Approximating the Fréchet Distance in Graphs with Low Highway Dimension*

Anne Driemel[1] and Marena Richter[1]

1 **Department of Computer Science, University of Bonn, Germany**

Abstract

In this paper, we study algorithms for the discrete Fréchet distance in graphs with low highway dimension. We describe a $(\frac{5}{3} + \varepsilon)$-approximation algorithm for the Fréchet distance between a shortest path P with n vertices and an arbitrary walk Q with m vertices in a graph $G = (V, E)$. The algorithm makes use of a collection of sparse shortest paths hitting sets which are precomputed for the graph G. After preprocessing, the algorithm has running time $\mathcal{O}\left(n \log D + m(h \log h \log D)^2\right)$, where h is the highway dimension and D is the diameter of G. The preprocessing for the graph is polynomial in $|G|$ and $1/\log(1+\varepsilon)$ and uses $\mathcal{O}\left(|V| \log D(1/\log(1+\varepsilon) + h \log h)\right)$ space.

1 Introduction

The notion of the Fréchet distance between polygonal curves was introduced to Computational Geometry by Alt and Godau in 1992 [5]. They also gave a $\mathcal{O}(n^2 \log n)$ algorithm to compute the continuous Fréchet distance between two curves with $\mathcal{O}(n)$ vertices in Euclidean metric spaces of fixed dimension. Bringmann showed that neither the discrete nor the continuous Fréchet distance between two curves can be computed in time $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$ unless the orthogonal vectors hypothesis fails [9] and there are known algorithms showing that this lower bound is tight in the discrete [4] and the continuous case [10]. There are various faster exact and approximation algorithms known in specialized settings for the continuous [6, 7, 11, 12, 13, 14] and discrete Fréchet distance [7, 15, 17, 19]. Some of these algorithms work with a preprocessing that stores one curve such that computing the Fréchet distance to any other curve can be done efficiently [13, 15, 17, 19]. All of the mentioned algorithms require that the curves are embedded in some sort of underlying metric space.

We consider the discrete Fréchet distance between walks in a graph with respect to the shortest path metric. This can for example be used to determine similarities between trajectories in street networks, which has been a question of interest in the past [8, 18]. Driemel, van der Hoog and Rotenberg showed that for this variant of the problem the near-quadratic conditional lower bound still holds [16]. They also study approximation algorithms for the setting that one of the walks is κ-straight, that is a near-shortest path. A path is κ-straight if any subpath between any two vertices p and q along the path has length at most κ times the shortest path distance from p to q. In particular, a shortest path is κ-straight for $\kappa = 1$. They show that one can compute a $(1+\varepsilon)$-approximation of the Fréchet distance between a κ-straight path P and any walk Q in a planar graph G in $\mathcal{O}\left(|G| \log |G| / \sqrt{\varepsilon} + |P| + \frac{\kappa}{\varepsilon} |Q|\right)$ time and they give a $(\kappa+1)$-approximation algorithm, with running time in $\mathcal{O}\left((|P| + |Q|) \log^{3+o(1)} |G|\right)$, after preprocessing. For general graphs, the

* This work was funded by 390685813 (Germany's Excellence Strategy – EXC-2047/1: Hausdorff Center for Mathematics); 416767905; and funded by the Lamarr Institute for Machine Learning and Artificial Intelligence lamarr-institute.org.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. 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.

second algorithm has running time in $\mathcal{O}\left((|P| + |Q|) \cdot T(G) \cdot \log D\right)$, where $T(G)$ denotes the time for a distance query in G.

In this paper, we focus on the case where the graph $G = (V, E)$ is not necessarily planar, but has low highway dimension, a property which has been studied in the context of road networks before [2]. We give an algorithm that computes a $(\frac{5}{3} + \varepsilon)$-approximation to the discrete Fréchet distance between a shortest path P and any walk Q in G. After preprocessing the graph, the running time of the algorithm is in $\mathcal{O}\left(|P| \log D + |Q| \left(h \log h \log D\right)^2\right)$, where h is the highway dimension of the graph and D is the diameter of G. The preprocessing is polynomial in $|V|$ and $1/\log(1+\varepsilon)$ and uses $\mathcal{O}\left(|V| \log D(1/\log(1+\varepsilon) + h \log h)\right)$ space.

1.1 Highway Dimension

Abraham, Delling, Fiat, Goldberg and Werneck introduced multiple definitions of the highway dimension over the years [3, 1, 2]. We work with the latest definition from 2016 [2].

The intuition behind a low highway dimension is that there exists a small set of vertices ("hubs") such that for any point in the graph every shortest path to a destination far away visits at least one of these hubs. Abraham et al. argue that this is a realistic model of road networks [2]. A low highway dimension is especially helpful for shortest path computations. We use these hitting sets of long shortest paths for Fréchet distance queries.

Let $G = (V, E)$ be a graph with non-negative integer edge weights ℓ that satisfy the triangle inequality and can all be expressed in a word of $\Theta(\log |V|)$ bits. A *walk* in G is a sequence of vertices with an edge between any two successive vertices. A *path* is a walk where every vertex is visited at most once. We assume that shortest paths in G are unique.

Let P be a shortest path with weight $\ell(P) > r$ for some value $r > 0$. Denote by $V(P)$ the set of vertices in P. In [2], Abraham et al. call a shortest path P' an r*-witness* for a shortest path P if $\ell(P') > r$ and P is either equal to P' or it arises from P' by deleting one or both end vertices of P'. All shortest paths that have an r-witness are called r*-significant*. This means that also single vertices can be r-significant. Denote by $\mathcal{P}_r$ all r-significant paths.

▶ **Definition 1.1** (Highway dimension [2]). The *highway dimension* h of the graph $G = (V, E)$ is the smallest integer such that for any real value $r > 0$ and $v \in V$ there exists a set $H \subseteq V$ with $|H| \leq h$ and $H \cap V(P) \neq \emptyset$ for all r-significant paths P with an r-witness P' satisfying $\operatorname{dist}(v, P') := \min_{w \in V(P')} \operatorname{dist}(v, w) \leq 2r$.

The sets H exist separately for every vertex and radius. Abraham et al. give a related definition of sparse hitting sets:

▶ **Definition 1.2** (Sparse Shortest Path Hitting Set (SPHS) [2]). For $r > 0$ an (h, r)*-SPHS* is a set $C \subseteq V(G)$ such that $|B_{2r}(v) \cap C| \leq h$ for all $v \in V(G)$ and $V(P) \cap C \neq \emptyset$ for all $P \in \mathcal{P}_r$, where $B_{2r}(v)$ is the set of vertices in G that have distance at most $2r$ to v.

It can be shown that there always exists an (h, r)-SPHS in a graph with highway dimension h [2]. Note that the set $B_{2r}(v) \cap C$ is very similar to the set H for v and r in the definition of the highway dimension but it might not hit all necessary r-significant paths even though C hits all these paths. One can extend this definition even further in the following way:

▶ **Definition 1.3** (μ-multiscale SPHS). For $\mu > 1$ a μ*-multiscale* SPHS with value h of G is a collection of sets C_i for $0 \leq i \leq \left\lceil \frac{\log D}{\log \mu} \right\rceil$, where C_i is a (h, μ^{i-1})-SPHS and $D := \max_{v,w \in V(G)} \operatorname{dist}(v, w)$ is the diameter of G.

In [2] the definition of a multiscale SPHS matches our definition of a 2-multiscale SPHS. Note that $\log D \in \mathcal{O}(\log |V|)$ because all edge weights can be expressed in a word of $\Theta(\log |V|)$

bits. Computing an (h, r)-SPHS in a graph with highway dimension h can be NP-hard. However, we can compute an approximation in polynomial time (Theorem 8.2 in [2]), which leads to the following theorem:

▶ **Theorem 1.4.** *In a graph with highway dimension h, we can compute a μ-multiscale SPHS with value $\mathcal{O}(h \log h)$ in running time polynomial in size(G) and $(\log \mu)^{-1}$.*

Using 2-multiscale SPHS, one can create a fast distance oracle in G as it is discussed in [2]:

▶ **Theorem 1.5** (see Theorem 8.3 in [2]). *With a polynomial-time preprocessing and $\mathcal{O}(|V| \log D \cdot h \log h)$ space we can preprocess a graph $G = (V, E)$ with highway dimension h such that a distance query between any two vertices takes $\mathcal{O}(h \log h \log D)$ time.*

1.2 Fréchet Distance

We follow [16] in our definition of the discrete Fréchet distance in a graph. The discrete Fréchet distance is a similarity measure between two walks in a graph. Assume we are given a graph G with a metric weight function w. Let $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_m)$ be walks in G. We denote by $[n] \times [m] \subset \mathbb{N} \times \mathbb{N}$ the integer lattice of n by m integers and say that an ordered sequence F of pairs in $[n] \times [m]$ is an *xy-monotone discrete walk* if for every consecutive pair $(i, j), (k, l) \in F$, we have $k \in \{i, i + 1\}$ and $l \in \{j, j + 1\}$.

▶ **Definition 1.6** (see Section 2 in [16]). The *strong discrete Fréchet distance* of two walks $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_m)$ is the minimum over the maximum pairwise distance of any xy-monotone discrete walk F from $(1, 1)$ to (n, m):

$$D_{\mathcal{F}}(P, Q) := \min_{F} \max_{(i,j)\in F} \text{dist}(p_i, q_i).$$

For brevity we just call this the Fréchet distance of P and Q. One can verify that the Fréchet distance satisfies the triangle inequality.

Given two walks P, Q and some real value d, we define a $|Q| \times |P|$ matrix M which we call the *free-space matrix* M_d. The i-th column of M_d corresponds to the i-th vertex in P and the j-th row corresponds to the j-th vertex in Q. We assign to each matrix cell $M_d[i, j]$ the integer -1 if $\text{dist}(p_i, q_j) \leq d$, and 0 if $\text{dist}(p_i, q_j) > d$.

The Fréchet distance between two walks P and Q is at most d, iff there exists an xy-monotone discrete walk F from $(1, 1)$ to (n, m) such that $\forall (i, j) \in F$ we have $M_d[i, j] = -1$.

2 Algorithm

On a high level, our algorithm first computes a simplification of the shortest path P using a certain SPHS. It then runs a BFS on a free space matrix between the walk Q and the simplification of P to approximately determine, whether $D_{\mathcal{F}}(P, Q) \leq \delta$. In the analysis, we show that the distance between the chosen simplification and P can be bounded from above, which then bounds the approximation factor of the algorithm. Then, we prove that the BFS does not visit too many vertices making use of the fact that almost all vertices on the simplification belong to the same SPHS. This then bounds the runtime of our algorithm. In the end we choose the values for the multiscale SPHS, the simplification and the free space matrix to achieve the desired approximation factor.

Throughout the rest of the paper, let $P = \langle p_1, \ldots, p_n \rangle$ be a shortest path in G and let $Q = \langle q_1, \ldots, q_m \rangle$ be an arbitrary walk in G. We denote by $|P|$ the number of vertices in P.

First, we focus on the case, where we are already given a certain SPHS and see that we can compute an approximation of the Fréchet distance between P and Q quite fast, using only a subset of the vertices of P.

Let H_δ be a (h', δ)-SPHS for some $h' \in \mathbb{N}$ and $\delta \geq 0$. Define by P^δ the subsequence of vertices in P, where we start in p_1, then only visit the points of $P \cap H_\delta$ and end in p_n. Note that P^δ is not necessarily a walk in G. However, we treat it like a walk by adding artificial edges in between any two consecutive points of P^δ with costs equal to their distance.

▶ **Lemma 2.1.** *It holds that $D_{\mathcal{F}}\left(P, P^\delta\right) \leq \frac{\delta}{2}$.*

Proof. Denote by a_i the index of the vertex p_i in P^δ. We define an xy-monotone path through the free space matrix $M_{\delta/2}$ of P and P^δ only visiting entries with value -1 and visiting all the tuples (i, a_i) for $p_i \in P^\delta$. We start with $(1, 1)$. Let p_i and p_j be consecutive in P^δ. We now want to define an xy-monotone subpath from (i, a_i) to (j, a_j). We start with (i, a_i). If $j = i + 1$, we can take (j, a_j) as the next tuple, which is a legal step and we are done.

Now suppose that $j > i + 1$. Then, p_i and p_j are not consecutive in P. Assume $\mathrm{dist}(p_i, p_j) > \delta$. Denote by $P[i, j]$ the subpath of P starting in p_i and ending in P_j. Observe that $P[i, j]$ is a shortest path. If we delete the two outer vertices of this subpath, we either have another subpath or a single vertex. Denote this subpath or singleton by $\tilde{P}$. The shortest path $\tilde{P}$ is δ-significant with $P[i, j]$ as a δ-witness. Hence, a vertex of $\tilde{P}$ has to be contained in H_δ because it is a hitting set for all δ-significant shortest paths. This is a contradiction to p_i and p_j being consecutive on P^δ and hence $\mathrm{dist}(p_i, p_j) \leq \delta$ must hold.

Let i' be the largest index such that $\mathrm{dist}(p_i, p_{i'}) \leq \frac{\delta}{2}$. This means that $\mathrm{dist}(p_{i'+1}, p_j) \leq \frac{\delta}{2}$ because P is a shortest path. So, we define the following xy-monotone subwalk:

$$(i, a_i), (i + 1, a_i), \ldots, (i', a_i), (i' + 1, a_j), \ldots, (j - 1, a_j), (j, a_j).$$

The distance of all tuples is at most $\frac{\delta}{2}$ and hence their entries in $M_{\delta/2}$ are -1. Hence, if we combine all such subpaths, we end up with an xy-monotone walk through $M_{\delta/2}$ only visiting entries with value -1, implying that the Fréchet distance between P and P^δ is $\leq \frac{\delta}{2}$. ◀

Using Lemma 2.1 and the triangle inequality of the Fréchet distance, we get the following:

▶ **Lemma 2.2.** *If $D_{\mathcal{F}}\left(P^\delta, Q\right) > \alpha$ for any $\alpha \geq 0$, then $D_{\mathcal{F}}\left(P, Q\right) > \alpha - \frac{\delta}{2}$.*

▶ **Proposition 2.3.** *Let H_δ be a (h', δ)-SPHS of G, let P be a shortest path and let P^δ be given. Assume that a distance query in G takes time $T(G)$. Then, we can decide in time $\mathcal{O}\left(mh'T(G)\right)$ if $D_{\mathcal{F}}\left(P^\delta, Q\right) \leq \alpha$ for any $0 \leq \alpha \leq 2\delta$.*

Proof. The algorithm performs an implicit breadth first search through the non-zero entries of the free space matrix M_α between P^δ and Q and checks if (n, m) can be reached. This means that we only compute $\mathrm{dist}(p_i, q_j)$, if the tuple (i, j) is considered in the BFS.

Note that only non-zero entries of M_α get added to the queue and every element in the queue has at most three legal successors. Hence, for the runtime it suffices to bound the number of non-zero entries in M_α. Let q be a vertex in Q. H_δ being a (h', δ)-SPHS implies $|B_\alpha(q) \cap H_\delta| \leq |B_{2\delta}(q) \cap H_\delta| \leq h'$. So, the inner vertices of P^δ with distance at most α to q all lie in this set. Adding p_1 and p_n this yields that there are at most $h' + 2$ non-zero entries in the row corresponding to q and at most $m(h' + 2)$ non-zero entries in M_α in total. For each of them we have at most three distance oracle calls. So, the BFS takes $\mathcal{O}\left(mh'T(G)\right)$. ◀

This gives us all the necessary tools to prove the following theorem:

▶ **Theorem 2.4.** *Let $G = (V, E)$ be a graph with a metric weight function and highway dimension h and let $\varepsilon > 0$. Suppose a distance query in G takes $\mathcal{O}(T(G))$ time using $\mathcal{O}(S(G))$ space. After preprocessing G in time polynomial in $|V|$ and $1/\log(1+\varepsilon)$, we can decide for any shortest path P with n vertices, any walk Q with m vertices and any $\delta > 0$, whether $D_{\mathcal{F}}(P, Q) \leq (\frac{5}{3}+\varepsilon)\delta$ or $D_{\mathcal{F}}(P, Q) > \delta$ in $\mathcal{O}(n + m(h \log h)T(G))$ time using $\mathcal{O}(|V| \log D/\log(1+\varepsilon) + S(G))$ space.*

Proof. Define $\mu \coloneqq 1 + \frac{9\varepsilon}{8+3\varepsilon} > 1$. In this case, $\log(\mu)^{-1} = \Theta(\log(1+\varepsilon)^{-1})$. In the preprocessing, we compute a μ-multiscale SPHS with value $\mathcal{O}(h \log h)$ in running time polynomial in $|V|$ and $1/\log(1+\varepsilon)$ by Theorem 1.4. We save this μ-multiscale SPHS as a matrix of booleans with a row for every vertex and a column for every SPHS. This takes $\mathcal{O}(|V| \log D/\log(1+\varepsilon))$ space and ensures that we can check in constant time whether a vertex is contained in a certain SPHS.

Now we choose $\alpha = \frac{\delta}{1-\frac{\mu}{4}}$ and i such that $\mu^i < \frac{\alpha}{2} \leq \mu^{i+1}$. From the μ-multiscale SPHS we compute the set $P^{\mu^{i+1}}$ in $\mathcal{O}(|P|)$ time. Then, we compute in $\mathcal{O}(|Q|(h \log h)T(G))$ time whether $D_{\mathcal{F}}\left(P^{\mu^{i+1}}, Q\right) \leq \alpha$ using Proposition 2.3. If this is true, using Lemma 2.1 and the triangle inequality, we can derive that

$$D_{\mathcal{F}}(P, Q) \leq \alpha + \frac{\mu^{i+1}}{2} \leq (1 + \frac{\mu}{4})\alpha = \frac{1+\frac{\mu}{4}}{1-\frac{\mu}{4}}\delta = \left(\frac{5}{3} + \varepsilon\right)\delta.$$

In the other case, we can use Lemma 2.2 to see that

$$D_{\mathcal{F}}(P, Q) > \alpha - \frac{\mu^{i+1}}{2} > (1 - \frac{\mu}{4})\alpha = \delta.$$ ◀

Since the Fréchet distance between P and Q can be at most D, we can apply the algorithm combined with a binary search on the value of the Fréchet distance to get the following result:

▶ **Corollary 2.5.** *Let $G = (V, E)$ be a graph with a metric weight function and highway dimension h and $\varepsilon > 0$. Suppose a distance query in G takes $\mathcal{O}(T(G))$ time using $\mathcal{O}(S(G))$ space. After preprocessing G in time polynomial in $|V|$ and $1/\log(1+\varepsilon)$, we can compute for any shortest path P with n vertices and any walk Q with m vertices a $(\frac{5}{3}+\varepsilon)$-approximation of $D_{\mathcal{F}}(P, Q)$ in $\mathcal{O}(\log D(n + m(h \log h)T(G)))$ time using $\mathcal{O}(|V| \log D/\log(1+\varepsilon) + S(G))$ space.*

Using the distance oracle from Theorem 1.5, our algorithm for approximating the Fréchet distance from Theorem 2.4 can be implemented in $\mathcal{O}\left(|P| \log D + |Q| (h \log h \log D)^2\right)$ time and using $\mathcal{O}(|V| \log D(1/\log(1+\varepsilon) + h \log h))$ space.

References

1 Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V. Goldberg, and Renato F. Werneck. VC-dimension and shortest path algorithms. In Luca Aceto, Monika Henzinger, and Jiří Sgall, editors, *Automata, Languages and Programming*, pages 690–699, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

2 Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V. Goldberg, and Renato F. Werneck. Highway dimension and provably efficient shortest path algorithms. *J. ACM*, 63(5), 12 2016. doi:10.1145/2985473.

3 Ittai Abraham, Amos Fiat, Andrew Goldberg, and Renato Werneck. Highway dimension, shortest paths, and provably efficient algorithms. pages 782–793, 01 2010. doi:10.1137/1.9781611973075.64.

4 Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014. doi:10.1137/130920526.

5 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 03 1995. doi:10.1142/S0218195995000064.

6 Helmut Alt, Christian Knauer, and Carola Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38:45–58, 10 2003. doi:10.1007/s00453-003-1042-5.

7 Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. pages 52–63, 09 2006. doi:10.1007/11841036_8.

8 Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. volume 2, pages 853–864, 01 2005.

9 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 04 2014. doi:10.1109/FOCS.2014.76.

10 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discrete Comput. Geom.*, 58(1):180–216, jul 2017. doi:10.1007/s00454-017-9878-7.

11 Kevin Buchin, Maike Buchin, Rolf van Leusden, Wouter Meulemans, and Wolfgang Mulzer. Computing the Fréchet distance with a retractable leash. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 241–252, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/s00454-016-9800-8.

12 Maike Buchin, Ivor van der Hoog, Tim Ophelders, Lena Schlipf, Rodrigo I. Silveira, and Frank Staals. Efficient Fréchet distance queries for segments, 2022. arXiv:2203.01794.

13 Anne Driemel and Sariel Har-Peled. Jaywalking your dog: Computing the fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013. arXiv:https://doi.org/10.1137/120865112, doi:10.1137/120865112.

14 Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. pages 365–374, 06 2010. doi:10.1145/1810959.1811019.

15 Anne Driemel, Ioannis Psarros, and Melanie Schmidt. Sublinear data structures for short Fréchet queries, 2019. arXiv:1907.04420.

16 Anne Driemel, Ivor van der Hoog, and Eva Rotenberg. On the discrete fréchet distance in a graph. In Xavier Goaoc and Michael Kerber, editors, *Proceedings of 38th International Symposium on Computational Geometry*, Leibniz International Proceedings in Informatics, LIPIcs. Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, June 2022. doi:10.4230/LIPIcs.SoCG.2022.36.

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

18 Roniel S. De Sousa, Azzedine Boukerche, and Antonio A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5), sep 2020. doi:10.1145/3406096.

19 Ivor van der Hoog, Eva Rotenberg, and Sampson Wong. Data structures for approximate discrete Fréchet distance, 2023. arXiv:2212.07124.