The k-Fréchet distance revisited and extended

Hugo A. Akitaya[1], Maike Buchin[2], Leonie Ryvkin[3], and Jérôme Urhausen[4]

1 **Department of Computer Science, Tufts University**
`hugo.alves_akitaya@tufts.edu`

2 **Faculty of Computer Science, TU Dortmund**
`maike.buchin@tu-dortmund.de`

3 **Department of Mathematics, Ruhr-Universität Bochum**
`leonie.ryvkin@rub.de`

4 **Department of Information and Computing Sciences, Universiteit Utrecht**
`j.e.urhausen@uu.nl`

Abstract

We recently introduced a new distance measure for polygonal curves, the k-Fréchet distance. It bridges between Hausdorff distance and weak Fréchet distance, and allows us to compare objects of rearranged pieces such as chemical structures or hand-written characters. Here we distinguish between two variants of the k-Fréchet distance, the *cut distance* and the *cover distance*. For the first one, we presented an NP-hardness proof at EuroCG 2018 [6]. The approximation algorithm we presented in that paper, however, only works for the cover distance. Here, we now prove NP-hardness of the cover version and present an XP- as well as an FPT-algorithm for this version.

1 Introduction

During the last decades, several methods for comparing geometrical shapes have been studied in a variety of applications, for example analysing geographic data, such as trajectories, or comparing chemical structures, e.g., protein chains or human DNA. The (weak) Fréchet distance has been well-studied in the past twenty years since it has proven to be helpful in several of the mentioned applications. The Hausdorff distance, another similarity measure, has also proven useful in applications and can be computed more efficiently than the Fréchet distance [2]. However, it provides us with less information by taking only the overall shape of curves into consideration, not how they are traversed.

We introduce both variants of the k-Fréchet distance as distance measures in between Hausdorff and weak Fréchet distance. They allow us to compare shapes consisting of several parts. Variants of partial curve matching using the Fréchet distance have been studied before [4, 5, 9]. Gheibi et al. used the weak Fréchet distance and minimized the length of the subcurves on which backtracking is necessary [8]. For the cover distance, we cover the input curves by at most k (possibly overlapping) subcurves each and ask for a matching of the subcurves such that each matched pair of subcurves has at most weak Fréchet distance ε (for given $\varepsilon > 0$). For the cut version, the subcurves may not overlap. This implies that, as opposed to the cover distance, both curves have to be cut into the same number of subcurves.

The paper is organized as follows: after recalling some basic definitions we formally define both variants of our k-Fréchet distance and state decision and optimization problems. In Section 2 we briefly discuss NP-hardness of the cover distance and sketch the proof. In Section 3 we present an FPT algorithm for the cover distance.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

1.1 Definitions

Recall the weak Fréchet distance [1], a well-known measure for curves, which is defined as follows: For curves $P, Q\colon [0,1] \to [0,1]$, the weak Fréchet distance is given by

$$\delta_{\mathrm{wF}}(P,Q) = \inf_{\sigma,\tau} \max_{t\in[0,1]} \|P(\sigma(t)) - Q(\tau(t))\|,$$

where the reparametrisations $\sigma, \tau\colon [0,1] \to [0,1]$ range over all continuous surjective functions.

A well-known characterisation which is key to efficient algorithms for computing the weak Fréchet distance [1] uses the free space diagram. First we recall the free space F_ε:

$$F_\varepsilon(P,Q) = \{(t_1,t_2) \in [0,1]^2\colon \|P(t_1) - Q(t_2)\| \leqslant \varepsilon\}.$$

The free space diagram puts this information into an $(n \times m)$-grid, where n and m are the number of segments in P and Q, respectively.

The weak Fréchet distance of two curves is at most a given value ε if there exists a continous path through the free space containing $(0,y), (1,y'), (x,0)$ and $(x',1)$ for some $x, x', y, y' \in [0,1]$. The Hausdorff distance δ_{H} can be characterised as the free space projecting surjectively onto both parameter spaces.

We define further terms connected to the free space diagram below: A *component* of a free space diagram is a connected subset $c \subseteq F_\varepsilon(P,Q)$. A set S of components *covers* a set $I \subseteq [0,1]_P$ of the parameter space (corresponding to the curve P) if I is a subset of the projection of S onto said parameter space, i.e., $\forall x \in I\colon \exists c \in S, y \in [0,1]_Q\colon (x,y) \in c$. Covering on the second parameter space is defined analogously.

▶ **Definition 1.1.** For polygonal chains P, Q we define the cut version of the k-Fréchet distance as

$$\delta_{\mathrm{cut}}(k,P,Q) = \inf_{\sigma,\tau} \max_{t\in[0,1]} \|P(\sigma(t)) - Q(\tau(t))\|,$$

where now $\sigma, \tau\colon [0,1] \to [0,1]$ range over all surjective functions that are piecewise defined, such that the images of the continuous parts partition the curve into at most k pieces. Note that σ and τ have to consist of the same number of continous pieces, namely at most k many.

That is, we cut the curves P and Q into at most k pieces or subcurves each, such that two resembling subcurves have small weak Fréchet distance. In the free space diagram, we can insert the cuts on our curves as horizontal and vertical grid lines. For every row and column of this "cutting grid", we need to select exactly one cell. The cell corresponds to a matched pair of subcurves and therefore needs to contain (a part of) a free space component that projects surjectively onto both subcurves. Next we define the cover distance:

▶ **Definition 1.2.** For polygonal chains P, Q we define the cover version of the k-Fréchet distance as

$$\delta_{\mathrm{cover}}(k,P,Q) = \inf_{\sigma,\tau} \max_{t\in[0,1]} \|P(\sigma(t)) - Q(\tau(t))\|,$$

where σ, τ range over all surjective functions that are piecewise defined, such that the images of the (at most k many) continuous parts may overlap but their union equals the curve.

Thus we define the cover distance as the minimal ε such that there is a set of at most k components of $F_\varepsilon(P,Q)$ that covers both parameter spaces. In other words, we cover the curves P and Q by at most k pieces (i.e., subcurves) such that there is a matching of the pieces where two matched subcurves have small weak Fréchet distance.

For the decision problem of both distance measures, we ask whether the weak Fréchet distance between pieces can be bounded by a given value ε, where the number of subcurves is upper bounded by k. For fixed ε, we want to minimize k (optimization problem).

By definition both variants lie in between Hausdorff and weak Fréchet distance:

$$\delta_{\mathrm{H}}(P,Q) \leq \delta_{\mathrm{cover}}(k,P,Q) \leq \delta_{\mathrm{cut}}(k,P,Q) \leq \delta_{\mathrm{wF}}(P,Q).$$

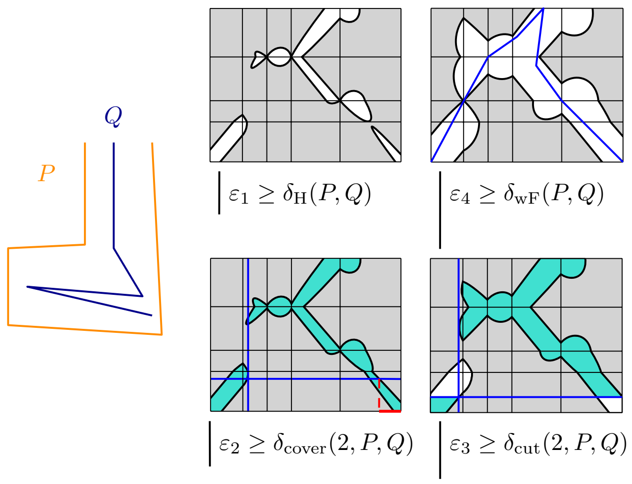

Q
P
$\varepsilon_1 \geq \delta_{\mathrm{H}}(P,Q)$
$\varepsilon_4 \geq \delta_{\mathrm{wF}}(P,Q)$
$\varepsilon_2 \geq \delta_{\mathrm{cover}}(2,P,Q)$
$\varepsilon_3 \geq \delta_{\mathrm{cut}}(2,P,Q)$

Figure 1 Comparison of distance measures. In the bottom left diagram, two components are sufficient to cover the parameter spaces, but cutting does not work, because by choosing the bottom left and top right cell the red section on the bottom parameter space would not be covered.

2 NP-hardness

In [6], Buchin and Ryvkin proved that the cut distance is NP-complete by reducing from Minimum Common String Partition (MCSP). For the cover distance, we reduce from the following variant of 3-SAT, which was proven to be NP-complete by de Berg and Khosravi [3].

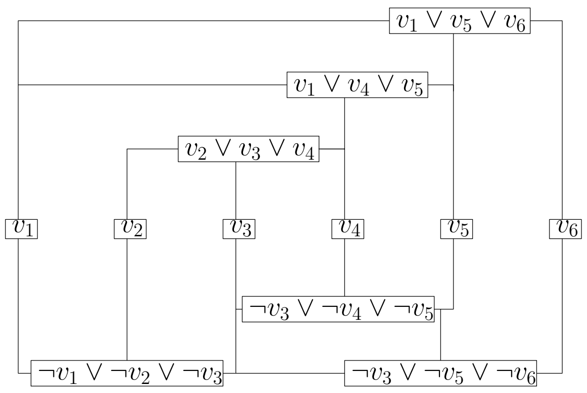
Rectilinear monotone planar 3-SAT:
INPUT: 3-SAT formula with strictly positive and strictly negative clauses, embedded as a graph with only rectilinear, non-crossing edges; variables are drawn as vertices on a horizontal line, positive clauses are vertices drawn above this line, negative clauses are drawn below;
OUTPUT: "Yes" if there exists a satisfying assignment for the variables, "No" otherwise.

Our goal is to construct curves that mimic our input graph (see Figure 2) and show that in the free space resulting from these curves we can find a covering selection of components of size k iff there exists a satisfying assignment for the underlying 3-SAT formula.

2.1 Construction

The construction is intricate, so we only give a brief overview here. A full version of this paper, named "The k-Fréchet distance", will be made available on arXiv.

Overall we create wires and clause gadgets to represent variables and clauses. They are connected as the given embedding of the 3-SAT instance. Wire gadgets allow a boolean



$v_1 \vee v_5 \vee v_6$
$v_1 \vee v_4 \vee v_5$
$v_2 \vee v_3 \vee v_4$
v_1
v_2
v_3
v_4
v_5
v_6
$\neg v_3 \vee \neg v_4 \vee \neg v_5$
$\neg v_1 \vee \neg v_2 \vee \neg v_3$
$\neg v_3 \vee \neg v_5 \vee \neg v_6$

Figure 2 Instance of rectilinear monotone planar 3-SAT.

choice that is propagated consistently throughout the wire. Clause gadgets test whether at least one incoming wire carries an appropriate choice.

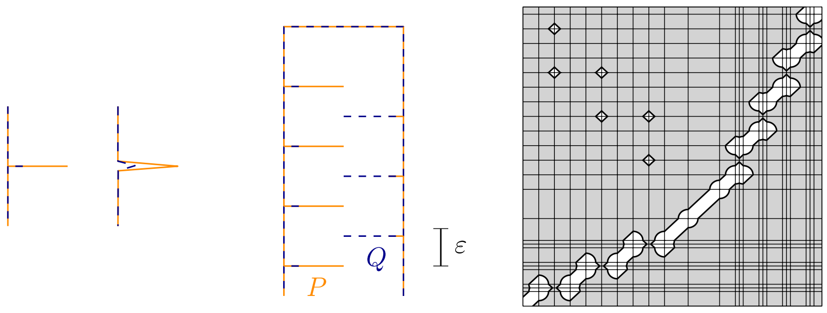


Figure 3 (Left) A spike and a small perturbation of it. (Right) The wire gadget and its corresponding free space diagram. Note that we connected the curves to give a small example, but the horizontal segment on top is not part of the gadget itself.

Figure 3 shows a wire gadget. Both the yellow and the blue curves run along the sides (the vertical parts of the curves, which we call *base curves*) and form *spikes*. The value ε is chosen such that two adjacent spikes are just within distance ε. It follows that the spikes induce components in the free space diagram that form a staircase. We say that a spike s is *covered* by an adjacent spike t of the other curve if the component of the free space diagram that covers the two intervals induced by these spikes is chosen for the covering selection. We choose k such that each blue spike in any gadget can only be covered by one single adjacent yellow spike. The choice for one blue spike must be consistent along a wire and encodes the assignment of the corresponding variable.

Of course, we need a number of other gadgets, too. The wires correspond to edges in the rectilinear monotone planar 3-SAT instance, but to draw them coherently we need to make sure we can make 90° turns (so called *bends*) and do T-crossings, i.e., *split* a wire into two. Last but not least we need to build a *clause gadget* where three wires connect. In Figure 4, we show a bend and a clause gadget, connected through wires. The only basic gadget not shown in Figure 4 is the split, which looks similar to the clause.

Additionally, we need to make sure that both curves are connected and follow the embedding of the input graph G. In order to do so, we establish a number of other gadgets which are explained in the full version of this paper, where we also explain how to connect all gadgets and build the curves.

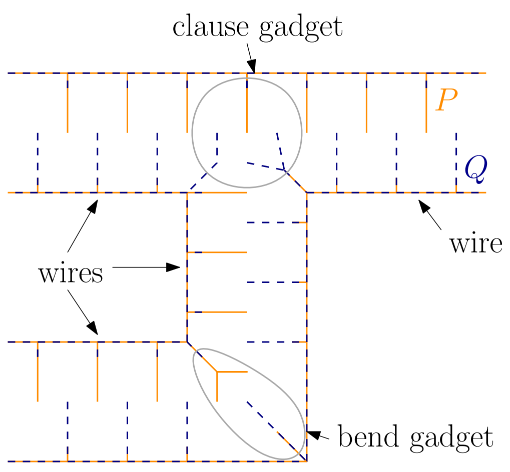


Figure 4 Small example part of the construction consisting of wires, a bend and a clause gadget.

2.2 Correctness

▶ **Theorem 2.1.** *For given polygonal curves P and Q, integer k, and $\varepsilon > 0$, it is NP-complete to decide whether $\delta_{\text{cover}}(k, P, Q) \leq \varepsilon$.*

Due to limited space, we only sketch the proof. We set k such that we can cover every blue spike exactly once, so the choice made for one spike is propagated throughout its wire (and corresponding gadgets). Therefore an assignment implies a unique selection of components and vice versa; following the choices made for the blue spikes we can derive a variable assignment for the underlying 3-SAT formula.

We can test in polynomial time whether the union of a selection of components covers the parameter spaces. Thus the problem of deciding the cover distance lies in NP.

3 Algorithmic approaches

3.1 Earlier results

In [6], Buchin and Ryvkin presented a straight-forward XP-algorithm that simply checked for all possible selections of size k, whether one of them covered both parameter sizes, which takes $\mathcal{O}(k \cdot n^{2k})$ time. As obvious, this approach does not apply to the cut version of the k-Fréchet distance: a selection of components does not imply cutting the curves, the subcurves may overlap. Cutting one curve affects the possible cuts of the other one, so determining correct cuts even for a given selection of components is non-trivial.

In the same paper the authors also gave an approximation algorithm for the optimization problem: by sweeping twice to determine the optimal selections to cover either parameter space the algorithm outputs a selection of components that is at worst doubly the size of an optimal one. Again, this only works for the cover distance, not for cutting.

3.2 Fixed-parameter tractability

Next we present an algorithm for deciding whether $\delta_{\text{cover}}(k, P, Q) \leq \varepsilon$ for given ε and k. The runtime of our algorithm is polynomial in the complexity of our curves P and Q, but exponential in the two parameters k (the selection size) and z (the neighborhood complexity).

We define the *neighborhood complexity* z of P and Q as the maximum number of segments of one curve that intersect with the ε-neighborhood of any point of the other curve. Now, in the free space diagram each horizontal and each vertical line intersects at most z components.

First of all, we build two bounded search trees T_P and T_Q (as described in Chapter 3 of [7]), see Figure 5 below. Consider the tree T_P. The root is labelled by the left boundary point of the parameter space of P (we assume without loss of generality that the bottom boundary of the free space diagram corresponds to P). Now we use a sweep line initialized at the left boundary of the free space diagram. We assign a node in the tree to all components intersecting the sweep line, so the root has as many children as there are components touching the left boundary. Whenever the sweep line is tangent to a component C, said component either becomes *active* (sweep line touches the leftmost point) or *inactive* (its rightmost point is touched). Becoming active does not immediately affect the tree; becoming inactive results in new entries in the tree: for every node labelled C we insert a child for each currently active component. The next time a compone By definition a node can never have more than z children. Also, with every node we store its depth (the root has depth 0) and stop assigning children at depth k - or as soon as a component touches the right boundary of the free space diagram. If a leaf v_l corresponds to a component touching the right boundary, the path from the root to v_l encodes a *feasible* selection of components for T_P. The tree T_Q is built analogously by sweeping from bottom to top.

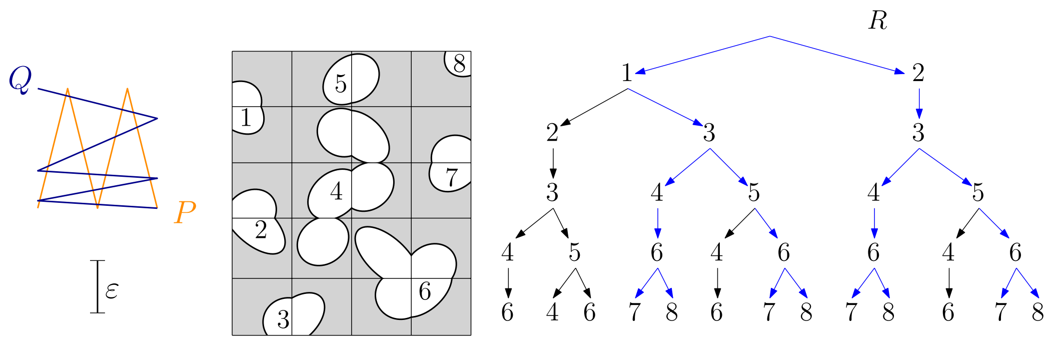


Figure 5 Curves P, Q, their free space diagram and the tree T_P. Feasible paths are marked blue.

We store the feasible selections obtained from T_P and T_Q in sorted lists L_P^T and L_Q^T. For each pair of selections $S_{P,i}$, $S_{Q,j}$, where $1 \leq i, j \leq z^k$, we test whether $|S_{P,i} \cup S_{Q,j}| \leq k$ and output this union if the answer is positive. The proof of the following theorem can also be found in the full version.

▶ **Theorem 3.1.** *The algorithm described above returns a selection S of k components in the free space that surjectively projects onto both parameter spaces if and only if such a selection exists. Therefore it decides whether $\delta_{\text{cover}}(k, P, Q) \leq \varepsilon$ in time $\mathcal{O}(nk + kz^{2k})$.*

References

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

2 Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2004.

3 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *Internat. J. Comput. Geom. Appl.*, 22(3):187–205, 2012. doi:10.1142/S0218195912500045.

4 Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proc. 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 645–654, 2009. URL: `http://dl.acm.org/citation.cfm?id=1496770.1496841`.

5 Maike Buchin, Anne Driemel, and Bettina Speckmann. Computing the Fréchet distance with shortcuts is NP-hard. In *Proc. 30th Annual Symposium on Computational Geometry*, SOCG'14, pages 367:367–367:376, 2014. `doi:10.1145/2582112.2582144`.

6 Maike Buchin and Leonie Ryvkin. The k-Fréchet distance of polygonal curves. In *34th European Workshop on Computational Geometry (EuroCG), Book of Abstracts*, page 4, 2018. URL: `conference.imp.fu-berlin.de/eurocg18/`.

7 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, Cham, 2015. `doi:10.1007/978-3-319-21275-3`.

8 Amin Gheibi, Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Minimum backward Fréchet distance. pages 381–388, 11 2014. `doi:10.1145/2666310.2666418`.

9 Christian Scheffer. More flexible curve matching via the partial Fréchet similarity. *Int. J. Comput. Geometry Appl.*, 26:33–52, 2016.