$, we obtain the following observation:

► **Observation 6.** *There are at most $18m$ dense boxes.*

Note that we can find the dense boxes in time $\tilde{\mathcal{O}}(|P|)$ by associating each point with the tuple of indices in $[n] \times [n]$ of its containing box and then sorting these tuples. Now, let $N(B_i)$ be the neighborhood of a box B_i , i.e., $N(B_i) := \{B_j \mid B_j \cap B_i \neq \emptyset\}$. Note that $B_i \in N(B_i)$, so each box has (up to) 9 neighbors. The crucial property of dense boxes is that any witness translation τ with $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$ has to be in the neighborhood of a dense box:

► **Lemma 7.** *If $d_{\text{DTW}}^T(\pi, \sigma) \leq \delta$, then there exists a dense box B_j , a neighbor $B_i \in N(B_j)$, and a $\tau \in B_i$ such that $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$.*

The proof is deferred to the full version.

As we have to approximately decide whether $d_{\text{DTW}}(\pi, \sigma + \tau) \leq \delta$ for any τ that is neighboring a dense box, we intersect each of these boxes with an $8\varepsilon \times 8\varepsilon$ grid and this gives us the set of points Q that we have to evaluate. More precisely, let $\mathcal{G} := \mathcal{G}_{\frac{\delta}{4n}\varepsilon} \cap B$, where again $B = [-\delta, \delta]^2 + \tau_{\text{start}}$. Note that all points of \mathcal{G} are still integer multiples of $\frac{\delta}{4n}\varepsilon$. We now define our set of evaluation points to be

$$Q := \{\mathcal{G} \cap B_i \mid B_i \in N(B_j) \text{ for some dense box } B_j\}.$$

Note that from Observation 6 and the bound $|\mathcal{G} \cap B_i| \in \mathcal{O}(\frac{1}{\varepsilon^2})$, it follows that $|Q| \in \mathcal{O}(\frac{m}{\varepsilon^2})$.

Computing $d_{\text{DTW}}(\pi, \sigma + q)$ for each $q \in Q$ suffices to implement an approximate decider. Indeed, if for some $q \in Q$ we find $d_{\text{DTW}}(\pi, \sigma + q) \leq (1 + \varepsilon)\delta$, then we conclude that $d_{\text{DTW}}^T(\pi, \sigma) \leq (1 + \varepsilon)\delta$. Otherwise, if $d_{\text{DTW}}(\pi, \sigma + q) > (1 + \varepsilon)\delta$ for all $q \in Q$, then we conclude that $d_{\text{DTW}}^T(\pi, \sigma) > \delta$, by the following lemma proven in the full version.

► **Lemma 8** (Correctness). *If $d_{\text{DTW}}(\pi, \sigma + q) > (1 + \varepsilon)\delta$ for all $q \in Q$, then $d_{\text{DTW}}^T(\pi, \sigma) > \delta$.*

If we just evaluate each point in Q naively, then the running time is $\mathcal{O}(nm^2(\frac{1}{\varepsilon})^2)$, as there are $\mathcal{O}(m)$ dense cells, each of them with $(\frac{1}{\varepsilon})^2$ grid points, and each DTW evaluation takes time $\mathcal{O}(nm)$. In the next section, instead of naively recomputing DTW for each translation, we dynamically update the DTW graph weights and then query for the shortest path.

3.2 Reduction to dynamic graph problem

Now we present the first step in solving DTW under translation in subcubic time. To this end, we transform our problem into a dynamic shortest path problem on a grid graph.

Dynamic graph problem

Recall that computing DTW for a fixed translation is a shortest path problem on a grid graph, see Section 2. More precisely, in the grid graph with node weights $w_{i,j} = \|\pi_i - (\sigma_j + q)\|$ the shortest path distance from $(1, 1)$ to (n, m) is equal to $d_{\text{DTW}}(\pi, \sigma + q)$. However, as we only want to compute a $(1 + \varepsilon)$ -approximation, we can relax the condition on the node weights to:

$$\frac{\|\pi_i - (\sigma_j + q)\|}{(1 + \varepsilon)} \leq w_{i,j} \leq (1 + \varepsilon)\|\pi_i - (\sigma_j + q)\|. \quad (1)$$

Observe that for such node weights the shortest path distance from $(1, 1)$ to (n, m) is equal to $d_{\text{DTW}}(\pi, \sigma + q)$ up to a factor $(1 + \varepsilon)$.

We choose the same set of query translations Q as in Section 3.1. We iterate over all $q \in Q$, and for each q we first update the node weights in the grid graph in order to satisfy (1) and then we query the shortest path distance from $(1, 1)$ to (n, m) in the grid graph, obtaining a $(1 + \varepsilon)$ -approximation of $d_{\text{DTW}}(\pi, \sigma + q)$. As in Section 3.1, this yields a $(1 + \mathcal{O}(\varepsilon))$ -approximation of $d_{\text{DTW}}^T(\pi, \sigma)$ (and after scaling ε this becomes a $(1 + \varepsilon)$ -approximation). Note that we did not fix the ordering of the query translations $q \in Q$ yet. In the following, we first fix this ordering, and then argue that our ordering guarantees that the total number of node weight updates is small, and furthermore we can efficiently determine which node weight updates have to be performed.

Query ordering

Consider the z-curve over the grid $\mathcal{G} = \mathcal{G}_{\frac{\delta}{4n}\varepsilon} \cap [-2\delta, 2\delta]^2 + \tau_{\text{start}}$. Note that this z-curve has depth $\log_2(16\frac{n}{\varepsilon})$, and recall that $\frac{n}{\varepsilon}$ is a power of 2. The points of Q lie on the z-curve, as π, σ are rounded and the grid has resolution $\frac{\delta}{4n}\varepsilon$. Thus, the z-curve induces an ordering of Q , and this is the ordering that we choose.

Updates

We now describe how we determine the node weight updates in the dynamic grid graph problem, to ensure that when we run the shortest path query corresponding to $q \in Q$ the node weights satisfy (1). We first argue why the total number of node weight updates is small, and subsequently discuss how to compute the sequence of node weight updates.

► **Lemma 9.** *Consider the sequence of points $\tau_1, \dots, \tau_{(16n/\varepsilon)^2}$ given by the z-curve on the grid \mathcal{G} , and fix i, j . Using only $\mathcal{O}(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon})$ updates to the node weight $w_{i,j}$, we can maintain $w_{i,j}$ as a $(1 + \varepsilon)$ -approximation of the distance $\|\pi_i - (\sigma_j + \tau_k)\|$ while iterating over $k = 1, \dots, (16n/\varepsilon)^2$.*

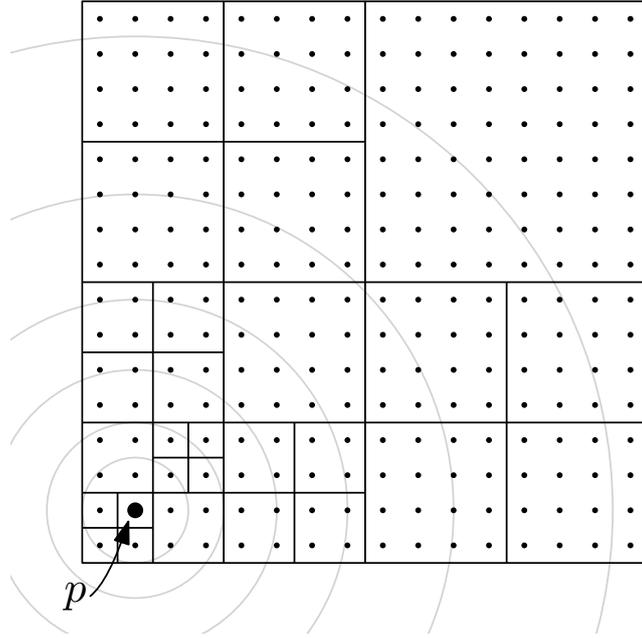
Proof. Note that for $p := \pi_i - \sigma_j$ we have $\|p - \tau\| = \|\pi_i - (\sigma_j + \tau)\|$, and thus the distance that we want to maintain is $\|p - \tau_k\|$. Additionally, note that p lies on the z-curve as the coordinates of the curves are rounded to $\frac{\delta}{4n}\varepsilon$ and, furthermore, if p is further than δ from $\tau_{\text{start}} = \pi_1 - \sigma_1$, then using the pair i, j in the traversal would incur a distance of more than δ for all $q \in Q$ and thus we can just set $w_{i,j} = \infty$.

Now, consider the following process on the recursive definition of the z-curve: Starting with the root $C_{0,0}$, recursively explore all children of each cell. Stop the process at a cell C if there is an $\ell \in \mathbb{Z}$ such that all points $\tau \in C$ have a distance

$$(1 + \varepsilon)^{\ell-1} \leq \|p - \tau\| \leq (1 + \varepsilon)^{\ell+1}.$$

In this case, for all points in C the value $(1 + \varepsilon)^\ell$ is a valid $(1 + \varepsilon)$ -approximation for the distance to p , so we associate the distance $(1 + \varepsilon)^\ell$ with C .

Note that the process is well-defined, since at the lowest level of the recursion a cell contains only a single point of \mathcal{G} , and thus at the latest on this level the process will stop. The process partitions the grid \mathcal{G} into cells C_1, \dots, C_t , which are exhaustively explored in this order by the z-curve, see Figure 5 for an illustration. In particular, the value of t is an upper bound on the number of updates needed to approximately maintain $\|p - \tau_k\|$ while iterating over all points in \mathcal{G} in z-order. We next bound the diameter of the cells for a specific associated distance, to subsequently show that this induces a small number of



■ **Figure 5** The partition of cells induced by the point p on the recursively defined cells of the z-curve such that for each cell a single update suffices to ensure a $(1 + \varepsilon)$ -approximation on the distance to p . Note that the cells become larger when further away from p .

cells in the partition. To this end, consider a specific cell C_r and its associated distance $(1 + \varepsilon)^\ell$. As we continued exploring the children of the parent cell C of C_r , there have to be two points $z_1, z_2 \in C$ such that either $\|z_1 - p\| < (1 + \varepsilon)^{\ell-1}$ and $\|z_2 - p\| > (1 + \varepsilon)^\ell$, or $\|z_1 - p\| < (1 + \varepsilon)^\ell$ and $\|z_2 - p\| > (1 + \varepsilon)^{\ell+1}$. By triangle inequality, C has diameter at least

$$\|z_1 - z_2\| \geq \|z_2 - p\| - \|z_1 - p\| > (1 + \varepsilon)^\ell - (1 + \varepsilon)^{\ell-1} = (1 + \varepsilon)^{\ell-1}((1 + \varepsilon) - 1) = (1 + \varepsilon)^{\ell-1}\varepsilon.$$

In the recursive definition of the z-curve, the diameter of a cell decreases at most by a constant factor from parent to child if the child is not a single point. Thus C_j has diameter $\Omega((1 + \varepsilon)^\ell \varepsilon)$ if $|C_j| > 1$. If $|C_j| = 1$, then the Voronoi cell of C_j of the Voronoi diagram of \mathcal{G} has diameter $\Omega((1 + \varepsilon)^\ell \varepsilon)$. Thus, for both cases it holds that there is a square with area $\Omega((1 + \varepsilon)^{2\ell} \varepsilon^2)$ that only contains points from C_j but no other cell $C_{j'}, j' \neq j$.

Recall that the area of a ball of radius R in the L_ρ -norm is equal to $\nu_\rho R^2$, where ν_ρ depends only on the L_ρ -norm and is thus a constant for our purpose. Hence, the area of all points between distance $(1 + \varepsilon)^{\ell-1}$ and $(1 + \varepsilon)^{\ell+1}$ from p is equal to

$$\nu_\rho(1 + \varepsilon)^{2(\ell+1)} - \nu_\rho(1 + \varepsilon)^{2(\ell-1)} = \nu_\rho(1 + \varepsilon)^{2(\ell-1)}((1 + \varepsilon)^4 - 1) = \mathcal{O}((1 + \varepsilon)^{2\ell} \varepsilon).$$

Thus, there can be at most

$$\frac{\mathcal{O}((1 + \varepsilon)^{2\ell} \varepsilon)}{\Omega((1 + \varepsilon)^{2\ell} \varepsilon^2)} = \mathcal{O}\left(\frac{1}{\varepsilon}\right)$$

cells associated with distance $(1 + \varepsilon)^\ell$. Finally, there are at most $\mathcal{O}(\log_{1+\varepsilon} \frac{n}{\varepsilon}) = \mathcal{O}(\frac{1}{\varepsilon} \log \frac{n}{\varepsilon})$ different associated distances, as the minimum non-zero distance is $\Omega(\frac{\delta}{n} \varepsilon)$ and the largest distance is $\mathcal{O}(\delta)$. Consequently, the total number of updates can be bounded by $\mathcal{O}(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon})$. ◀

We now discuss how we explicitly compute the updates. Note that explicitly checking for updates in each node of the traversal of the z-curve is prohibitive. Thus, we have to devise a non-naive way of computing the updates. Indeed, Lemma 9 can be turned into an algorithm.

► **Lemma 10.** *The updates in Lemma 9 can explicitly be computed in time $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \log \frac{n}{\varepsilon}\right)$.*

Proof. Lemma 9 already is constructive, as we associated updates to cells and thereby we can simply perform these updates at the first point of such cells. It therefore only remains to bound the running time of all steps. The running time for exploring the z-curve tree is dominated by the number of cells in the partition, i.e., by the number of updates, multiplied with the running time of deciding whether to explore further or not. If the point $p = \pi_i - \sigma_j$ of Lemma 9 is contained in the currently considered cell, then we have to continue exploring. Otherwise, we can check if all points lie in a $(1 + \varepsilon)^{\ell-1}$ to $(1 + \varepsilon)^{\ell+1}$ distance window for any $\ell \in \mathbb{Z}$ by computing the distance to the closest and furthest point in the cell from p . All of the above steps can be done in $\mathcal{O}(1)$ time. Finally, note that no sorting of the updates is necessary, as exploring the z-curve tree via depth-first search in the order of the z-curve already constructs the updates in sorted order. Hence, the running time of explicitly computing the updates is dominated by the number of updates itself. ◀

We can directly use Lemma 10 to compute the updates for all node weights $w_{i,j}$. However, computing them separately would additionally incur the cost of merging them into a sorted order. We can avoid this sorting step by constructing the updates for all node weights $w_{i,j}$ in parallel using a single DFS on the z-order tree. During the DFS, we maintain a set E of pairs (i, j) for which recursing further is necessary; in the top cell, this is set to $[n] \times [m]$. Then for each cell in the DFS, we need to decide for each pair $(i, j) \in E$ whether a single weight $w_{i,j} = (1 + \varepsilon)^\ell$ suffices to approximate the distance in this cell, which in total takes time $\mathcal{O}(|E|)$. If for $(i, j) \in E$ this is the case, then we add an update of $w_{i,j}$ to $(1 + \varepsilon)^\ell$ for the first point τ in this cell, and remove (i, j) from E for the recursive calls that explore the children of this cell. (We add back (i, j) after the exploration.) This process creates the updates in order and thus we do not have to sort them in a postprocessing step. It follows that the updates for all node weights $w_{i,j}$ can explicitly be computed in time $\mathcal{O}\left(nm \frac{1}{\varepsilon^2} \cdot \log \frac{n}{\varepsilon}\right)$.

Main theorem

Finally, we obtain our main theorem.

► **Theorem 11.** *Assume a data structure for approximate shortest paths in a directed grid graph with N vertices and fixed vertices s, t , supporting updates of an edge weight in time $U(N)$ and $(1 + \varepsilon)$ -approximate s - t -distance queries in time $Q(N)$. We can $(1 + \varepsilon)$ -approximate DTW under translation in L_p -norm in time $\mathcal{O}\left(U(nm) \frac{nm}{\varepsilon^2} \log^2 \frac{n}{\varepsilon} + Q(nm) \frac{m}{\varepsilon^2} \log \frac{n}{\varepsilon}\right)$.*

The proof is deferred to the full version; it follows easily by combining the above arguments.

3.3 Solving the dynamic graph problem

Consider the data structure assumed in Theorem 11 for maintaining shortest paths in a directed grid graph. Das et al. [16] obtain a trade-off of update time $U(N) = \tilde{\mathcal{O}}(N^r)$ and query time $Q(N) = \tilde{\mathcal{O}}(N^{1-r})$ even for *exact* distance queries in directed planar graphs where $r \in [0, \frac{1}{2}]$ is an adjustable parameter, all updates are given in advance, and all edge-weights are non-negative (both are the case in our setting). The aforementioned result improves

bounds due to Fakcharoenphol and Rao [21] and Klein [28], also see [27, 22], by considering the offline setting. For tight conditional lower bounds for the offline setting, we refer to [2]. By setting r such that $N^r = \sqrt{m}$ (which satisfies $r \in [0, \frac{1}{2}]$), we obtain the following corollary.

► **Corollary 12.** *We can $(1+\varepsilon)$ -approximate DTW under translation in \mathbb{R}^2 under the L_p -norm in time $\tilde{O}(nm^{1.5}/\varepsilon^2)$.*

Note that for $n = m$ this becomes $\tilde{O}(n^{2.5}/\varepsilon^2)$. It is straightforward to generalize our algorithm to \mathbb{R}^d for constant d . To this end, we have to replace the 2-dimensional ε -grid and the 2-dimensional space-filling curve by their d -dimensional counterparts and adapt the analysis accordingly. The running time then merely increases with respect to the dependency on ε . See the full version for the proof of the following corollary.

► **Corollary 13.** *We can $(1+\varepsilon)$ -approximate DTW under translation in \mathbb{R}^d under the L_p -norm with $d \in \mathcal{O}(1)$ in time $\tilde{O}(nm^{1.5}/\varepsilon^d)$.*

4 Conclusion and open problems

We give the first rigorous algorithms for Dynamic Time Warping under translation, specifically an exact $\mathcal{O}(n^{2(d+1)})$ -time algorithm for the L_1 norm in \mathbb{R}^d , as well as a $(1+\varepsilon)$ -approximate $\tilde{O}(n^{2.5}/\varepsilon^2)$ -time algorithm for the L_p -norm in \mathbb{R}^2 .

The most interesting open problem is to determine whether under the L_2 -norm, DTW under translation admits an $\tilde{O}(n^2 f(1/\varepsilon))$ -time approximation scheme. In fact, one might be able to improve over our $\tilde{O}(n^{2.5}/\varepsilon^2)$ -time algorithm via purely graph-theoretic improvements for dynamic shortest path algorithms in grid graphs, applying Theorem 11 as a black box. Specifically, we showed how to reduce $(1+\varepsilon)$ -approximate DTW under translation to (approximately) maintaining the s - t distance in a directed grid graph undergoing edge-weight updates. Our precise bound follows from plugging in a data structure due to Das et al. [16] that maintains all *exact* distances. In fact, compared to their setting, our target problem has several important restrictions that may help to design faster algorithms:

- Instead of exact distances, our application only requires a $(1+\varepsilon)$ -approximation.
- Our restriction to directed grid graphs might turn out significantly simpler than general planar digraphs.
- We only ever query the distance between a single source-sink pair.

Finally, if no further algorithmic improvements can be found, can we give improved conditional hardness results, going beyond our reduction from DTW for a fixed translation?

References

- 1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.14.
- 2 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 477–486. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.58.
- 3 Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, volume 51 of *LIPICs*, pages 6:1–6:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.SoCG.2016.6.

- 4 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.
- 5 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'01)*, pages 63–74, 2001.
- 6 Chandrajit L. Bajaj. The algebraic degree of geometric optimization problems. *Discret. Comput. Geom.*, 3:177–191, 1988. doi:10.1007/BF02187906.
- 7 Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. A faster algorithm for the discrete Fréchet distance under translation. *ArXiv preprint*, 2015. arXiv:1501.03724.
- 8 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *Proc. 31st International Conf. Very Large Data Bases (VLDB'05)*, pages 853–864, 2005.
- 9 Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, l) -medians clustering of trajectories using continuous dynamic time warping. In Chang-Tien Lu, Fusheng Wang, Goce Trajcevski, Yan Huang, Shawn D. Newsam, and Li Xiong, editors, *SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, November 3-6, 2020*, pages 99–110. ACM, 2020. doi:10.1145/3397536.3422245.
- 10 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly sub-quadratic algorithms unless SETH fails. In *Proc. 55th Ann. IEEE Symposium on Foundations of Computer Science (FOCS'14)*, pages 661–670, 2014.
- 11 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 79–97. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.15.
- 12 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. doi:10.4230/LIPICs.ESA.2020.25.
- 13 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.
- 14 Maike Buchin. *On the computability of the Fréchet distance between triangulated surfaces*. PhD thesis, Freie Universität Berlin, 2007. PhD Thesis.
- 15 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In Daniel Wichs and Yishay Mansour, editors, *Proc. 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2016)*, pages 9–21. ACM, 2016. doi:10.1145/2897518.2897647.
- 16 Debarati Das, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. A near-optimal offline algorithm for dynamic all-pairs shortest paths in planar digraphs. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3482–3495, 2022. doi:10.1137/1.9781611977073.138.
- 17 Mark de Berg and Atlas F. Cook. Go with the flow: The direction-based Fréchet distance of polygonal curves. In Alberto Marchetti-Spaccamela and Michael Segal, editors, *Theory and Practice of Algorithms in (Computer) Systems - First International ICST Conference, TAPAS 2011, Rome, Italy, April 18-20, 2011. Proceedings*, volume 6595 of *Lecture Notes in Computer Science*, pages 81–91. Springer, 2011. doi:10.1007/978-3-642-19754-3_10.
- 18 Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity between Curves. *Journal of Mathematical Imaging and Vision*, 27(3):203–216, April 2007. doi:10.1007/s10851-006-0647-0.

20:16 (Approximation) Algorithms for Dynamic Time Warping Under Translation

- 19 Alon Efrat, Piotr Indyk, and Suresh Venkatasubramanian. Pattern matching for sets of segments. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pages 295–304, 2001.
- 20 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 1994.
- 21 Jittat Fakcharoenphol and Satish Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *J. Comput. Syst. Sci.*, 72(5):868–889, 2006. doi:10.1016/j.jcss.2005.05.007.
- 22 Pawel Gawrychowski and Adam Karczmarz. Improved bounds for shortest paths in dense distance graphs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPICs*, pages 61:1–61:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ICALP.2018.61.
- 23 Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4):50:1–50:17, 2018. doi:10.1145/3230734.
- 24 Joachim Gudmundsson, André van Renssen, Zeinab Saeidi, and Sampson Wong. Translation invariant Fréchet distance queries. *Algorithmica*, 83(11):3514–3533, 2021. doi:10.1007/s00453-021-00865-0.
- 25 Sarel Har-Peled. *Geometric approximation algorithms*, chapter Fréchet distance: How to walk your dog, pages 383–412. American Mathematical Society, 2017. Online chapter.
- 26 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein structure–structure alignment with discrete Fréchet distance. *J. Bioinformatics and Computational Biology*, 6(01):51–64, 2008.
- 27 Haim Kaplan, Shay Mozes, Yahav Nussbaum, and Micha Sharir. Submatrix maximum queries in monge matrices and partial monge matrices, and their applications. *ACM Trans. Algorithms*, 13(2):26:1–26:42, 2017. doi:10.1145/3039873.
- 28 Philip N. Klein. Multiple-source shortest paths in planar graphs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 146–155. SIAM, 2005. URL: <http://dl.acm.org/citation.cfm?id=1070432.1070454>.
- 29 Marvin Künnemann and André Nusser. Polygon placement revisited: (Degree of Freedom + 1)-SUM hardness and an improvement via offline dynamic rectangle union. In *Proc. 33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, 2022. To appear.
- 30 William Kuszmaul. Dynamic time warping in strongly subquadratic time: Algorithms for the low-distance regime and approximate evaluation. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 80:1–80:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.80.
- 31 Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Approximating the integral Fréchet distance. *Comput. Geom.*, 70-71:13–30, 2018. doi:10.1016/j.comgeo.2018.01.001.
- 32 Axel Mosig and Michael Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Computational Geometry: Theory and Applications*, 30(2):113–127, 2005.
- 33 Mario E. Munich and Pietro Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*, pages 108–115. IEEE Computer Society, 1999. doi:10.1109/ICCV.1999.791205.
- 34 Mark H. Overmars and Chee-Keng Yap. New upper bounds in klee’s measure problem. *SIAM J. Comput.*, 20(6):1034–1045, 1991. doi:10.1137/0220065.
- 35 Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Elastic Translation Invariant Matching of Trajectories. *Machine Learning*, 58(2):301–334, February 2005. doi:10.1007/s10994-005-5830-9.

- 36 Carola Wenk. *Shape matching in higher dimensions*. PhD thesis, Freie Universität Berlin, 2003. PhD Thesis.
- 37 Carola Wenk, Randall Salas, and Dieter Pfoser. Addressing the need for map-matching speed: Localizing globalb curve-matching algorithms. In *18th International Conference on Scientific and Statistical Database Management, SSDBM 2006, 3-5 July 2006, Vienna, Austria, Proceedings*, pages 379–388. IEEE Computer Society, 2006. doi:10.1109/SSDBM.2006.11.
- 38 Rex Ying, Jiangwei Pan, Kyle Fox, and Pankaj K. Agarwal. A simple efficient approximation algorithm for dynamic time warping. In Siva Ravada, Mohammed Eunus Ali, Shawn D. Newsam, Matthias Renz, and Goce Trajcevski, editors, *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2016, Burlingame, California, USA, October 31 - November 3, 2016*, pages 21:1–21:10. ACM, 2016. doi:10.1145/2996913.2996954.
- 39 Yu Qiao and M. Yasuhara. Affine Invariant Dynamic Time Warping and its Application to Online Rotated Handwriting Recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 905–908, August 2006. ISSN: 1051-4651. doi: 10.1109/ICPR.2006.228.