

APPROXIMATING THE (CONTINUOUS) FRÉCHET DISTANCE.

by

Connor Lee Colombe



APPROVED BY SUPERVISORY COMMITTEE:

Kyle Fox, Chair

Sergey Bereg

Benjamin Raichel

Copyright © 2020

Connor Lee Colombe

All rights reserved

APPROXIMATING THE (CONTINUOUS) FRÉCHET DISTANCE.

by

CONNOR LEE COLOMBE, BS

THESIS

Presented to the Faculty of
The University of Texas at Dallas
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN
COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

August 2020

ACKNOWLEDGMENTS

I would like to first say thank you to my advisor, Professor Kyle Fox, for his guidance, encouragement, and patience. You have been a great mentor to me and your passion for Computer Science is infectious. I greatly enjoyed my research experience and your geometry course. I hope to take the lessons and skills from my experience and use them to launch a successful research career.

I would also like to say thank you to my parents, Chris and Lisa, my siblings, Cameron and Erin, and my girlfriend, Morgan. The support and encouragement you all have provided me with has been instrumental to my educational success. I am very grateful to have people like you in my life.

I also would like to thank Karl Bringmann and Marvin Künnemann for some helpful discussions concerning turning an approximate decision procedure into a proper approximation algorithm.

July 2020

APPROXIMATING THE (CONTINUOUS) FRÉCHET DISTANCE.

Connor Lee Colombe, MSCS
The University of Texas at Dallas, 2020

Supervising Professor: Kyle Fox, Chair

We describe the first strongly subquadratic time algorithm with subexponential approximation ratio for approximately computing the Fréchet distance between two polygonal chains. Specifically, let P and Q be two polygonal chains with n vertices in d -dimensional Euclidean space, and let $\alpha \in [\sqrt{n}, n]$. Our algorithm deterministically finds an $O(\alpha)$ -approximate Fréchet correspondence in time $O((n^3/\alpha^2) \log n)$. In particular, we get an $O(n)$ -approximation in near-linear $O(n \log n)$ time, a vast improvement over the previously best known result, a linear time $2^{O(n)}$ -approximation. As part of our algorithm, we also describe how to turn any approximate decision procedure for the Fréchet distance into an approximate optimization algorithm whose approximation ratio is the same up to arbitrarily small constant factors. The transformation into an approximate optimization algorithm increases the running time of the decision procedure by only an $O(\log n)$ factor.

TABLE OF CONTENTS

| | |
|--|-----|
| ACKNOWLEDGMENTS | iv |
| ABSTRACT | v |
| LIST OF FIGURES | vii |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Our results | 5 |
| CHAPTER 2 PRELIMINARIES | 8 |
| CHAPTER 3 THE APPROXIMATE DECISION PROCEDURE | 11 |
| 3.1 Greedy mapping subroutines | 11 |
| 3.2 Remaining details | 18 |
| 3.3 Running time | 22 |
| CHAPTER 4 APPROXIMATION ALGORITHM | 24 |
| CHAPTER 5 CONCLUSION | 28 |
| REFERENCES | 29 |
| BIOGRAPHICAL SKETCH | 32 |
| CURRICULUM VITAE | |

LIST OF FIGURES

| | | |
|-----|---|----|
| 3.1 | Basic setup for $\text{GREEDYMAPPINGP}(i, t)$ | 13 |
| 3.2 | $\text{GREEDYMAPPINGP}(i, t)$: The case where $P[i^e - 1, i^e]$ is good | 13 |
| 3.3 | Left: Definition of $\langle t_1, \dots, t_\ell \rangle$. Right: Designating approximate reachability intervals near a pair (s^e, t_k) | 15 |
| 3.4 | A correspondence of cost δ between $P[i^e - 1, i^e]$ and $Q[t^c, t^f]$. A subset of matched points are represented by thin green line segments. | 18 |

CHAPTER 1

INTRODUCTION

The Fréchet distance is a commonly used method of measuring the similarity between a pair of curves. Both its standard (continuous) variant and its discrete variant for point sequences have seen use in map construction and mapping [1, 2], handwriting recognition [3], and protein alignment [4].

Formally, it is defined as follows: Let $P : [1, m] \rightarrow \mathbb{R}^d$ and $Q : [1, n] \rightarrow \mathbb{R}^d$ be two curves in d -dimensional Euclidean space. We'll assume P and Q are represented as ***polygonal chains***, meaning there exist ordered ***vertex*** sequences $\langle p_1, \dots, p_m \rangle$ and $\langle q_1, \dots, q_n \rangle$ such that $P(i) = p_i$ for all $1 \leq i \leq m$, $Q(j) = q_j$ for all $1 \leq j \leq n$, and both P and Q are linearly parameterized along line segments or ***edges*** between these positions. We define a ***re-parameterization*** $\sigma : [0, 1] \rightarrow [1, m]$ of P as any continuous, non-decreasing function such that $\sigma(0) = 1$ and $\sigma(1) = m$.¹ We define a re-parameterization $\theta : [0, 1] \rightarrow [1, n]$ of Q similarly. We define a ***Fréchet correspondence*** between P and Q as a pair (σ, θ) of re-parameterizations of P and Q respectively, and we say any pair of reals $(\sigma(r), \theta(r))$ for any $0 \leq r \leq 1$ are ***matched*** by the correspondence. Let $d(p, q)$ denote the Euclidean distance between points p and q in \mathbb{R}^d . The ***cost*** of the correspondence is defined as

$$\mu((\sigma, \theta)) := \max_{0 \leq r \leq 1} d(P(\sigma(r)), Q(\theta(r))).$$

Let Π_{FD} denote the set of all Fréchet correspondences between P and Q . The (***continuous***) ***Fréchet distance*** of P and Q is defined as

$$\text{FD}(P, Q) := \min_{(\sigma, \theta) \in \Pi_{\text{FD}}} \mu((\sigma, \theta)).$$

¹Re-parameterizations are normally required to be bijective, but we relax this requirement to simplify definitions and arguments throughout in the paper.

The standard intuition given for this definition is to imagine a person and their dog walking along P and Q , respectively, without backtracking. The person must keep the dog on a leash, and the goal is to pace their walks as to minimize the length of leash needed to keep them connected. There also exists a variant of the distance called the ***discrete Fréchet distance*** where the input consists of two finite ***point sequences***. Here, we replace the person and dog by two frogs. Starting with both frogs on the first point of their sequences, we must iteratively move the first, the second, or both frogs to the next point in their sequences. As before, the goal is to minimize the maximum distance between the frogs.

Throughout this paper, we'll assume $2 \leq m \leq n$. Computing the discrete Fréchet distance is easily done in $O(mn)$ time using dynamic programming. The first, and for a long time fastest, polynomial time algorithm for computing the continuous case was described by Alt and Godau [5]. They use parametric search [6, 7] and a quadratic time decision procedure (see Chapter 2) to compute the Fréchet distance in $O(mn \log n)$ time. Almost two decades passed before Agarwal *et al.* [8] improved the running time for the discrete case to $O(mn \log \log n / \log n)$. Buchin *et al.* [9] later improved the running time for the continuous case to $O(mn(\log \log n)^2)$ (these latter two results assume we are working in the word RAM model of computation). Recently, Gudmundsson *et al.* [10] described an $O(n \log n)$ time algorithm for special case of continuous Fréchet between chains P and Q assuming all edges have length a sufficiently large constant larger than $\text{FD}(P, Q)$. In short, they showed existence of a greedy process for moving the person and dog along their chains; the long edges prevent them from making mistakes during their walks.

From this brief history, one might believe that substantially faster algorithms are finally forthcoming for general cases of the continuous and discrete Fréchet distance, perhaps with polynomial improvements on the running times' dependency on m and n . Unfortunately, it seems unlikely that such meaningful improvements can ever be made; Bringmann [11]

showed that **strongly subquadratic** ($O(n^{2-\Omega(1)})$) time algorithms would violate the **Strong Exponential Time Hypothesis** (SETH) that solving CNF-SAT over n variables requires $\Omega(2^{(1-o(1))n})$ time [12].

Therefore, we are motivated to forego exactness and instead look for faster **approximation algorithms** for this problem. The first such result was a $(1 + \varepsilon)$ -approximation algorithm for the discrete Fréchet distance by Aronov *et al.* [13]. This algorithm runs in subquadratic and often near-linear time if at least one of P or Q fall into one of a few different “realistic” families of curves that include curves that are κ -**bounded** or ones modeling protein backbones. Driemel *et al.* [14] describe a $(1 + \varepsilon)$ -approximation for the standard continuous Fréchet distance that again runs more quickly if one of the curves belongs to a realistic family than it would otherwise. This latter algorithm was improved for some cases by Bringmann and Künnemann [15]. In the same work mentioned above, Gudmundsson *et al.* [10] described a \sqrt{d} -approximation algorithm that runs in linear time if the input polygonal chains have sufficiently long edges.

Approximation becomes much more difficult when there are no assumptions made about the input. Bringmann [11] showed there is no 1.001-approximation for the Fréchet distance, assuming SETH, although this result does not preclude the existence of constant-factor approximation algorithms with larger approximation ratio. For arbitrary point sequences, Bringmann and Mulzer [16] described an $O(\alpha)$ -approximation algorithm for any $\alpha \in [1, n/\log n]$ that runs in $O(n \log n + n^2/\alpha)$ time. Chan and Rahmati [17] later described an $O(n \log n + n^2/\alpha^2)$ time $O(\alpha)$ -approximation algorithm for any $\alpha \in [1, \sqrt{n/\log n}]$. However, for arbitrary polygonal chains, the only subquadratic time algorithm known with bounded approximation ratio is a linear time greedy procedure of Bringmann and Mulzer [16] that has an **exponential** worst case approximation ratio of $2^{\Theta(n)}$. We note that there is also a substantial body of work on the (approximate) nearest neighbor problem using Fréchet distance as the metric; see Mirzanezhad [18] for a survey of recent results.

These results assume the query curve or the curves being searched are short, so they do not appear directly useful in approximating the Fréchet distance between two curves of arbitrary length.

The closely related problems of computing the dynamic time warping distance and the geometric edit distance have a very similar history to that given above for the discrete Fréchet distance.² They both have straightforward quadratic time dynamic programming algorithms; these algorithms have been improved by (sub-)polylogarithmic factors for some low dimensional cases [19]; substantial improvements such as strongly subquadratic time algorithms violate SETH or other complexity theoretic assumptions [20, 21, 22, 23]; and there exist fast $(1+\varepsilon)$ -approximation algorithms specialized for realistic input sequences [24, 25]. Finally, there exist relatively efficient approximation algorithms for arbitrary point sequences as well. Kuszmaul [26] described $O((n^2/\alpha)\text{polylog } n)$ time $O(\alpha)$ -approximation algorithms for dynamic time warping distance over point sequences in *well separated tree metrics* of exponential spread and geometric edit distance over point sequences in arbitrary metrics. Fox and Li [27] described a randomized $O(n \log^2 n + (n^2/\alpha^2) \log n)$ time $O(\alpha)$ -approximation algorithm for geometric edit distance for points in low dimensional Euclidean space. Note that substantially better approximation algorithms exist for the traditional string (or Levenshtein) edit distance where all substitutions have cost exactly 1; see Andoni and Nosatzki [28] and its references.

Each of the problems of discrete Fréchet distance, dynamic time warping distance, and geometric edit distance admit strongly subquadratic approximation algorithms with polynomial approximation ratios for any input sequence in low dimensional Euclidean

²The dynamic time warping distance is defined similarly to the discrete Fréchet distance, except the goal is to minimize the *sum* of distances between the frogs over all pairs of points they stand upon. The geometric edit distance can be defined as the minimum number of point insertions and deletions plus the minimum total cost of point substitutions needed to transform one input sequence into another. The cost of a substitution is the distance between its points.

space. However, such a result remains conspicuously absent for the continuous Fréchet distance over arbitrary polygonal chains.

1.1 Our results

We describe the first strongly subquadratic time algorithm with subexponential approximation ratio for computing the Fréchet distance between two polygonal chains. Specifically, let P and Q be two polygonal chains of m and n vertices, respectively, in d -dimensional Euclidean space, and let $\alpha \in [\sqrt{n}, n]$. Our algorithm deterministically finds a Fréchet correspondence between P and Q of cost $O(\alpha) \cdot \text{FD}(P, Q)$ in time $O((n^3/\alpha^2) \log n)$. In particular, we get an $O(n)$ -approximation in near-linear $O(n \log n)$ time, a vast improvement over the linear time $2^{O(n)}$ -approximation of Bringmann and Mulzer [16]. Our algorithm combines ideas from the original exact algorithm of Alt and Godau [5] for continuous Fréchet distance with the high level strategy used by Chan and Rahmati [17] for approximating the discrete Fréchet distance. We also take inspiration from Gudmundsson *et al.*'s [10] greedy approach for computing the continuous Fréchet distance between polygonal chains with long edges.

Let $\delta > 0$. We describe an ***approximate decision procedure*** that either determines $\text{FD}(P, Q) > \delta$ or finds a Fréchet correspondence of cost $O(\alpha) \cdot \delta$. The ***exact*** decision procedure of Alt and Godau [5] computes a set of ***exact reachability intervals*** in the ***free space diagram*** of P and Q with respect to δ (see Chapter 2). Intuitively, these intervals represent all points on a single edge of Q that can be matched to a vertex of P (or vice versa) in a Fréchet correspondence of cost at most δ . For our approximate decision procedure, we compute a set of ***approximate reachability intervals*** such that the re-parameterizations realizing these intervals have cost $O(\alpha) \cdot \delta$. We cannot afford to compute intervals for all $\Theta(mn)$ vertex-edge pairs, so we instead focus on a set of $O(n^2/\alpha^2)$ vertex-edge pairs as described below that contain the first and last vertices and edges of

both chains. The approximate interval we compute for any vertex-edge pair contains the exact interval for that same pair. So if $\text{FD}(P, Q) \leq \delta$, we know (p_m, q_n) is approximately reachable and our desired Fréchet correspondence exists.

The vertex-edge pairs chosen to hold the approximate reachability intervals follow from the idea of good and bad points/vertices used by Chan and Rahmati [17]. Similar to their algorithm, we place a grid of side length $\alpha \cdot \delta$ so that at most $O(n/\alpha)$ vertices of P and Q lie within distance 3δ of the side of a grid box. We call these $O(n/\alpha)$ vertices **bad** and the rest **good**. Similarly, we call any edge with a bad endpoint bad. Our approximate reachability intervals are recorded only on vertex-edge pairs where the edge is bad and the vertex has at least one bad incident edge. To compute these intervals, we describe a method that essentially traces how a Fréchet correspondence of cost δ must behave starting from one of our approximate reachability intervals until it reaches some others we wish to compute. In short, when the correspondence leaves an approximate reachability interval, either the next edge of P or Q to leave the box is good and therefore long, or it is bad, and we can afford to compute some new approximate reachability intervals using this edge. We can easily compute correspondences between long edges and arbitrary length edges on the other curve, and we can greedily match the portions of the curves before they leave the box at cost at most $O(\alpha) \cdot \delta$. The traces take only $O(n)$ time each, and we perform at most $O(n^2/\alpha^2)$ traces, so our decision procedure takes $O(n^3/\alpha^2)$ time total.

The ultimate goal of our algorithm is to compute a Fréchet correspondence of cost $O(\alpha) \cdot \text{FD}(P, Q)$, preferably by doing black box reductions to our own $O(\alpha)$ -approximate decision procedure. Unfortunately, we are unaware of any known general method to do so.³ Therefore, we describe how to turn any approximate decision procedure into an algorithm

³Bringmann and Künnemann [15, Lemma 2.1] state there is a general method for turning an approximate decision procedure into an approximate optimization algorithm, at least when the approximation ratio of the decision procedure is at most 2. However, they rely on a method of Driemel *et al.* [14] that uses certain structural properties of the input polygonal chains that we cannot assume.

with the same approximation ratio up to arbitrarily small constant factors with only an $O(\log n)$ factor increase in running time. As a consequence, we see any improvement to our approximate decision procedure would lead to an immediate improvement in our overall approximation algorithm. Our method involves binary searching over a set of $O(n)$ values approximating distances between pairs of vertices. If there is a large gap between the Fréchet distance and the nearest of these $O(n)$ values, we can *simplify* both P and Q without losing much accuracy in the Fréchet distance computation while allowing for the long edge exact algorithm of Gudmundsson *et al.* [10] to succeed.

The rest of our paper is organized as follows. We describe preliminary notions in Chapter 2. We describe our decision procedure in Chapter 3. We describe how to turn it into an approximation algorithm in Chapter 4. Finally, we conclude with some closing thoughts in Chapter 5.

CHAPTER 2

PRELIMINARIES

Notes on notation Let $R : [1, n] \rightarrow \mathbb{R}^d$ be a polygonal chain in d -dimensional Euclidean space. We let $R[r, r']$ denote the restriction of R to $[r, r']$. In other words, the notation refers to the portion of R between points $R(r)$ and $R(r')$. We generally use s to refer to members of the domain of a polygonal chain P and t to refer to members of the domain of a polygonal chain Q . We use i and j , respectively, when these members are integers. We use superscript notation (s^a) to label particular members of these domains (and *not* to take the a th power of s), and we use subscript notation (s_k) when we are working with an ordered list of these members.

Free space diagram and reachability Let $P : [1, m] \rightarrow \mathbb{R}^d$ and $Q : [1, n] \rightarrow \mathbb{R}^d$ be two polygonal chains. Fix some $\delta > 0$. Alt and Godau [5] introduced the **free space diagram** as a useful tool for deciding if $\text{FD}(P, Q) \leq \delta$. It can be described by the set $F = \{(s, t) \in [1, m] \times [1, n]\}$. Each point $(s, t) \in F$ represents the pair of points $P(s)$ and $Q(t)$. We say point (s, t) is **free** if $\text{d}(P(s), Q(t)) \leq \delta$. The **free space** denoted $\mathcal{D}_{\leq \delta}(P, Q)$ consists of all free points between P and Q for a given δ . Formally, it is given by the set

$$\mathcal{D}_{\leq \delta}(P, Q) := \{(s, t) \in [1, m] \times [1, n] : \text{d}(P(s), Q(t)) \leq \delta\}.$$

We say that a point (s', t') in F is **reachable** if there exists an s and t -monotone path from $(1, 1)$ to (s', t') through the free space. We say a Fréchet correspondence (σ, θ) between P and Q **uses** or **passes through** a reachability interval if there exists some point $(\sigma(r), \theta(r))$ within that interval.

The standard decision procedure for determining if $\text{FD}(P, Q) \leq \delta$ works by dividing F into cells $C_{i,j} := [i-1, i] \times [j-1, j]$ for all $i \in \langle 2, \dots, m \rangle$ and $j \in \langle 2, \dots, n \rangle$. The intersection of a cell $C_{i,j}$ with the free space is convex [5]. The intersection of an edge of the free space

diagram cell $C_{i,j}$ with the free space forms a **free space interval**. The subset of reachable points within a free space interval form what is called an (exact) **reachability interval**.

Alt and Godau [5] showed that given the bottom and left reachability intervals of a free space diagram cell, we can compute the top and right reachability intervals of the same cell in $O(1)$ time. Their algorithm therefore loops through the cells in increasing order of i and j , computing reachability intervals one-by-one. Let $\alpha \in [\sqrt{n}, n]$. We cannot afford to compute all $\Theta(mn)$ reachability intervals, so instead we compute a collection of $O(n^2/\alpha^2)$ **α -approximate reachability intervals** (or approximate reachability intervals when α is clear from context). The approximate reachability intervals are subsets of the free space intervals such that for any point (s, t) on an approximate reachability interval, there exists a Fréchet correspondence between $P[1, s]$ and $Q[1, t]$ of cost $O(\alpha) \cdot \delta$. We often refer to exact or approximate reachability intervals by the subset of F they contain; for example, given $j - 1 \leq t^a \leq t^b \leq j$, we will use $\{i\} \times [t^a, t^b]$ to refer to an interval on the right side of cell $C_{i,j}$.

Grids, good points, bad points, and dangerous points Chan and Rahmati [17] utilize a d -dimensional grid to create the useful notion of good and bad vertices for their discrete Fréchet distance approximation algorithm. We adopt their use of a d -dimensional grid and create new constructs of good and bad to suit the continuous curves of our problem.

Let $P : [1, m] \rightarrow \mathbb{R}^d$ and $Q : [1, n] \rightarrow \mathbb{R}^d$ be two polygonal chains. Fix $\delta > 0$ and $\alpha \in [\sqrt{n}, n]$. Let G be a d -dimensional grid consisting of **boxes** of side length $\alpha \cdot \delta$. (We do not use the term **cell** here to avoid confusion with the free space diagram.) We say a vertex of P or Q is **good** if it is more than distance 3δ from any edge of G . If a vertex is not good, then we call it **bad**. For simplicity, we also designate $p_1, q_1, p_m,$ and q_n as bad, regardless of their position within boxes of G .

We also extend the constructs of good and bad to the edges of P and Q . We say an edge on either chain is **good** if both its endpoints are good vertices. Otherwise, the edge is **bad**. Lastly, we say that a vertex is **dangerous** (but not necessarily good or bad) if at least one of its incident edges is bad. Chan and Rahmati [17, Lemma 1] demonstrate how to compute a grid G with $O(n/\alpha)$ bad vertices in $O(n)$ time. Because each bad vertex has up to two incident edges, there are also $O(n/\alpha)$ bad edges. Each bad edge is incident to two vertices, so there are $O(n/\alpha)$ dangerous vertices as well. Our approximate decision procedure will compute approximate reachability intervals only between dangerous vertices and bad edges. Therefore, there will be at most $O(n^2/\alpha^2)$ such intervals.

Curve simplification Let $R : [1, n] \rightarrow \mathbb{R}^d$ be a polygonal chain with vertices $\langle r_1, \dots, r_n \rangle$. Our approximation algorithm relies on a method for simplifying chains so their edges are not too short. We use a slight modification of a curve simplification procedure of Driemel *et al.* [14]. Let $\nu > 0$ be a parameter. We mark r_1 and set it as the **current vertex**. We then repeat the following procedure until we no longer have a designated current vertex. We scan R from the current vertex until reaching the first vertex r_i of distance at least ν from the current vertex. We mark r_i , set it as the current vertex, and then go to the next iteration of the loop. The ν -**simplification** of R , denoted \hat{R} , is the polygonal chain consisting of exactly the marked vertices in order. Note that unlike Driemel *et al.* [14], we do not require the final vertex of R to be marked. We can easily verify that all edges of \hat{R} have length at least ν . Also, $\text{FD}(R, \hat{R}) \leq \nu$ [14, Lemma 2.3].

CHAPTER 3

THE APPROXIMATE DECISION PROCEDURE

In this chapter, we present our $O(\alpha)$ -approximate decision procedure. Let $P : [1, m] \rightarrow \mathbb{R}^d$ and $Q : [1, n] \rightarrow \mathbb{R}^d$ be two polygonal chains in d -dimensional Euclidean space as defined before, and let $\alpha \in [\sqrt{n}, n]$. Let $\delta > 0$. We begin by computing the grid G along with $O(n/\alpha)$ bad edges and points as defined in Chapter 2.

As previously discussed, our algorithm explicitly computes and records a set of $O(n^2/\alpha^2)$ *approximate reachability intervals* between dangerous vertices and bad edges. To compute these intervals, it occasionally performs a linear time greedy search for a good correspondence. We describe this greedy search procedure in Section 3.1 before giving the remaining details of the decision procedure and analyzing its approximation ratio in Section 3.2. Finally, we analyze its running time in Section 3.3.

3.1 Greedy mapping subroutines

We now describe a pair of subroutines used to greedily compute a Fréchet correspondence along two lengths of P and Q . The first of these procedures $\text{GREEDYMAPPINGP}(i, t)$ takes as its input an integer $i \in \langle 1, \dots, m \rangle$ such that p_i is a good vertex of P along with a real value $t \in [1, n]$ such that $\mathbf{d}(p_i, Q(t)) \leq \delta$. Informally, the procedure attempts to compute a Fréchet correspondence along a portion of P and Q of cost $O(\alpha) \cdot \delta$ that approximately follows what would happen with a correspondence of cost δ that maps i ‘close to’ t . We define another procedure $\text{GREEDYMAPPINGQ}(j, s)$ similarly, exchanging the roles of P and Q . As they are rather technical, the precise definitions of these procedures are best expressed in the following lemmas.

Lemma 3.1. *Let $i \in \langle 1, \dots, m \rangle$ and $t \in [1, n]$ such that p_i is good and $\mathbf{d}(p_i, Q(t)) \leq \delta$. The procedure $\text{GREEDYMAPPINGP}(i, t)$ described below outputs zero or more approximate*

reachability intervals between a bad edge of P or Q and a dangerous vertex on Q or P , respectively. For each pair $(s', t') \in [1, m] \times [1, n]$ lying on one of the approximate reachability intervals computed by the procedure, there exists a Fréchet correspondence of cost $O(\alpha) \cdot \delta$ between $P[i, s']$ and $Q[t, t']$. The procedure $\text{GREEDYMAPPINGQ}(j, s)$ has the same properties with the roles of P and Q exchanged.

Lemma 3.2. *Let $i \in \langle 1, \dots, m \rangle$ and $t \in [1, n]$ such that p_i is good and $\mathbf{d}(p_i, Q(t)) \leq \delta$. Suppose there exists a Fréchet correspondence (σ, θ) between P and Q of cost at most δ that matches i with some $t^* \geq t$ such that every point of $Q[t, t^*]$ is at most distance 3δ from p_i . Then, (σ, θ) passes through at least one approximate reachability interval output by the procedure $\text{GREEDYMAPPINGP}(i, t)$ described below. The procedure $\text{GREEDYMAPPINGQ}(j, s)$ has the same properties with the roles of P and Q exchanged.*

We now describe the procedure $\text{GREEDYMAPPINGP}(i, t)$ in detail before formally proving that it meets the guarantees given by the lemmas. The procedure works as follows. Observe p_i and $Q(t)$ lie in the same box B of grid G , because p_i is good. Let $s^e = m$ if P never leaves B after p_i . Otherwise, let s^e be the minimum value in $(i, m]$ such that $P(s^e)$ lies on the boundary of B (the ‘e’ stands for *exit*). Define t^e similarly for Q . See Figure 3.1. If either $s^e = m$ (resp. $t^e = n$), we check if all points of $Q[t, n]$ (resp. $P[i, m]$) lie in or within distance δ of B . If so, we output the trivial approximate reachability interval of $\{(m, n)\}$ and terminate the procedure. Otherwise, we output zero approximate reachability intervals. From here on, we assume neither $s^e = m$ nor $t^e = n$. Let $i^e \in \langle 1, \dots, m \rangle$ such that $i^e - 1 \leq s^e \leq i^e$, and define j^e similarly. We begin by considering cases where one or both curves leave box B through good edges.

Suppose the edge $P[i^e - 1, i^e]$ is good. In this case, let t^f be the minimum value in $(t, n]$ such that $\mathbf{d}(p_{i^e}, Q(t^f)) \leq \delta$, and let t^c be the *maximum* value in $[t, t^f)$ such that $\mathbf{d}(p_{i^e-1}, Q(t^c)) \leq \delta$ (the ‘f’ stands for *far*, and the ‘c’ stands for *close*). See Figure 3.2. If

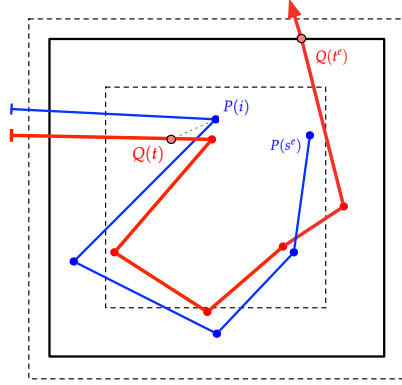


Figure 3.1. Basic setup for $\text{GREEDYMAPPINGP}(i, t)$

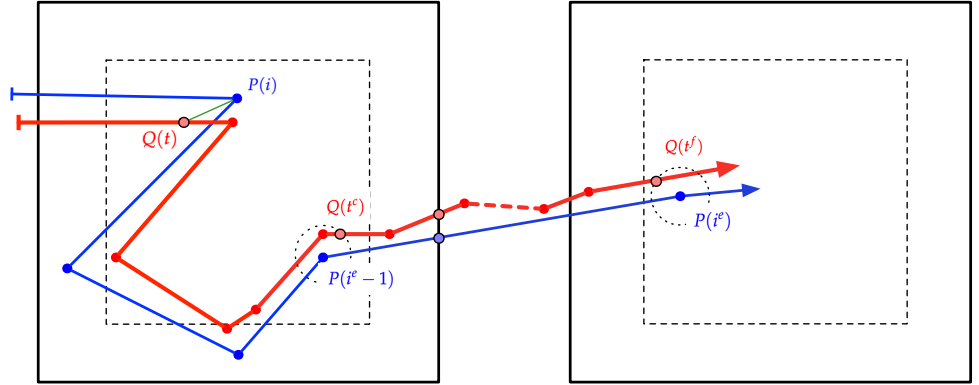


Figure 3.2. $\text{GREEDYMAPPINGP}(i, t)$: The case where $P[i^e - 1, i^e]$ is good

t^f or t^c is not well-defined, then we output zero reachability intervals. Otherwise, we check if every point of $Q[t, t^c]$ lies in or within distance δ of B and if $\text{FD}(P[i^e - 1, i^e], Q[t^c, t^f]) \leq \delta$. If so, we run $\text{GREEDYMAPPINGP}(i^e, t^f)$ and use its output. Otherwise, we output zero approximate reachability intervals.

Now suppose the previous case does not hold but edge $Q[j^e - 1, j^e]$ is good. Here, we perform similar steps to those described in the previous case, exchanging the roles of P and Q . Specifically, we let s^f be the minimum value in $(i, m]$ such that $d(q_{j^e}, P(s^f)) \leq \delta$, and let s^c be the maximum value in $[i, s^f)$ such that $d(q_{j^e - 1}, Q(s^c)) \leq \delta$. We check if every point of $P[i, s^c]$ lies in or within distance δ of B and if $\text{FD}(P[s^c, s^f], Q[j^e - 1, j^e]) \leq \delta$. If so, we run $\text{GREEDYMAPPINGQ}(j^e, s^f)$ and use its output. Otherwise, we output zero

approximate reachability intervals. From here on, we assume neither curve leaves box B through a good edge.

Let $t \leq t_1 < t_2 < \dots < t_\ell \leq t^e$ be the exhaustive list of first positions along their respective edges of Q such that $d(P(s^e), Q(t_k)) \leq \delta$ for each $k \in \langle 1, \dots, \ell \rangle$. See Figure 3.3, left. For each $k \in \langle 1, \dots, \ell \rangle$, we compute a pair of approximate reachability intervals as follows: Let $j_k \in \langle 1, \dots, n \rangle$ such that $j_k - 1 \leq t_k \leq j_k$. Let t_k^a be the minimum value in $[t_k, j_k]$ such that $d(p_{i^e}, Q(t_k^a)) \leq \delta$ and let t_k^b be the maximum value in $[t_k, j_k]$ such that $d(p_{i^e}, Q(t_k^b)) \leq \delta$. If t_k^a and t_k^b are well-defined, then we designate the interval $\{i^e\} \times [t_k^a, t_k^b]$ as approximately reachable. (If we have already designated a subset of $\{i^e\} \times [j_k - 1, j_k]$ as approximately reachable earlier in the decision procedure, then we extend the approximately reachable area by taking the union with what was computed previously. Every interval of $\{i^e\} \times [j_k - 1, j_k]$ we compute will end at (i^e, t_k^b) , so their union will itself be an interval.) Similarly, let s_k^a be the minimum value in $[s^e, i^e]$ such that $d(P(s_k^a), q_{j_k}) \leq \delta$ and let s_k^b be the maximum value in $[s^e, i^e]$ such that $d(P(s_k^b), q_{j_k}) \leq \delta$. If s_k^a and s_k^b are well-defined, then we designate the interval $[s_k^a, s_k^b] \times \{j_k\}$ as approximately reachable. See Figure 3.3, right. *In addition* to the above set of approximate reachability intervals, we also create some based on points of P between p_i and $P(s^e)$ that pass close to $Q(t^e)$. We use the same method as above, exchanging the roles of P and Q . We have concluded our description of $\text{GREEDYMAPPINGP}(i, t)$.

Procedure $\text{GREEDYMAPPINGQ}(j, s)$ has an analogous description, with the roles of P and Q exchanged. We are now ready to prove our lemmas.

Proof of Lemma 3.1. We use the same notation as given in the description of $\text{GREEDYMAPPINGP}(i, t)$. We first argue that we only output reachability intervals between bad edges and dangerous vertices. If we only output the trivial interval $\{(m, n)\}$ then the statement is trivially true. Otherwise, suppose we create an interval while

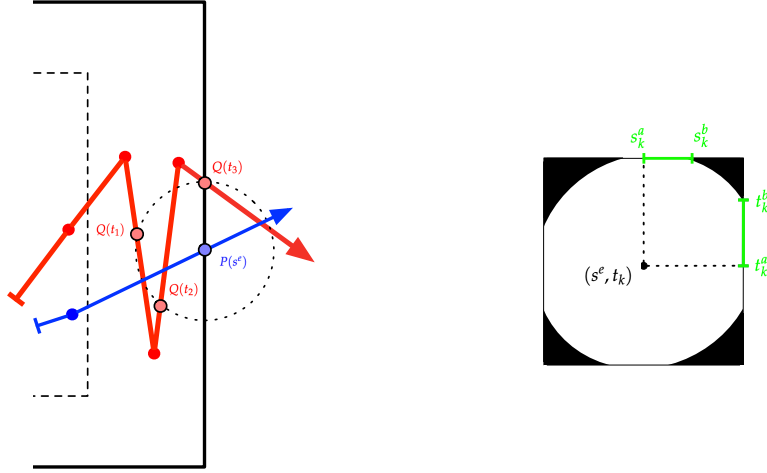


Figure 3.3. Left: Definition of $\langle t_1, \dots, t_\ell \rangle$. Right: Designating approximate reachability intervals near a pair (s^e, t_k) .

working with $P(s^e)$ and some nearby point $Q(t_k)$. We are not doing a recursive call to GREEDYMAPPINGP in this case, so $P[i^e - 1, i^e]$ is bad, and p_{i^e} is dangerous. Similarly, we are not doing a recursive call to GREEDYMAPPINGQ, so $Q[j^k - 1, j^k]$ is not a good edge with endpoint q_{j^k} outside of box B . Point $Q(t_k)$ is within distance δ of the boundary of B , so $Q[j^k - 1, j^k]$ cannot be a good edge with both endpoints in B , either. We conclude $Q[j^k - 1, j^k]$ is bad as well, and q_{j^k} is dangerous. A similar argument holds if we create an interval while working with $Q(t^e)$ and some nearby point of P .

We now argue that for any pair of points (s', t') on one of the approximate reachability intervals output by the procedure, there exists a correspondence of cost $O(\alpha) \cdot \delta$ between $P[i, s']$ and $Q[t, t']$. First, suppose GREEDYMAPPINGP(i, t) creates one or more approximate reachability intervals without doing a recursive call. Suppose $s^e = m$ or $t^e = n$, implying $(s', t') = (m, n)$. All points of $P[i, m]$ and $Q[t, n]$ lie in or within distance δ of B , so they are all distance at most $\sqrt{d}(\alpha + 1) \cdot \delta$ from each other and any Fréchet correspondence between $P[i, s']$ and $Q[t, t']$ has cost $O(\alpha) \cdot \delta$.

Now suppose otherwise, but (s', t') lies on an interval created while working with $P(s^e)$ and some nearby point $Q(t_k)$. All points of $P[i, s^e]$ and $Q[t, t_k]$ lie in B , so they are all

distance at most $\sqrt{d}\alpha \cdot \delta$ from each other and any Fréchet correspondence between $P[i, s^e]$ and $Q[t, t_k]$ has cost $O(\alpha) \cdot \delta$. The set of pairs $(x, y) \in P[i^e - 1, i^e] \times Q[j_k - 1, j_k]$ such that $d(P(x), Q(y)) \leq \delta$ includes (s^e, t_k) and (s', t') , and the set is convex [5], so we can extend our correspondence to include another between $P[s^e, s']$ and $Q[t_k, t']$ of cost at most δ . A similar argument covers the case where (s', t') lies on an interval created while working with $Q(t^e)$ and a nearby point of P .

Finally, suppose $\text{GREEDYMAPPINGP}(i, t)$ makes a recursive call to the $\text{GREEDYMAPPINGP}(i^e, t^f)$. Every point of $P[i, i^e - 1]$ and $Q[t, t^c]$ lies in or within distance δ of B , so every correspondence between $P[i, i^e - 1]$ and $Q[t, t^c]$ has cost at most $O(\alpha) \cdot \delta$. Also, we have $\text{FD}(P[i^e - 1, i^e], Q[t^c, t^f]) \leq \delta$. We can combine these correspondences with the one inductively guaranteed by the call to $\text{GREEDYMAPPINGP}(i^e, t^f)$ to get our desired correspondence between $P[i, s']$ and $Q[t, t']$. Again, a similar argument covers the case where we do a recursive call $\text{GREEDYMAPPINGQ}(j^e, s^f)$.

The proof for $\text{GREEDYMAPPINGQ}(j, s)$ is the same, but with the roles of P and Q exchanged. \square

Proof of Lemma 3.2. Again, we use the same notation as given in the description of $\text{GREEDYMAPPINGP}(i, t)$. By assumption and the fact that p_i is good, every point of $Q[t, t^*]$ lies within B . Let r^{se} be the smallest value such that $\sigma(r^{se}) \geq i$ and either $\sigma(r^{se}) = m$ or $\sigma(r^{se})$ lies on the boundary of B . Similarly, let r^{te} be the smallest value such that $\sigma(r^{te}) \geq i$ and either $\theta(r^{te}) = n$ or $\theta(r^{te})$ lies on the boundary of B . We see $\sigma(r^{se}) = s^e$ and $\theta(r^{te}) = t^e$.

Now, suppose $\text{GREEDYMAPPINGP}(i, t)$ does not do a recursive call. If we output the trivial interval $\{(m, n)\}$, then the lemma is trivially true. Suppose we do not output the trivial interval and $r^{se} \leq r^{te}$. Point $Q(\theta(r^{se}))$ lies on an edge $Q[j_k - 1, j_k]$ with one of the points $Q(t_k)$ where $d(P(s^e), Q(t_k)) \leq \delta$. By definition of t_k , we have $t_k \leq \theta(r^{se})$. The

set of $s^e \leq s' \leq i^e$ such that $\text{FD}(P[s^e, s'], Q[\theta(r^{se}), j_k]) \leq \delta$ is precisely the approximate reachability interval $[s_k^a, s_k^b] \times \{j_k\}$ we computed. Similarly, the set of $\theta(r^{se}) \leq t' \leq j_k$ such that $\text{FD}(P[s^e, i^e], Q[\theta(r^{se}), t']) \leq \delta$ is actually a *suffix* of the approximate reachability interval $\{i^e\} \times [t_k^a, t_k^b]$ we computed. A similar argument holds if $r^{te} < r^{se}$.

Lastly, suppose $\text{GREEDYMAPPINGP}(i, t)$ recursively calls $\text{GREEDYMAPPINGP}(i^e, t^f)$. Let t^{f*} be matched with i^e and t^{c*} be matched with $i^e - 1$ by (σ, θ) . Because p_{i^e-1} and p_{i^e} are both good, $\text{d}(p_{i^e-1}, Q(t^{c*})) \leq \delta$, and $\text{d}(p_{i^e}, Q(t^f)) \leq \delta$, points $Q(t^{c*})$ and $Q(t^f)$ lie within the same boxes as p_{i^e-1} and p_{i^e} , respectively. These boxes are distinct, so we may conclude $t^{c*} \leq t^f$, and therefore,

$$t^{c*} \leq t^c \leq t^f \leq t^{f*}.$$

Let $s^c \geq i^e - 1$ and $s^f \leq i^e$ be matched to t^c and t^f , respectively, by (σ, θ) .

Now, consider the following correspondence between $P[i^e - 1, i^e]$ and $Q[t^c, t^f]$. We match every point of $P[i^e - 1, s^c]$ to $Q(t^c)$, match $P[s^c, s^f]$ to $Q[t^c, t^f]$ exactly as done by (σ, θ) , and match every point of $P[s^f, i^e]$ to $Q(t^f)$. See Figure 3.4. We have $\text{d}(p_{i^e}, Q(t^c)) \leq \delta$ and $\text{d}(P(s^c), Q(t^c)) \leq \delta$, so the entire line segment $P[i^e - 1, s^c]$ lies within distance δ of $Q(t^c)$. Similarly, the line segment $P[s^f, i^e]$ lies within distance δ of $Q(t^f)$. Our correspondence has cost at most δ .

Now, consider any point $Q(t')$ with $t^{c*} \leq t' \leq t^c$ and let s' be matched to t' by (σ, θ) . We have $\text{d}(P(s'), Q(t')) \leq \delta$. We just argued that line segment $P[i^e - 1, s^c]$ is within distance δ of $Q(t^c)$, implying $\text{d}(P(s'), Q(t^c)) \leq \delta$. Finally, $\text{d}(p_{i^e-1}, Q(t^c)) \leq \delta$. By triangle inequality, $\text{d}(p_{i^e-1}, Q(t')) \leq 3\delta$, implying $Q(t')$ lies in B . As explained above, every point of $Q[t, t^*]$ lies in B . Every point of $Q[t^*, t^{c*}]$ lies within distance δ of a point in $P[i, i^e - 1]$ and therefore lies in or within distance δ of B . And, we just showed every point of $Q[t^{c*}, t^c]$ lies in B . Our algorithm will succeed at all its distance checks and recursively call $\text{GREEDYMAPPINGP}(i^e, t^f)$. Finally, a similar triangle inequality argument

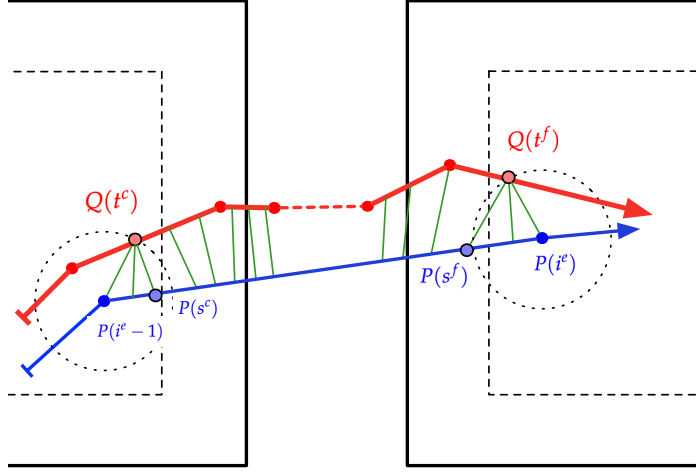


Figure 3.4. A correspondence of cost δ between $P[i^e - 1, i^e]$ and $Q[t^c, t^f]$. A subset of matched points are represented by thin green line segments.

implies every point of $Q[t^f, t^{*f}]$ is at most distance 3δ from p_{i^e} . We are inductively guaranteed that (σ, θ) passes through an approximate reachability interval output during the recursive call. Similar arguments apply if $\text{GREEDYMAPPINGP}(i, t)$ does a recursive call $\text{GREEDYMAPPINGQ}(j^e, s^f)$.

The proof for $\text{GREEDYMAPPINGQ}(j, s)$ is the same, but with the roles of P and Q exchanged. \square

3.2 Remaining details

We now fill in the remaining details of our approximate decision procedure. Recall, we have computed a grid G with boxes of side length $\alpha \cdot \delta$ such that there are $O(n/\alpha)$ bad vertices of P and Q . Also recall, $p_1, p_m, q_1,$ and q_n are designated as bad regardless of their position in G 's boxes.

We first check if $\mathbf{d}(p_1, q_1) \leq \delta$. If not, our procedure reports failure. Otherwise, let t^b and s^b be the maximum values in $[1, 2]$ such that $\mathbf{d}(p_1, Q(t^b)) \leq \delta$ and $\mathbf{d}(P(s^b), q_1) \leq \delta$, respectively. We designate intervals $\{1\} \times [1, t^b]$ and $[1, s^b] \times \{1\}$ as (approximately)

reachable. Now, for each $i \in \langle 2, \dots, m \rangle$ such that p_{i-1} is dangerous in order, for each $j \in \langle 2, \dots, n \rangle$ such that q_{j-1} is dangerous in order, we do the following.

Suppose we have designated an interval $\{i-1\} \times [t^a, t^b]$ as approximately reachable where $j-1 \leq t^a \leq t^b \leq j$. Suppose edge $P[i-1, i]$ is good. Then, we run the procedure $\text{GREEDYMAPPINGP}(i-1, t^a)$. If edge $P[i-1, i]$ is bad, we compute new approximate reachability intervals more directly as follows. First, let $t^{a'}$ be the minimum value in $[t^a, j]$ such that $\mathbf{d}(p_i, Q(t^{a'})) \leq \delta$, and let $t^{b'}$ be the maximum value in $[t^a, j]$ such that $\mathbf{d}(p_i, Q(t^{b'})) \leq \delta$. We designate interval $\{i\} \times [t^{a'}, t^{b'}]$ as approximately reachable (again, we may end up extending a previously compute approximately reachability interval on $\{i\} \times [j-1, j]$). Similarly, let $s^{a'}$ be the minimum value in $[i-1, i]$ such that $\mathbf{d}(P(s^{a'}), q_j) \leq \delta$, and let $s^{b'}$ be the maximum value in $[i-1, i]$ such that $\mathbf{d}(P(s^{b'}), q_j) \leq \delta$. We designate interval $[s^{a'}, s^{b'}] \times \{j\}$ as approximately reachable. We are done working with interval $\{i-1\} \times [t^a, t^b]$.

Now, suppose we have designated interval $[s^a, s^b] \times \{j-1\}$ as approximately reachable where $i-1 \leq s^a \leq s^b \leq i$. Suppose edge $Q[j-1, j]$ is good. If so, we run the procedure $\text{GREEDYMAPPINGQ}(j-1, s^a)$. If edge $Q[j-1, j]$ is bad, we compute new approximate reachability intervals more directly as follows. First, let $t^{a'}$ be the minimum value in $[j-1, j]$ such that $\mathbf{d}(p_i, Q(t^{a'})) \leq \delta$, and let $t^{b'}$ be the maximum value in $[j-1, j]$ such that $\mathbf{d}(p_i, Q(t^{b'})) \leq \delta$. We designate interval $\{i\} \times [t^{a'}, t^{b'}]$ as approximately reachable. Similarly, let $s^{a'}$ be the minimum value in $[s^a, i]$ such that $\mathbf{d}(P(s^{a'}), q_j) \leq \delta$, and let $s^{b'}$ be the maximum value in $[s^b, i]$ such that $\mathbf{d}(P(s^{b'}), q_j) \leq \delta$. We designate interval $[s^{a'}, s^{b'}] \times \{j\}$ as approximately reachable. We are done working with interval $[s^a, s^b] \times \{j-1\}$.

Once we have completed the iterations, we do one final step. We check if (m, n) lies on an approximate reachability interval. If so, we report there is a Fréchet correspondence between P and Q of cost $O(\alpha) \cdot \delta$. Otherwise, we report failure.

We remark that our decision procedure is easily extended to actually output the approximate correspondence itself instead of just determining if one exists by following the smaller correspondences we discover directly during the iterations or during runs of GREEDYMAPPINGP and GREEDYMAPPINGQ as we compute approximate reachability intervals.

Lemma 3.3. *The approximate decision procedure creates approximate reachability intervals only between bad edges of P or Q and dangerous vertices of Q or P , respectively.*

Proof. Vertices p_1 and q_1 are bad, so the intervals we compute before beginning the for loops are between bad edges and dangerous vertices. Now, consider working with some approximate reachability interval $\{i - 1\} \times [t^a, t^b]$ with $j - 1 \leq t^a \leq t^b \leq j$. Inductively, we may assume $Q[j - 1, j]$ is bad, implying q_j is dangerous. If $P[i - 1, i]$ is good, then Lemma 3.1 guarantees we only create approximate reachability intervals between bad edges and dangerous vertices. Otherwise, p_i is dangerous, and both approximate reachability intervals we directly create are for bad edge/dangerous vertex pairs. A similar argument applies when working with some interval $[s^a, s^b], \times \{j - 1\}$. \square

Lemma 3.4. *The decision procedure is correct if it reports a Fréchet correspondence of cost $O(\alpha) \cdot \delta$.*

Proof. Let (s', t') be any member of an approximate reachability interval created by the procedure. We will show there exists a Fréchet correspondence between $P[1, s']$ and $Q[1, t']$ of cost $O(\alpha) \cdot \delta$. If (s', t') lies on either interval created before the for loops begin, there is a trivial correspondence between $P[1, s']$ and $Q[1, t']$ of cost at most δ that only uses one point of either P or Q .

Now, consider working with some approximate reachability interval $\{i - 1\} \times [t^a, t^b]$ with $j - 1 \leq t^a \leq t^b \leq j$. Inductively, we may assume there is a correspondence of

cost $O(\alpha) \cdot \delta$ between $P[1, i - 1]$ and $Q[1, t^a]$. Suppose $P[i - 1, i]$ is good, and we call $\text{GREEDYMAPPINGP}(i - 1, t^a)$. By Lemma 3.1, we can extend our inductively guaranteed correspondence to one of cost $O(\alpha) \cdot \delta$ ending at any point (s', t') in any approximate reachability interval output by $\text{GREEDYMAPPINGP}(i - 1, t^a)$.

Finally, suppose instead that $P[i - 1, i]$ is bad. As in the proof of Lemma 3.1 or the original exact algorithm of Alt and Godau [5], there is a Fréchet correspondence of cost at most δ between $P[i - 1, s']$ and $Q[t^a, t']$ for any (s', t') on the approximate reachability intervals we directly compute. Again, we can extend the inductively guaranteed correspondence to end at any such (s', t') . A similar argument applies when working with some interval $[s^a, s^b] \times \{j - 1\}$. \square

Lemma 3.5. *Suppose there exists a Fréchet correspondence (σ, θ) between P and Q of cost at most δ . The decision procedure will report there exists a Fréchet correspondence of cost $O(\alpha) \cdot \delta$.*

Proof. Suppose (σ, θ) matches a pair $(i - 1, t^*)$ on some approximate reachability interval $\{i - 1\} \times [t^a, t^b]$. Suppose $P[i - 1, i]$ is good. Every point of $Q[t^a, t^*]$ lies within distance δ of p_{i-1} . Lemma 3.2 guarantees $\text{GREEDYMAPPINGP}(i - 1, t^a)$ will output at least one approximate reachability interval which includes a matched pair of (σ, θ) . We can easily verify that the interval must involve a later vertex of P than p_{i-1} .

Now, suppose instead that $P[i - 1, i]$ is bad. The set of $i - 1 \leq s' \leq i$ such that $\text{FD}(P[i - 1, s'], Q[t^*, j]) \leq \delta$ is precisely the approximate reachability interval $[s^a, s^b] \times \{j\}$ we computed. Similarly, the set of $t^* \leq t' \leq j$ such that $\text{FD}(P[i - 1, i], Q[t^*, t']) \leq \delta$ is actually a *subset* of the approximate reachability interval $\{i\} \times [t^a, t^b]$ we computed.

Either way, we have (σ, θ) using an interval for a later vertex of P or Q . If the interval contains (m, n) , the decision procedure will report there exists a cheap correspondence. Otherwise, we may assume it will report one inductively. Similar arguments apply if (σ, θ) includes a point on some approximate reachability interval $[s^a, s^b] \times \{j - 1\}$.

Finally, we observe that (σ, θ) does include a point on at least one approximate reachability interval, because our procedure begins by computing two intervals that include $(1, 1)$. \square

3.3 Running time

We are now ready to analyze the running time of our decision procedure. We start by analyzing the procedures `GREEDYMAPPINGP` and `GREEDYMAPPINGQ`.

Lemma 3.6. *Procedures `GREEDYMAPPINGP`(i, t) and `GREEDYMAPPINGQ`(j, s) can be implemented to run in at most $O(n)$ time.*

Proof. We use the notation given in the description of `GREEDYMAPPINGP`. Let $m' = m - i + 1$, and let n' be the number of vertices remaining in Q after $Q(t)$. If $s^e = m$ or $t^e = n$, then we spend $O(m' + n')$ time checking if vertices lies in or near box B . From here, assume neither $s^e = m$ nor $t^e = n$.

Suppose edge $P[i^e - 1, i^e]$ is good. Let $m'' = i^e - i \geq 1$, and let n'' be the number of vertices in $Q[t, t^f]$. We need to scan P and Q to find i^e , t^c , and t^f . We also need to check if every point of $Q[t, t^c]$ lies in or close to B . Doing these steps takes $O(m'' + n'')$ time. We need to check if $\text{FD}(P[i^e - 1, i^e], Q[t^c, t^f]) \leq \delta$. The portion of P in this check consists of a single line segment, so it can be done in $O(n'')$ time. Finally, we do a recursive call to `GREEDYMAPPINGP`(i^e, t^f) that inductively takes $O(n' + m' - n'' - m'')$ time. In total, we spend $O(n' + m')$ time. A similar argument holds if $P[i^e - 1, i^e]$ is bad but $Q[j^e - 1, j^e]$ is good.

Finally, suppose both edges are bad. We spend $O(n' + m')$ time total searching for s^e and t^e , finding points from the other curve that lie close to s^e and t^e , and computing approximate reachability intervals for each of these pairs of points. \square

Lemma 3.7. *The approximate decision procedure can be implemented to run in $O(n^3/\alpha^2)$ time.*

Proof. Finding the grid G with the set of $O(n/\alpha)$ bad vertices takes $O(n)$ time [17, Lemma 1]. There are at most twice as many bad edges as bad vertices, and at most twice as many dangerous vertices as bad edges, so there are $O(n/\alpha)$ dangerous vertices. Therefore, the decision procedure iterates over $O(n^2/\alpha^2)$ values of i and j . For each pair, we do at most two $O(n)$ time calls to `GREEDYMAPPINGP` or `GREEDYMAPPINGQ`, or we compute up to four approximate reachability intervals directly in constant time each. \square

Combining the above lemmas, we are now able to state the main result of this section.

Lemma 3.8. *Let $\alpha \in [\sqrt{n}, n]$. There exists an $O(\alpha)$ -approximate $O(n^3/\alpha^2)$ time decision procedure for the Fréchet distance between two polygonal chains in \mathbb{R}^d of at most n vertices each.*

CHAPTER 4

APPROXIMATION ALGORITHM

We now describe how to turn our approximate decision procedure into an approximation algorithm whose approximation ratio is arbitrarily close to that of the decision procedure. We emphasize that our techniques use the decision procedure as a black box subroutine, so any improvement to the running time of our approximate decision procedure will imply the same improvement to our approximation algorithm. In short, we use our approximate decision procedure to binary search over a set of $O(n)$ distances approximating the distances between vertices of P and Q . If the Fréchet distance lies in a large enough gap between a pair of these approximate distances, then we can simplify both polygonal chains so that their edge lengths become large compared to their Fréchet distance. We then run an exact Fréchet distance algorithm of Gudmundsson *et al.* [10] designed for this case.

Let $P : [1, m] \rightarrow \mathbb{R}^d$ and $Q : [1, n] \rightarrow \mathbb{R}^d$ be two polygonal chains in d -dimensional Euclidean space, and suppose we have an approximate decision procedure for the Fréchet distance between two polygonal chains with approximation ratio α . We assume α is at most a polynomial function of n (although it may be constant). Let $T(n, \alpha)$ denote the worst-case running time of the procedure on two polygonal chains of at most n vertices each. We assume $T(n, \alpha) = \Omega(n)$. Finally, consider any $0 < \varepsilon \leq 1$. We describe how to compute an $O((1 + \varepsilon)\alpha)$ -approximation of $\text{FD}(P, Q)$ in $O(T(n, \alpha) \log(n/\varepsilon))$ time.

We begin by doing a binary search over a set Z of $O(n)$ values close to all of the distances between pairs of vertices in P and Q . Let V denote the set of vertex points in P and Q . Our set Z is such that for any pair of distinct points $o_1, o_2 \in V$, there exist $x, x' \in Z$ such that $x \leq \text{d}(o_1, o_2) \leq x' \leq 2x$. Such a set can be computed in $O(n \log n)$ time [14, Lemma 3.9]. To perform the binary search, we simply search “down” if the approximate decision procedure finds an α -approximate correspondence, and we search

“up” if it does not. Let a and b be the largest value of Z for which the procedure fails and the smallest value for which it succeeds, respectively. If a does not exist, then we return the correspondence of cost $\alpha \cdot b$ found for b . We are guaranteed b exists, because the maximum distance between P and Q is achieved at a pair of vertices. From here on, we assume a exists.

We check if the approximate decision procedure finds a correspondence when given parameter $\delta := 12a/\varepsilon$. If so, let Z^a denote the sequence of distances

$$Z^a = \langle (1 + \varepsilon)^0 \cdot a, (1 + \varepsilon)^1 \cdot a, \dots, (1 + \varepsilon)^{\lceil 12/\varepsilon \rceil} \cdot a \rangle.$$

We binary search over Z^a and return the cheapest correspondence found.

Suppose no correspondence is found for $12a/\varepsilon$. We check if the approximate decision procedure finds a correspondence when given parameter $\delta := b/(2(1 + \varepsilon/2)(1 + \sqrt{d})\alpha)$. If *not*, let Z^b denote the sequence of distances

$$Z^b = \langle b/(1 + \varepsilon)^0, b/(1 + \varepsilon)^1, \dots, b/(1 + \varepsilon)^{\lceil 2(1 + \varepsilon/2)(1 + \sqrt{d})\alpha \rceil} \rangle.$$

We binary search over Z^b and return the cheapest correspondence found.

Finally, suppose no correspondence is found for $12a/\varepsilon$ but one is found for $b/(2(1 + \varepsilon/2)(1 + \sqrt{d})\alpha)$. We perform a $3a$ -simplification of P and Q , yielding the polygonal chains \hat{P} and \hat{Q} with at most n vertices each. Gudmundsson *et al.* [10] describe an $O(n \log n)$ time algorithm that computes the Fréchet distance of two polygonal chains *exactly* if all of their edges have length at least $(1 + \sqrt{d})$ times their Fréchet distance. As we prove in the next lemma, their algorithm will succeed in finding an optimal Fréchet correspondence between \hat{P} and \hat{Q} . This correspondence can be modified to create one for P and Q of cost at most $(1 + \varepsilon)\alpha \cdot \text{FD}(P, Q)$ (see Driemel *et al.* [14, Lemmas 2.3 and 3.5]).

Lemma 4.1. *The approximation algorithm finds a correspondence between P and Q of cost at most $(1 + \varepsilon)\alpha \cdot \text{FD}(P, Q)$.*

Proof. Suppose value a as defined in the procedure does not exist. We find a correspondence of cost at most $\alpha \cdot b \leq \alpha \cdot \mathbf{d}(p_1, q_1) \leq \alpha \cdot \text{FD}(P, Q)$. We assume from here on that a exists.

Suppose a binary search over Z^a or Z^b is performed. There exists values a' and $b' = (1 + \varepsilon)a'$ such that the approximate decision procedure fails with a' but succeeds at finding a correspondence of cost at most $\alpha \cdot b'$. We have $a' \leq \text{FD}(P, Q) \leq \alpha \cdot b' = (1 + \varepsilon)\alpha \cdot a'$.

Finally, suppose we perform neither binary search over Z^a or Z^b . In this case, we observe $12a/\varepsilon < \text{FD}(P, Q) \leq b/(2(1 + \varepsilon/2)(1 + \sqrt{d}))$. Every distance between a pair of vertices in P or Q is either at most $2a < (\varepsilon/6)\text{FD}(P, Q)$ or at least $b/2 \geq (1 + \sqrt{d})(1 + \varepsilon/2)\text{FD}(P, Q)$. We observe $\text{FD}(\hat{P}, \hat{Q}) \leq \text{FD}(P, Q) + 6a < (1 + \varepsilon/2)\text{FD}(P, Q)$ [14, Lemma 2.3]. Polygonal chains \hat{P} and \hat{Q} have no edges of length at most $2a$, implying all edges have length at least $(1 + \sqrt{d})(1 + \varepsilon/2)\text{FD}(P, Q) > (1 + \sqrt{d})\text{FD}(\hat{P}, \hat{Q})$. The conditions for the algorithm of Gudmundsson *et al.* [10] are met, and as explained earlier, their algorithm will lead to the desired correspondence between P and Q . \square

Lemma 4.2. *The approximation algorithm can be implemented to run in $O(T(n, \alpha) \log(n/\varepsilon))$ time.*

Proof. We spend $O(n \log n)$ time computing Z . We do $O(\log n)$ calls to the approximate decision procedure binary searching over Z . Sequences Z^a and Z^b contain $O(\log_{1+\varepsilon}(1/\varepsilon)) = O((1/\varepsilon) \log(1/\varepsilon))$ and $O(\log_{1+\varepsilon} \alpha) = O((1/\varepsilon) \log n)$ values, respectively. Therefore, binary searching over Z^a or Z^b requires $O(\log((1/\varepsilon) \log(n/\varepsilon))) = O(\log(n/\varepsilon))$ calls to the approximate decision procedure. The case where we have to simplify the polygonal chains and run the algorithm of Gudmundsson *et al.* [10] requires only $O(n \log n)$ additional time. The lemma follows. \square

We may now state the main result of this section.

Theorem 4.3. *Suppose we have an α -approximate decision procedure for Fréchet distance that runs in time $T(n, \alpha)$ on two polygonal chains in \mathbb{R}^d of at most n vertices each. Let $0 < \varepsilon \leq 1$. Given two such chains P and Q , we can find a Fréchet correspondence between P and Q of cost at most $(1 + \varepsilon)\alpha \cdot \text{FD}(P, Q)$ in $O(T(n, \alpha) \log(n/\varepsilon))$ time.*

Finally, we combine Theorem 4.3 with Lemma 3.8 while setting $\varepsilon := 1$ to get the main result of this paper.

Corollary 4.4. *Let P and Q be two polygonal chains in \mathbb{R}^d of at most n vertices each, and let $\alpha \in [\sqrt{n}, n]$. We can compute a Fréchet correspondence between P and Q of cost at most $O(\alpha) \cdot \text{FD}(P, Q)$ in $O((n^3/\alpha^2) \log n)$ time.*

CHAPTER 5
CONCLUSION

We described the first strongly subquadratic time approximation algorithm for the continuous Fréchet distance that has a subexponential approximation guarantee. Specifically, it computes an $O(\alpha)$ -approximate Fréchet correspondence in $O((n^3/\alpha^2) \log n)$ time for any $\alpha \in [\sqrt{n}, n]$. We admit that our result is not likely the best running time one can achieve and that it serves more as a first major step toward stronger results. In particular, it would be interesting to know if our running time analysis is even tight; perhaps a more involved analysis applied to a slight modification of our decision procedure could lead to a better running time. We leave open further improvements such as the one described above.

REFERENCES

- [1] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk, *Map Construction Algorithms*. Springer, 2015.
- [2] D. Chen, A. Driemel, L. J. Guibas, A. Nguyen, and C. Wenk, “Approximate map matching with respect to the Fréchet distance,” in *Proc. 13th Meeting on Algorithm Engineering and Experiments*, 2011, pp. 75–83.
- [3] E. Sriraghavendra, K. Karthik, and C. Bhattacharyya, “Fréchet distance based approach for searching online handwritten documents,” in *Proc. 9th Intern. Conf. Document Analysis and Recognition*, 2007, pp. 461–465.
- [4] M. Jiang, Y. Xu, and B. Zhu, “Protein structure-structure alignment with discrete Fréchet distance,” *J. Bioinformatics and Computational Biology*, vol. 6, no. 1, pp. 51–64, 2008.
- [5] H. Alt and M. Godau, “Computing the Fréchet distance between two polygonal curves,” *Int. J. Comput. Geom. Appl.*, vol. 5, pp. 75–91, 1995.
- [6] R. Cole, “Slowing down sorting networks to obtain faster sorting algorithms,” *J. ACM*, vol. 34, no. 1, pp. 200–208, 1987.
- [7] N. Megiddo, “Applying parallel computation algorithms in the design of serial algorithms,” *J. Assoc. Comput. Mach.*, vol. 30, no. 4, pp. 852–865, 1983.
- [8] P. K. Agarwal, R. B. Avraham, H. Kaplan, and M. Sharir, “Computing the discrete Fréchet distance in subquadratic time,” *SIAM J. Comput.*, vol. 43, no. 2, pp. 429–449, 2014.
- [9] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer, “Four Soviets walk the dog: Improved bounds for computing the Fréchet distance,” *Discrete Comput. Geom.*, vol. 58, no. 1, pp. 180–216, 2017.
- [10] J. Gudmundsson, M. Mirzanezhad, A. Mohades, and C. Wenk, “Fast Fréchet distance between curves with long edges,” *Int. J. Comput. Geom. Appl.*, vol. 29, no. 2, pp. 161–187, 2019.
- [11] K. Bringmann, “Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails,” in *Proc. 55th. Ann. IEEE Sympos. on Found. of Comp. Sci.*, 2014, pp. 661–670.
- [12] R. Impagliazzo and R. Paturi, “On the complexity of k-SAT,” *J. Comp. Sys. Sci.*, vol. 62, no. 2, pp. 367–375, 2001.

- [13] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk, “Fréchet distance for curves, revisited,” in *Proc. 14th Ann. Euro. Sympos. Algo.*, 2006, pp. 52–63.
- [14] A. Driemel, S. Har-Peled, and C. Wenk, “Approximating the Fréchet distance for realistic curves in near linear time,” *Discrete Comput. Geom.*, vol. 48, no. 1, pp. 94–127, 2012.
- [15] K. Bringmann and M. Künnemann, “Improved approximation for Fréchet distance on c -packed curves matching conditional lower bounds,” *Int. J. Comput. Geom. Appl.*, vol. 27, no. 1-2, pp. 85–120, 2017.
- [16] K. Bringmann and W. Mulzer, “Approximability of the discrete Fréchet distance,” *J. Comput. Geom.*, vol. 7, no. 2, pp. 46–76, 2016.
- [17] T. M. Chan and Z. Rahmati, “An improved approximation algorithm for the discrete Fréchet distance,” *Inf. Process. Lett.*, vol. 138, pp. 72–74, 2018.
- [18] M. Mirzanezhad, “On the approximate nearest neighbor queries among curves under the fréchet distance,” *CoRR*, vol. abs/2004.08444, 2020. [Online]. Available: <https://arxiv.org/abs/2004.08444>
- [19] O. Gold and M. Sharir, “Dynamic time warping and geometric edit distance: Breaking the quadratic barrier,” *ACM Trans. Algorithms*, vol. 14, no. 4, pp. 50:1–50:17, 2018.
- [20] A. Backurs and P. Indyk, “Edit distance cannot be computed in strongly subquadratic time (unless SETH is false),” *SIAM J. Comput.*, vol. 47, no. 3, pp. 1087–1097, 2018.
- [21] A. Abboud, A. Backurs, and V. V. Williams, “Tight hardness results for LCS and other sequence similarity measures,” in *Proc. 56th Ann. IEEE Sympos. on Found. Comp. Sci.*, 2015, pp. 59–78.
- [22] K. Bringmann and M. Künnemann, “Quadratic conditional lower bounds for string problems and dynamic time warping,” in *Proc. 56th Ann. IEEE Sympos. on Found. of Comp. Sci.*, 2015, pp. 79–97.
- [23] A. Abboud, T. D. Hansen, V. V. Williams, and R. Williams, “Simulating branching programs with edit distance and friends: Or: a polylog shaved is a lower bound made,” in *Proc. 48th Ann. ACM Sympos. Theory of Comput.*, 2016, pp. 375–388.
- [24] P. K. Agarwal, K. Fox, J. Pan, and R. Ying, “Approximating dynamic time warping and edit distance for a pair of point sequences,” in *Proc. 32nd Int. Conf. Comput. Geom.*, 2016, pp. 6:1–6:16.
- [25] R. Ying, J. Pan, K. Fox, and P. K. Agarwal, “A simple efficient approximation algorithm for dynamic time warping,” in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geo. Inf. Sys.*, 2016.

- [26] W. Kuszmaul, “Dynamic time warping in strongly subquadratic time: Algorithms for the low-distance regime and approximate evaluation,” in *Proc. 46th Intern. Colloqu. Automata, Languages, Programming*, 2019, pp. 80:1–80:15.
- [27] K. Fox and X. Li, “Approximating the geometric edit distance,” in *Proc. 30th Int. Symp. Algo. Comput.*, 2019, pp. 26:1–26:16.
- [28] A. Andoni and N. S. Nosatzki, “Edit distance in near-linear time: it’s a constant factor,” *CoRR*, vol. abs/2005.07678, in FOCS 2020, to appear.

BIOGRAPHICAL SKETCH

Connor Colombe was born in Plano, Texas in June of 1995. After graduating from Plano West Senior High in 2014, he attended Harvey Mudd College in Claremont, California to pursue a joint degree in Physics and Mathematics. In the Summer months of 2016 and 2017, he worked as an intern at NASA's Jet Propulsion Laboratory. Upon graduating from Harvey Mudd in 2018, he pursued a master's degree in Computer Science at The University of Texas at Dallas. Starting in Fall 2020, he will be enrolled at the University of Texas at Austin to pursue a PhD in Operations Research. When he is not working nor engaged in recreational mathematics, he can be found at the local rock climbing gym.

CURRICULUM VITAE

Connor Lee Colombe

July 20, 2020

Contact Information:

Department of Computer Science
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: connorcolombe@utdallas.edu

Educational History:

BS, Physics and Mathematics Joint Major, Harvey Mudd College, 2014-2018

MS, Computer Science, University of Texas at Dallas, 2018-2020

Chitosan Nanoparticles and Their Use in The Brain Patch Project

Senior Undergraduate Thesis

Department of Physics, Harvey Mudd College

Advisor: Dr. Tom Donnelly

Employment History:

Summer Intern, NASA Jet Propulsion Laboratory, May 2016 – July 2016

Summer Intern, NASA Jet Propulsion Laboratory, May 2017 – July 2017