AN ABSTRACT OF THE THESIS OF

Hanzhong Xu for the degree of Doctor of Philosophy in Computer Science presented on January 28, 2020.

Title: Computing the Fréchet Distance Between Surfaces

Abstract approved: ______________________________

Amir Nayyeri

The Fréchet distance is a measure of similarity between curves or surfaces. The Fréchet distance between two polygons can be computed in polynomial time, but it is much harder to compute the Fréchet distance between surfaces. We present the first $(1+\varepsilon)$-approximation algorithm and the first exact algorithm for computing the Fréchet distance between two surfaces. Next, we show that computing the Fréchet distance between a surface and a triangle is in PSPACE. Combining the approximation algorithm and the exact algorithm, we present an improved version of $(1+\varepsilon)$-approximation algorithm. Finally, we present a new restricted class of surface, surfaces composed of large triangles, for which the Fréchet distance between them can be computed faster.

Computing the Fréchet Distance Between Surfaces

by

Hanzhong Xu

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented January 28, 2020
Commencement June 2020

Doctor of Philosophy thesis of Hanzhong Xu presented on January 28, 2020.

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Hanzhong Xu, Author

ACKNOWLEDGEMENTS

I would like to thank my advisor, Amir Nayyeri, for his guidance through each stage of the process.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Chapter 1: Introduction

1.1 Objective

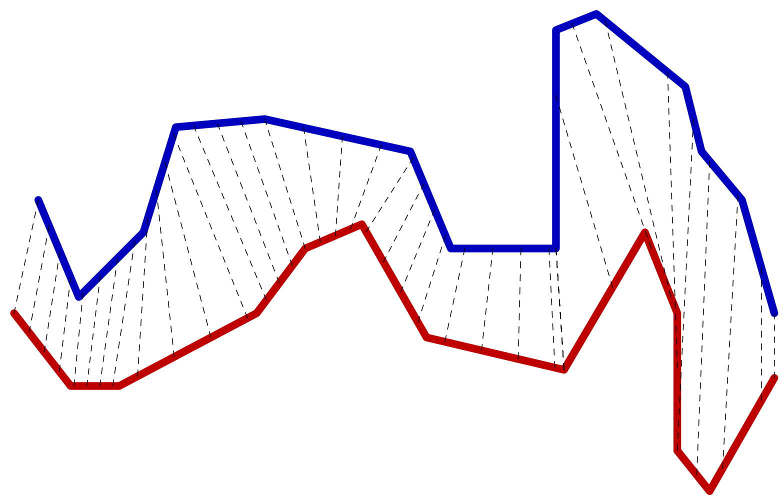
The purpose of this study is to: (1) explore algorithms to compute or approximate the Fréchet distance between surfaces; (2) explore restricted classes of surfaces for which we can compute or approximate the Fréchet distance quickly. In this thesis, I present exact and approximation algorithms and an exact algorithm for computing the Fréchet distance between surfaces, and show that computing the Fréchet distance between a triangle and a surface is in PSPACE.

1.2 Background

Shape matching is a central problem in many applications, such as object recognition in image processing and function detection in protein modeling. Given two geometric objects, we are interested in measuring the similarity between them. The Fréchet distance is a natural way to measure similarity for geometric objects such as curves and surfaces. The Fréchet distance and its variants have been used in many applications, such as matching of time-series in databases, speech and handwriting recognition [20, 21, 23].

The Fréchet distance between two curves P and Q denoted $\delta_F(P, Q)$, can be defined as follows. Imagine a man walks along the first curve P, his dog walks along the second curve Q, and they are connected by a leash. They can vary their speed, but they are not allowed to backtrack. The Fréchet distance between P and Q is the minimal length of a leash required for them to traverse the curves.

Compared to other similarity measures between spaces, the Fréchet distance is more natural when it is important to capture the topology of the underlying spaces. For a comparison, consider the Hausdorff distance between two point sets P and Q. Mapping each point of P to its closest point in Q, and similarly each point of Q to its closest point in P, the Hausdorff distance is the maximum distance between a point and its image. We can use the Hausdorff distance to measure the distance between curves if we treat



curves as sets of points, however, with this similarity measure we are disregarding the topology. The Fréchet distance, on the other hand, takes the topology into account.

The Fréchet distance between curves can be computed or approximated efficiently. Let n be the total number of vertices of two polygonal curves. The Fréchet distance can be computed between such polygonal curves in $O(n^2 \log n)$ time [3]. Moreover the Fréchet distance can be approximated in nearly linear time for more restricted classes of curves. For example, there is a $(1+\varepsilon)$-approximation algorithm for c-packed curves that runs in $O((cn/\epsilon) \log n)$ time [14]. Computing the Fréchet distance between surfaces is much harder. It is NP-complete to decide the Fréchet distance even between a triangle and a surface [16], or between two real valued surfaces [10]. On the positive side, Alt and Buchin [2] show that the Fréchet distance between triangulated surfaces is uppersemicomputable: there is an algorithm that outputs a sequence of values converging to the Fréchet distance. For more restricted classes, such as polygons [8], folded polygons [12] and polygonal spaces with a constant number of holes [22], there are polynomial time algorithms. Buchin et al. [9] studied the Fréchet distance between moving curves. They show that some variants are polynomial-time solvable or NP-complete depending on the restrictions imposed on how the moving curves are matched.

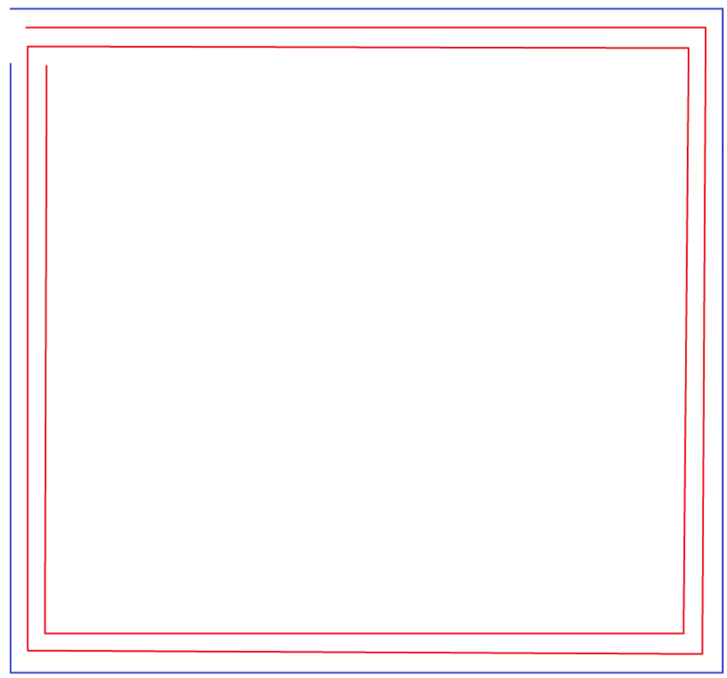


Figure 1.1: These two curves are close under Hausdorff distance, but far from each other under Fréchet distance.

1.3 Our Results

In this thesis, we describe exact and approximation algorithms. We show two different $(1+\varepsilon)$-approximation algorithms, a modification of the other with improvement in the running time. The first algorithm is simpler, but with weaker bounds on the running time. We describe both in the thesis to make the exposition smoother.

Theorem 1.3.1. *Let $\mathcal{R}$ and $\mathcal{S}$ be two triangulated surfaces with m and n vertices, and let $\varepsilon > 0$. There exists a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance $\delta_F(\mathcal{R}, \mathcal{S})$ between $\mathcal{R}$ and $\mathcal{S}$ with running time*

$$2^{O\left(\left(m+n+\frac{Area(\mathcal{R})+Area(\mathcal{S})}{(\varepsilon\delta)^2}\right)^2\right)}.$$

Theorem 1.3.2. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces, and $\delta = \delta_F(\mathcal{R}, \mathcal{S})$. There is* *a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ with respect to the ℓ_1 norm in*

$$\log(\delta + 1/\delta)(1/\epsilon)^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)}$$

time, where $\mathcal{R}_V$ and $\mathcal{S}_V$ are the set of vertices of $\mathcal{R}$ and $\mathcal{S}$.

Moreover, we describe exact decision algorithms for computing the Fréchet distance between two triangulated surfaces.

Theorem 1.3.3. *Let $\mathcal{R}$ and $\mathcal{S}$ be triangulated surfaces with m and n vertices, and let $\delta \geq 0$. There is an algorithm to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$.*

For a triangle and a triangulated surfaces, we show that their Fréchet distance can be decided in PSPACE. It was shown that this special case is NP-hard [16].

Theorem 1.3.4. *Let $\mathcal{R}$ be a triangulated surface with m vertices, $\mathcal{S}$ be a triangle, and let $\delta \geq 0$. There is an algorithm to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$ in PSPACE.*

Finally, we study the problem for restricted classes of surfaces. For terrains our Theorem 7.5.2 already implies a $O((D+1)/(\varepsilon\delta)^2) \cdot n + 2^{O((D+1)^4/(\varepsilon^4\delta^2))}$ running time approximation algorithm where D is the maximum slope of the two terrains. We also study the Fréchet distance between surfaces that are composed of large triangles relative to their Fréchet distance. We show an exact algorithm for computing the Fréchet distance between two surfaces composed of large triangles.

Theorem 1.3.5. *Let $\mathcal{R}$ and $\mathcal{S}$ be two surfaces composed of 2δ-large triangles, such that $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$. There is an exact algorithm for computing $\delta_F(\mathcal{R}, \mathcal{S})$ in*

$$2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)^2}$$

time.

Chapter 2: Related Work

In this chapter, we review existing results related to Fréchet distance between curves and surfaces. For curves, we mainly focus on a few results that are more directly related to the contribution of this thesis, and refer the interested reader to [14] [18] for more exhaustive literature review. For surfaces, we try to include all the related results.

2.1 Fréchet Distance between Curves

Computing the Fréchet distance between curves has been well studied. Alt et al. [3] give the first exact algorithm for computing the Fréchet distance between two polygonal curves in $O(nm \log(nm))$ time, where n and m are the number of vertices of two curves. For general curves, it is still an open problem to compute or approximate the Fréchet distance between curves with subquadratic running time. However, recently Agarwal et al. [1] found a subquadratic time algorithm for the discrete Fréchet distance. Many researchers studied the Fréchet distance for restricted families of curves with the goal of finding faster (nearly linear time) algorithms. For example, see the $(1+\varepsilon)$-approximation algorithm of Aronov et al. [4] for k-bounded curves and backbone curves.

2.1.1 Exact Algorithms for Curves

General curves. Alt and Godau [3] describe a decision algorithm to check whether the Fréchet distance between two polygonal curves is no more than a given δ in $O(nm \log(nm))$ time. The idea is to build the so called free space diagram for the given δ. Roughly speaking, the free space diagram is a subset of an $n \times m$ axis parallel rectangle. There is a monotonic path from the bottom left corner to the top right corner within the free space diagram, if and only if the Fréchet distance is at most δ. More specifically, let $P : [0, n-1] \to \mathbb{R}^d$ and $Q : [0, m-1] \to \mathbb{R}^d$ be two polygonal curves with n and m vertices. The free space diagram for P, Q and δ is defined as $F_\delta = \{(s,t) \in$

$[0, n-1] \times [0, m-1] \mid dist(P(s), Q(t)) \leq \delta\}$. The points in the free space diagram correspond to the points in $P \times Q$ with distance at most δ. A monotone path from $(0,0)$ to $(n-1, m-1)$ within the free space diagram corresponds to a homeomorphism with Fréchet length at most δ.

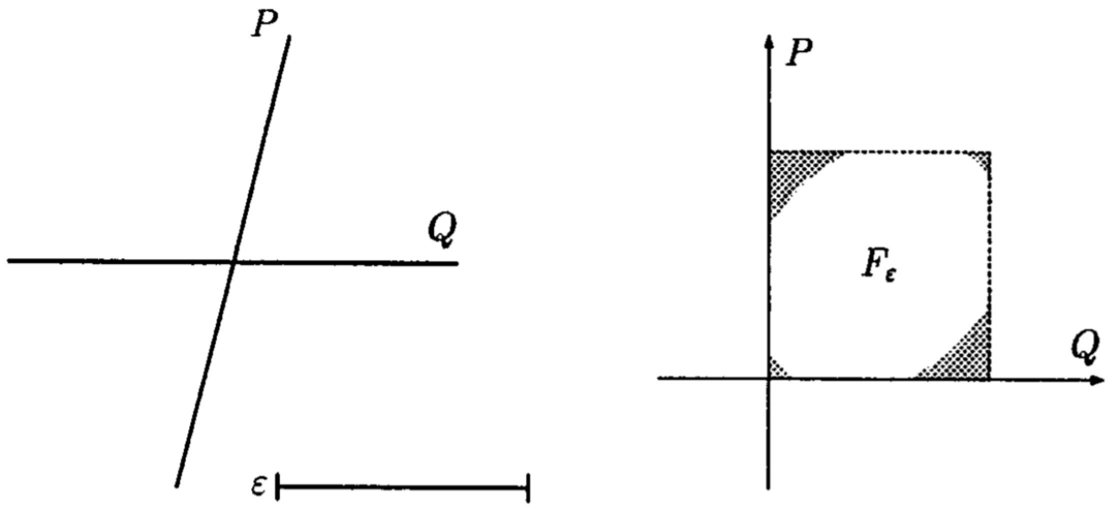


Figure 2.1: Free space diagram F_δ for two segments P and Q [3].

Alt and Godau show several interesting properties of the free space diagram. They show that the free space diagram is composed of $n \times m$ cells, where each cell is the intersection of a square with a perhaps degenerate ellipse. Each cell is basically the free space diagram between two segments, see Figure 2.1. Based on this observation, they design an algorithm to check whether a path from $(0,0)$ to $(n-1, m-1)$ exists in F_δ, or equivalently whether the Fréchet distance between P and Q is at most δ. This algorithm is in spirit similar to a dynamic programming algorithm that works for the discrete case.

Having an algorithm that can verify whether the Fréchet distance between two curves is at most δ, Alt and Godau use a binary search together with this algorithm to look for the value of the Fréchet distance. To obtain an exact algorithm they would still need a finite set of candidate values. In fact, they show a set of polynomial size for the candidates, which they call critical values. Critical values are composed of three categories: (a) the distances between every pair of vertices from different curves, (b) the distances between a vertex of one curve and a segment of the other curve, and (c) the value determined by two vertices v and u of one curve, and a point p on a segment of the other curve that minimizes $\max(dist(p, v), dist(p, u))$. See Figure 2.2 for an illustration. One of the open problems that we propose in this thesis is whether it is possible to obtain a finite set of critical values for two surfaces.

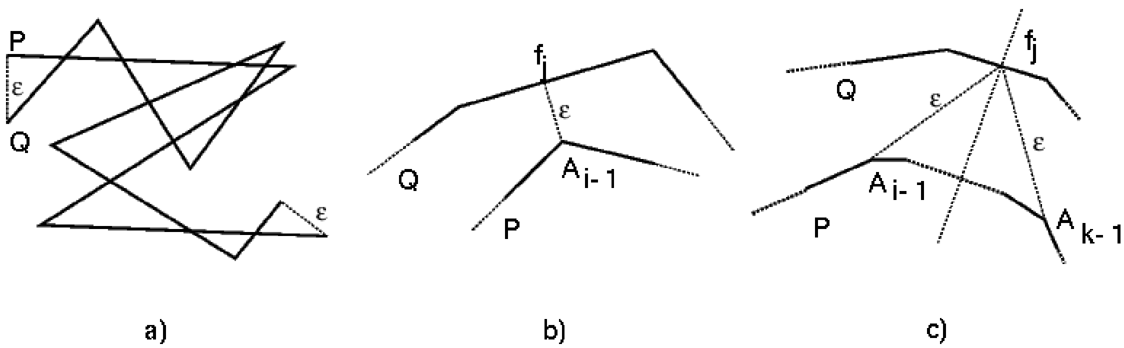


Figure 2.2: Three kinds of critical values of the Fréchet distance between curves [3].

Most of the existing algorithms for computing the Fréchet distance between curves are based on these two techniques, free space diagrams and critical values. Notice that the exact decision algorithm uses dynamic programming on the free space diagram, which visits all cells in the free space diagram, so the running time is at least $O(nm)$. Hence, the high level idea to reduce the running time is trying to reduce the number of cells visited in the free space diagram.

Curves with long edges. Researchers have studied the Fréchet distance for restricted classes of curves. In this thesis, we are particularly interested in the restricted class of curves with relatively long edges that Gudmundsson et al. [17] studied and its possible generalization to surfaces. Given P and Q with Fréchet distance δ, they show fast algorithms for computing δ assuming the edges of P and Q are longer than 2δ. They also show a similar result assuming one curve is composed of 4δ-long edges and the other of edges with arbitrary length. In high level, long edges guarantee that a straight segment only has one combinatorially distinct path in the free space diagram, thus, it heavily restricts options for the path that connects the lower left to the upper right in the free space diagram. Therefore, this path can be computed quickly if it exists. Specifically, they show that the greedy mapping from the prefix subcurve of P to the maximum reachable prefix subcurve of Q finds this path, and therefore a mapping of length at most δ. In this thesis, we ask whether it is possible to obtain similar polynomial time exact or approximation algorithms for surfaces given that they consist of triangles whose inscribed circles are large relative to δ.

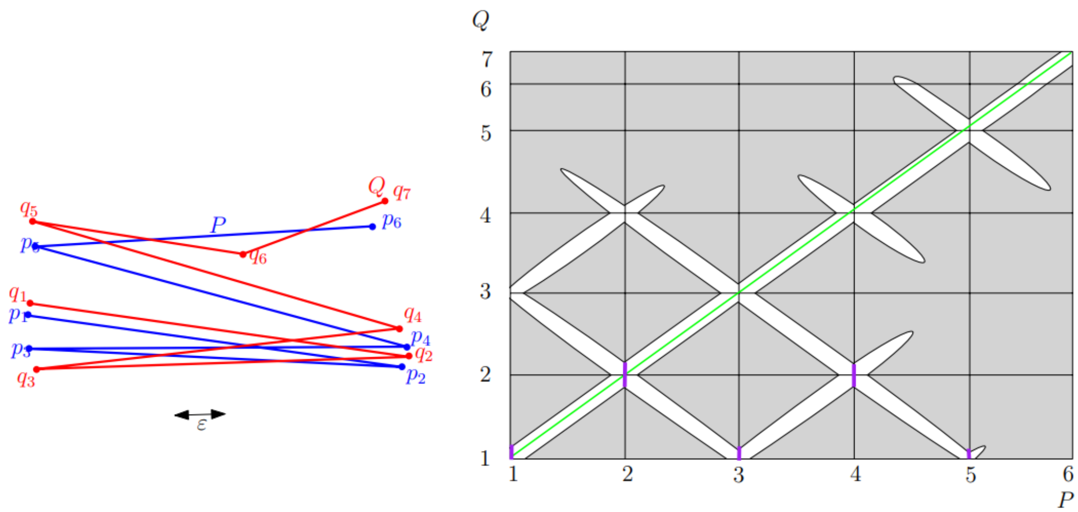


Figure 2.3: The free space diagram for curves with long edges [17].

2.1.2 Approximation Algorithms for Curves

It is still an open problem to compute or approximate the Fréchet distance between two general curves with subquadratic running time. Currently, there are a few restricted classes of curves, for which we can approximate the Fréchet distance quickly. One useful technique for the approximation algorithms for restricted classes of curves is curve simplification. Curve simplification is a very natural technique that has been very effective for curves. We ask, in this thesis, whether a similar technique can work for surfaces.

Curve simplification. Given a polygon curve $P = \{p_1, \ldots, p_n\}$, let $Q = \{q_1, \ldots, q_m\}$ be the μ-simplified curve of Q. we obtain Q as follows:

1. Base case: $q_1 = p_1$.

2. Inductive step: (1) Let $q_i = p_j$, and p_k be the first vertex out of $Ball(p_j, \mu)$ in $\{p_{j+1}, \ldots, p_n\}$, then $q_{i+1} = p_k$. (2) If all points of $\{p_{j+1}, \ldots, p_n\}$ are in $Ball(p_j, \mu)$, we set p_n as the last vertex of Q.

The μ-simplified curves have two useful properties. First, the Fréchet distance between the simplified curve and the original curve is at most μ. Second, the lengths of the simplified edges except possibly the last one are at least μ. Several $(1 + \varepsilon)$-approximation algorithms are obtained for different restricted families of curves based on these two properties, detailed below.

Aronov et al. [4] give a near linear time $(1 + \varepsilon)$-approximation algorithm for the discrete Fréchet distance between backbone curves. Driemel, et al. present a $(1 + \varepsilon)$-approximation algorithm for the Fréchet distance between c-packed curves and the Fréchet distance between k-bounded curves.

Curves type	Problem	Running time	See
c-packed	Fréchet distance	$O(cn/\epsilon + cn \log n)$	Driemel, et al. [14]
κ-bounded	Fréchet distance	$(\kappa/\epsilon)^d n + \kappa^d n \log n$	Driemel, et al. [14]
backbone	discrete Fréchet distance	$O(n^{4/3} \log nm)$	Aronov, et al. [4]

2.2 Fréchet Distance between Surfaces

Computing the Fréchet distance becomes much harder for surfaces, as the main techniques do not easily generalize to this higher dimensional variant of the problem. First, the free space diagram is the main technique used for the decision algorithms for the Fréchet distance between curves. However, the free space diagram for two surfaces is not even easy to define. A straightforward definition would give us a subset of a 4-dimensional rectangular cube as our free space. The homeomorphism, in this setting, would correspond to surface in this 4-dimensional cube. For curves, we use the linear ordering of points to find a bi-monotonic path in the free space diagram quickly. Clearly, we cannot apply the same idea to the free space diagram of surfaces, because surfaces don't have the linear ordering of points.

Second, the exact algorithm for computing the Fréchet distance between curves searches within a polynomial size set of critical values for the value of the Fréchet distance. For surfaces, we still do not know how to compute a finite set of critical values for the Fréchet distance between general surfaces. Therefore, even given an exact decision algorithm to check whether the Fréchet distance between two surfaces is at most a given δ, we can not extend the decision algorithm to an exact algorithm for computing the Fréchet distance between surfaces. Computing a finite set of critical values is an open problem we propose in this thesis.

There are only a few papers regarding computing the Fréchet distance between surfaces, perhaps because of the inherent difficulty of the problem. We categorize these papers into positive or algorithmic results versus negative or hardness results.

2.2.1 Algorithms

To our knowledge, there was no exact or approximation algorithm that is guaranteed to terminate in finite time for general surfaces before the work of this thesis. However, there was an algorithm that generates a sequence of numbers converging to the Fréchet distance.

Semi-computability Alt and Buchin [2] show that the Fréchet distance between triangulated surfaces is upper semi-computable: they describe an algorithm that runs forever but whose output converges to the Fréchet distance. The algorithm keeps refining two triangulated surfaces $\mathcal{R}$ and $\mathcal{S}$ to surfaces $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ with smaller and smaller triangles. For each pair of triangulations the algorithm enumerates all possible mesh homeomorphisms. Mesh homeomorphisms map each edge of $\widetilde{\mathcal{R}}$ to a path in the 1-skeleton of $\widetilde{\mathcal{S}}$; therefore, it is possible to enumerate mesh homeomorphism. Since the algorithm runs forever, the triangles of $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ are going to become arbitrarily small, so the enumeration process is guaranteed to converge to an optimal homeomorphism between the surfaces.

2.2.2 Restricted Classes of Surfaces

Computing the Fréchet distance between general surfaces is hard, but solving this problem is straightforward for two triangles. A very useful observation is that the Fréchet distance between two triangles is the same as the Fréchet distance between their boundary curves. In fact, this property holds for two convex polygons possibly in two different planes: the Fréchet distance between two convex polygons is the same as the Fréchet distance between their boundary curves. The case for non-convex polygons is much more complicated as detailed below.

Simple Polygons. A simple polygon is defined by its boundary: a non-self-intersecting piecewise linear curve in a plane. Buchin et al. [8] present a polynomial time algorithm to compute the Fréchet distance between two simple polygons P and Q (possibly on different planes). First, they show that the Fréchet distance between a convex polygon and a simple polygon equals the Fréchet distance between their boundary curves. For two possibly non-convex simple polygons P and Q, their algorithm first decomposes P into convex polygons. Then, it looks for the optimal image for each convex subpolygon of P in Q. Note, that after finding these images, the problem reduces to a set of Fréchet distance computation problems between curves.

After decomposing P into convex polygons, the authors need to find the image of each diagonal of P added for the decomposition. The authors observe that it is safe to assume that the image of a diagonal in an optimal map is a shortest path in the other polygon Q between two points on the boundary of Q. Therefore, deciding whether the Fréchet distance between P and Q is at most δ reduces to deciding whether an orientation preserving homeomorphism from the boundary of P together with the diagonals added for the decomposition into convex polygons exists into Q such that the following properties hold: (1) the boundary of P should map to the boundary of Q, (2) each diagonal on the P side should map to a shortest path, and (3) the Fréchet length of the map must be at most δ. The authors refer to a monotone path with condition (2) as a feasible path in the free space diagram of P, Q and δ. The algorithm checks the condition (2) by building a reachability graph. For each vertex p of P, the algorithm computes a maximum interval $I(p, e)$ on each boundary edge e of Q such that the distance between p and

$I(p, e)$ is no more than δ; the number of intervals is $O(nm)$. Then, we are able to build a reachability graph, such that a vertex in the graph represents an interval and a directed edge from $I(p_0, e_0)$ to $I(p_1, e_1)$ represents the existence of a map from the diagonal (p_0, p_1) of P to a shortest path from $I(p_0, e_0)$ to $I(p_1, e_1)$ in Q with Fréchet length no more than δ. To find such a shortest path, the authors define the set of all shortest paths from $I(p_0, e_0)$ to $I(p_1, e_1)$ as an hourglass. They show that if there exists one shortest path in the hourglass with Fréchet distance at most δ to the diagonal (p_0, p_1), then all shortest paths in the hourglass have Fréchet distance at most δ to the diagonal (p_0, p_1). Once we have the reachability graph,if and only if two endpoints of a monotone path are connected in the reachability graph, the monotone path is a feasible path in the graph.

For computing the Fréchet distance, the algorithm performs a binary search over a set of critical values, and checks each value using the decision algorithm. Because each edge of the decomposition of P maps to a shortest path in Q, we enumerate all pairs of a diagonal and an hourglass, and hence a set of critical values can be computed in polynomial time.

Folded Polygons. A folded polygon is a polygonal surface that does not have any interior vertices. Cook et al. [12] generalize the work of Buchin et al. [8] to a polynomial time algorithm for folded polygons. First, the authors show that the Fréchet distance between a convex polygon and a folded polygon equals the Fréchet distance between their boundary curves. The core idea of their algorithms is also decomposing P into convex polygons, and looking for the optimal image for each convex sub-polygon. Because the Fréchet distance between each convex sub-polygon and its image, a folded polygon, equals the Fréchet distance between their boundary curves, the problem reduces to looking for the optimal homeomorphism for each diagonal and boundary edge. However, it is not enough to consider only mapping diagonals to shortest paths on Q. Fortunately, the paths on Q that diagonals of P map to still have some good properties. The authors show that there exists an optimal map, such that it maps diagonals to a polygonal path that follows the shortest path between two endpoints. If a path goes through the same sequence of faces as the shortest path, we say that the path follows the shortest path. Based on this observation, the authors describe three algorithms:

(1) a fixed-parameter tractable algorithm, where the number of diagonals in a convex sub-polygon of P and Q is constant. This algorithm is similar to our exact algorithm for computing the Fréchet distance between surfaces: both of them reduce to a global optimization problem with a set of constraints. (2) a constant-factor approximation algorithm. The algorithm first finds a shortest path map for diagonals, such that the image paths may cross others. Then, the algorithm fixes the image paths to non-crossing paths one-by-one from outside diagonals to inside diagonal. The authors show that it gives a 9-approximation of the Fréchet distance between P and Q. (3) an exact algorithm for computing the Fréchet distance between axis-parallel folds under the l_∞ norm. For this special case, the authors show that it suffices to use a shortest path as the optimal image of a diagonal, and then the algorithm for simple polygons works.

Polygons with constant numbers of holes Nayyeri and Sidiropoulos [22] describe a polynomial time algorithm for polygons with constant numbers of holes. The authors show how to build a bounded set of candidates for the homotopy classes of the diagonals in the optimal homeomorphism. Their main technique is a shortcutting argument that shows unnecessarily complicated images (of the diagonals) can be simplified without changing the Fréchet length of the map. In this thesis, we use similar ideas to combinatorially constraint maps for our problems.

2.2.3 Hardness Results

Godau [16] showed that computing the Fréchet distance between a self-intersecting surface and a triangle is NP-hard. Buchin, Buchin and Schulz [7] extended this work to show that computing the Fréchet distance between two-dimensional terrains and between polygons with holes is also NP-hard.

Chapter 3: Preliminaries

Surfaces. A **surface** $\mathcal{Q}$ (or a 2-manifold) is a space in which every point has a neighborhood that is homeomorphic to the plane or half-plane. The set of points that are homeomorphic to the half-plane form the **boundary** of $\mathcal{Q}$. An **embedding** $\Phi : \mathcal{Q} \to \mathbb{R}^3$ is a continuous one-to-one map. An **immersion** $\varphi : \mathcal{Q} \to \mathbb{R}^3$ is a continuous map, such that for any $x \in \mathcal{Q}$ there is a neighborhood N_x of x on which f is an embedding.

A **piecewise linear surface** is a surface $\mathcal{Q}$ that is constructed from a set of Euclidean triangles by identifying pairs of equal-length edges. We denote the constituent vertices, edges, and triangles of $\mathcal{Q}$ by $\mathcal{Q}_V$, $\mathcal{Q}_E$, and $\mathcal{Q}_T$, in order. In short, we write $\mathcal{Q} = (\mathcal{Q}_V, \mathcal{Q}_E, \mathcal{Q}_T)$. In this paper, we consider **locally isometric immersions**, those that map each triangle to a congruent triangle in $\mathbb{R}^3$.

Maps. Let $f : A \to B$ be a function. For any $U \subseteq A$, we define $f(U) = \{f(u)|u \in U\}$. The function $f|_U : U \to B$, called the **restriction** of f to the subset U, is defined as for all $u \in U$, $f|_U(u) = f(u)$. In this case, we also say, that f is an **extension** of $f|_U$ to A. If A and B are topological space, f is a **homeomorphism** if (1) it is a bijection, (2) it is continuous, and (3) its inverse is continuous.

Ball. Let (M, d) be a metric space. We define $Ball(p, r)$ as the ball of radius $r > 0$ centered at a point p in M, $Ball(p, r) = \{x \in M \mid d(x, p) \leq r\}$.

Discrete Fréchet distance for curves. Let $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_m)$ be two polygonal curves. Their discrete Fréchet distance is defined as

$$\delta_D(P, Q) = \min_{M} \max_{(p,q)\in M} d(p, q)$$

where M ranges over all order-preserving complete correspondences between P and Q.

Fréchet distance for curves. Let A and B be two given curves, and $\sigma : A \to B$ be a homeomorphism. The Fréchet length of σ is $\delta_F(\sigma) = \max_{x \in A} ||x - \sigma(x)||_2$. The Fréchet distance between A and B is

$$\delta_F(A, B) = \inf_{\sigma} \delta_F(\sigma).$$

where σ ranges over all homeomorphisms between P and Q.

Fréchet length of homeomorphisms. Let $\mathcal{R}$ and $\mathcal{S}$ be two surfaces, and $\sigma : \mathcal{R} \to \mathcal{S}$ be a homeomorphism. The **Fréchet length of** σ is

$$\delta_F(\sigma) = \max_{x \in \mathcal{R}} ||x - \sigma(x)||_2$$

.

Fréchet distance for surfaces. Let $\mathcal{R}$ and $\mathcal{S}$ be two surfaces. The Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ is

$$\delta_F(\mathcal{R}, \mathcal{S}) = \inf_{\sigma} \delta_F(\sigma),$$

where σ ranges over all homeomorphisms between $\mathcal{R}$ and $\mathcal{S}$.

Existential theory of the reals. Let $x_1, \ldots, x_n$ be variables over reals, and let $F(x_1, \ldots, x_n)$ be a quantifier-free formula involving real polynomial equalities and inequalities. The decision problem for the existential theory of the reals is to decide if the following formula is true:

$$\exists x_1 \cdots \exists x_n F(x_1, \ldots, x_n).$$

The problem is to decide whether real numbers $\overline{x}_1 \ldots \overline{x}_n$ exist such that $F(\overline{x}_1, \ldots, \overline{x}_n)$ is true. Canny [11] shows that the existential theory of the reals can be decided in PSPACE. Note this is polynomial space under the Turing Machine model, that is, the required space is a polynomial function of the number of bits used to specify the problem.

Lemma 3.0.1 (Canny [11], Theorem 3.3)**.** *The existential theory of the reals is decidable in PSPACE.*

Semidefinite Program. Let $c \in \mathbb{R}^m$ be a vector, and $x \in \mathbb{R}^m$ be a variable vector. A **semidefinite program** is an optimization problem of minimizing a linear function of x subject to a matrix inequality:

$$\begin{array}{ll} \text{minimize} & c^t x \\ \text{subject to} & \sum_{k \in [m]} x_k A^{(k)} - B \succeq 0 \end{array}$$

where $A^{(1)}, \ldots, A^{(m)}, B \in \mathbb{R}^{n \times n}$ are symmetric matrices, and $M \succeq 0$ means that M is positive semidefinite.

Lemma 3.0.2 (Ben-Tal et al. [6], Section 4.6.3)**.** *A semidefinite program can be solved by an interior point method in*

$$O(n^{0.5}m^3 + n^{2.5}m^2 + n^{3.5}m) \log(1/\epsilon)$$

time, where n is the number of variables, m is the number of constraints, and ϵ is the additive error.

Linear Program. A **Linear program** is a special case of a semidefinite program, where all matrices are diagonal.

Lemma 3.0.3 (Ben-Tal et al. [6], Section 4.6.1)**.** *A linear program can be solved in*

$$O(n^2(m+n)^{3/2} \log(1/\epsilon))$$

time, where n is the number of variables, m is the number of constraints, and ϵ is the additive error.

Chapter 4: Overview

Before we get into details, it's better to understand the ideas and keep the big picture in mind. Let's start with computing the Fréchet distance between curves. Later we show that similar ideas are helpful for computing the Fréchet distance between surfaces.

Let A and B be two piecewise linear curves with Fréchet distance δ. The goal of exact or approximation algorithms is to compute a homeomorphism $f : A \to B$ with Frechet length (close to) δ. Computing such an f reduces to computing the optimal vertex map $(g : A_V \to B, h : B_V \to A)$ that specifies f restricted to the vertices of A and B (A_V and B_V, respectively). This reduction holds because g and h can be extended to a full homeomorphism between A and B with the same Fréchet length as the maximum of g's and h's. Therefore, to compute the Fréchet distance between curves, we only need to worry about computing the vertex map (g, h).

To compute an optimal vertex map, the classic algorithm will search a set of candidate values of the Fréchet distance. To build the set of candidate values, for each vertex, the algorithm computes a set of candidate locations of its image. The set of candidate values of the Fréchet distance is composed of all distances between each vertex and its candidate images. However, for surfaces, we cannot compute such a list of candidate locations of images for each vertex. Now, we will show another way to compute the Fréchet distance between curves. It is slower and more complex than the classic algorithm but can be extended to computing Fréchet distance between surfaces.

Let A_E and B_E be the edge sets of A and B. For piecewise linear curves, computing the vertex map (g, h) can be simplified to computing the combinatorial version of the vertex map $(g^c : A_V \to B_E, h^c : B_V \to A_E)$. For each v in A_V, $g^c(v)$ specifies which edge of B_E contains $g(v)$ (similarly, $h^c(v)$ specifies which edge of A_E contains $h(v)$), instead of specifying the exact location of $g(v)$. Therefore, there are only finitely many choices for g^c and h^c. A slow algorithm may try all possible choices for g^c and h^c. For each choice, we would like to compute g and h that are consistent with g^c and h^c and that extend to a homeomorphism with small Fréchet length. We consider two approaches leading to an approximation algorithm and an exact algorithm respectively.

We observe that misplacing the images of vertices within an edge can increase the Fréchet length by at most the length of the edge. Thus, if the maximum edge length is r, then any vertex map that is consistent with the optimal combinatorial vertex map can be extended to a homeomorphism of length at most $\delta + r$. This leads to a $(1 + \varepsilon)$-approximation algorithm by subdividing edges to edges of length $\epsilon\delta$. Note that the running time of the algorithm will depend on the lengths of the input curves as well as their complexities.

A main ingredient of our exact algorithms is a decision procedure that can decide for any given value δ whether the Fréchet distance between A and B is smaller than or equal to δ. Given g^c and h^c, we can formulate this decision problem as a set of (quadratic) inequalities. We will have two types of inequalities: (i) inequalities that ensure the images of vertices appear in order on the other curve, and (ii) inequalities that ensure the distance between any vertex and its image is at most δ. Inequalities of type (i) are linear and those of type (ii) are quadratic. Hence, the problem reduces to solving a system of inequalities.

We have understood how to compute the Fréchet distance between curves using a combinatorial vertex map. Next, we generalize this idea to compute the Fréchet distance between surfaces. Let $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$ be two piecewise linear surfaces with Fréchet distance δ. Our goal is to compute a homeomorphism $f : \mathcal{R} \to \mathcal{S}$ that realizes the Fréchet distance δ. Computing f can reduce to computing the optimal edge map $f_1 : \mathcal{R}_E \to \mathcal{S}$ that specifies f restricted to the edges of $\mathcal{R}$, because f_1 can be extended to a full homeomorphism between $\mathcal{R}$ and $\mathcal{S}$ with the same Fréchet length as the maximum of the f_1's.

We show that there exists an optimal edge map f_1', such that for each edge e in $\mathcal{R}_E$, the image $f_1'(e)$ is a piecewise linear curve. Then, computing the Fréchet distance between a line segment and a piecewise linear curve reduces to computing the vertex map from $f_1'(e)$ to e. There are three types of vertices of $f_1'(e)$: (i) vertices of $\mathcal{S}_V$, (ii) images of vertices of $\mathcal{R}_V$, $f_1'(\mathcal{R}_V)$, and (iii) points on edges of $\mathcal{S}_E$. In this thesis, we refer to these edge maps as **scaffold maps**, and the type (iii) vertices of $f_1'(e)$ as **crossing points**.

We introduce a combinatorial version of scaffold maps. It is composed of two parts: a combinatorial vertex map ($g^c : \mathcal{R}_V \to \mathcal{S}_T, h^c : \mathcal{S}_V \to \mathcal{R}_T$) and $f^c : \mathcal{R}_E \to Q$ where Q is a sequence of vertices of $\mathcal{S}_V$, and edges of $\mathcal{S}_E$. For each v in $\mathcal{R}_V$, $g^c(v)$ specifies which triangle of $\mathcal{S}_T$ contains $f_1'(v)$. For each edge e in $\mathcal{R}_E$, $f^c(e)$ specifies the sequence of vertices and edges that are crossed by $f_1'(e)$.

Similarly to the curve case, we observe that misplacing the images of vertices and crossing points on the edges can increase the Fréchet length by at most the length of the edge. Let r be the maximum edge length. We show that any scaffold map that is consistent with the optimal combinatorial scaffold map can be extended to a homeomorphism of length at most $\delta + r$. We obtain a $(1 + \epsilon)$-approximation algorithm by refining the surfaces into triangulations composed of triangles of diameter $O(\epsilon\delta)$.

Note that $f^c(e)$ specifies a sequence of vertices and edges, and it may have arbitrary length. We show that there exists an optimal combinatorial scaffold map, in which the length of $f^c(e)$ is bounded for all edges. Thus, we can build a finite-size list L of combinatorial scaffold maps that contains the optimal scaffold map.

To obtain our exact algorithm, we build a system of inequalities for each combinatorial scaffold map f^c in L. The system is feasible if and only if there exists a scaffold map that is consistent with f^c and has Fréchet length at most δ. There are four types of inequalities: (i) inequalities that ensure the images of vertices lie in the correct triangles. (ii) inequalities that ensure the crossing points appear in order on edges. (iii) inequalities that ensure the relation of images of vertices and images of edges. (iv) inequalities that ensure the distance between a vertex or crossing point and its image is at most δ. The inequalities of type (iii) ensure that images of vertices and edges are consistent with the combinatorial scaffold map f^c, and it will convert to inequalities for maintaining the refinement of $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ in our system.

Our algorithm builds a system of polynomial inequalities, and the number of variables and the number of constraints is $O(2^{|\mathcal{R}_V|+|\mathcal{S}_V|})$. For the case that $\mathcal{S}$ is a triangle, we show that the system of inequalities has polynomial size. Hence, the Fréchet distance between a triangulated surface and a triangle can be decided in PSPACE.

Chapter 5: Scaffold Map and Its Properties

In this chapter, we introduce the notions of vertex maps, refinements, and scaffold maps. As earlier, the scaffold map is the core component of our algorithm, mapping edges to the other surface. First, we show that a scaffold map is able to extend to a full homeomorphism with the same Fréchet length as the maximum of the scaffold map's. Second, we show that there exists an optimal scaffold map f_1', such that for any edge $e \in \mathcal{R}_E$, $f_1'(e)$ crosses edges of $\mathcal{S}$ a bounded number of times.

5.1 Vertex Maps and Refinements

First, we define vertex maps, which either map vertices to points on the other surface, or map points to vertices on the other surface.

Definition 5.1.1. *Let $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$ be two piecewise linear surfaces. A bijection $f_0 : \widetilde{\mathcal{R}}_V \to \widetilde{\mathcal{S}}_V$ is a **vertex map** if and only if it has the following properties.*

(1) $\widetilde{\mathcal{R}}_V = \mathcal{R}_V \cup f_0^{-1}(S_V)$, $\widetilde{\mathcal{S}}_V = \mathcal{S}_V \cup f_0(R_V)$.

(2) f_0 maps boundary vertices of $\widetilde{\mathcal{R}}_V$ to boundary vertices of $\widetilde{\mathcal{S}}_V$, and it preserves the cyclic order of boundary vertices on each boundary component.

Given a vertex map, to define the scaffold map more easily, we refine the surfaces by $\widetilde{\mathcal{R}}_V$ and $\widetilde{\mathcal{S}}_V$ so that f_0 becomes a vertex-to-vertex map on $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$.

Definition 5.1.2. *Let $f_0 : \widetilde{\mathcal{R}}_V \to \widetilde{\mathcal{S}}_V$ be a vertex map between $\mathcal{R}$ and $\mathcal{S}$; $\widetilde{\mathcal{R}}_V = \mathcal{R}_V \cup f_0^{-1}(S_V)$ and $\widetilde{\mathcal{S}}_V = \mathcal{S}_V \cup f_0(R_V)$. Let $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ and $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$ be geometric refinements of $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$, respectively.*

We say that $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ and f_0 are **consistent**, as the preimage and image of f_0 are the vertex sets of $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$, respectively.

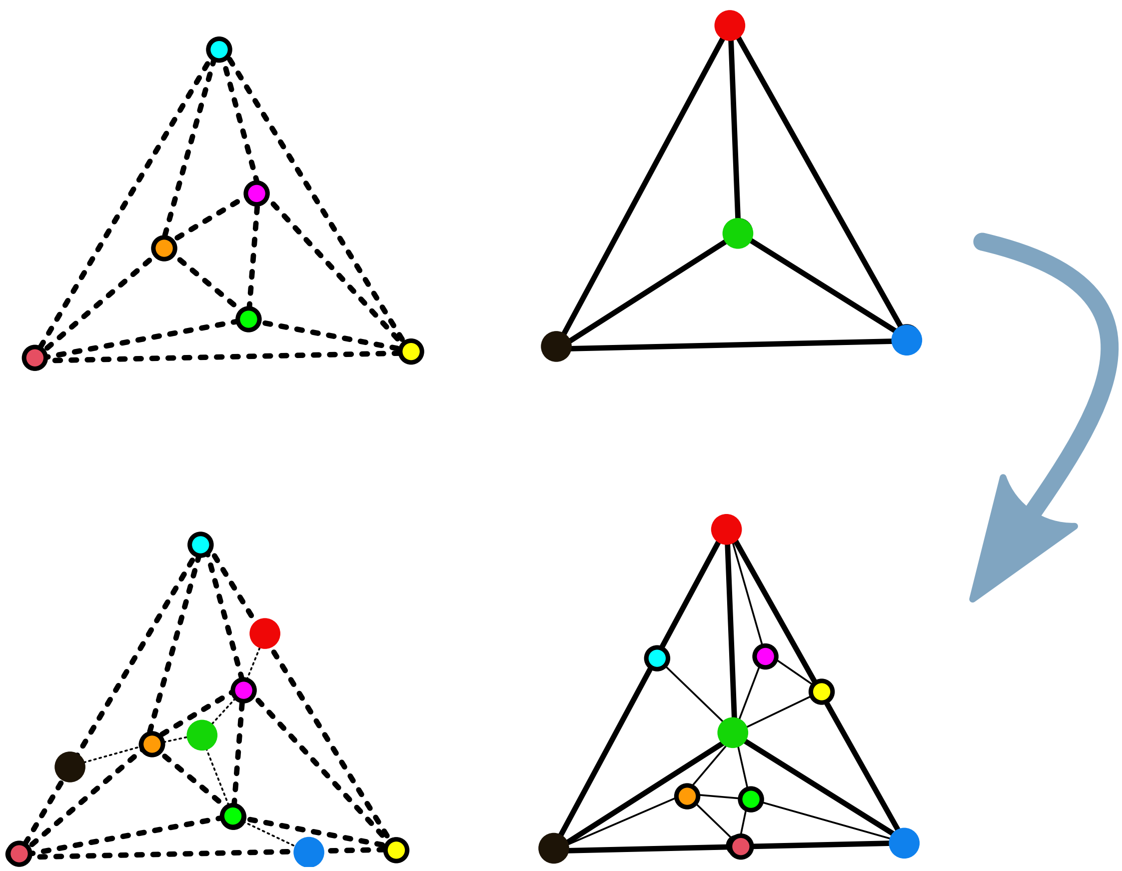


Figure 5.1: upper: $\mathcal{R}$ and $\mathcal{S}$ before refinement, lower: $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ after refinement. The cyclic order of boundary vertices is preserved.

Lemma 5.1.3. *For any vertex map f_0, there are $2^{O((m+n)^2)}$ refinements that are consistent with f_0.*

Proof. Because $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ are planar graphs, they are subgraphs of the complete graph with $m+n$ vertices. The number of subgraphs of the complete graph K_{m+n} is $2^{O((m+n)^2)}$. Therefore, the number of possible refinements is bounded by $2^{O((m+n)^2)}$. □

5.2 Scaffold Maps

We are ready to define scaffold maps. A scaffold map can be viewed as a collection of maps from the edges in $\widetilde{\mathcal{R}}_E$ to the underlying surface of $\widetilde{\mathcal{S}}$.

Definition 5.2.1. *Let $f_0 : \widetilde{\mathcal{R}}_V \to \widetilde{\mathcal{S}}_V$ be a vertex map. Let $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ and $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$ be refinements of $\mathcal{R}$ and $\mathcal{S}$, respectively, that are consistent with f_0. A* ***scaffold map*** *(over refinements $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$) is a continuous one-to-one map $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ with the following properties.*

(1) $f_1(\widetilde{\mathcal{R}}_V) = f_0(\widetilde{\mathcal{R}}_V)$.

(2) f_1 is a cellular embedding of $(\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E)$ on $\widetilde{\mathcal{S}}$.

(3) f_1 maps boundary edges to boundary edges (so, it preserves the cyclic order of boundary edges around boundary components).

(4) f_1 preserves the cyclic order of edges around each vertex: for any $u \in \widetilde{\mathcal{R}}_V$ with neighbors $\{w_1, \ldots, w_k\}$, the cyclic order of the edges $\{(u, w_1), (u, w_2), \ldots, (u, w_k)\}$ around u is identical to the cyclic order of curves $\{f_1(u, w_1), \ldots, f_1(u, w_k)\}$ around $f_1(u)$.

(5) For each $e \in \widetilde{\mathcal{R}}_E$ and each $t \in \widetilde{\mathcal{S}}_T$, $f_1(e) \cap t$ is a collection of straight line segments that intersect $\partial(t)$ at their endpoints.

Properties (1) to (4) are formal definitions of the intuition of edge maps. In addition, scaffold maps do not just map edges to arbitrary curves on the other surface. Property (5) requires that all intersections between edge images and triangles are straight line segments. The reason for this extra requirement will be detailed in the following section. Briefly, we will introduce a technique called a **shortcutting operation**. For a curve c and a line segment l, the shortcutting operation allows us to replace any subcurve of c by a line segment to reduce the Fréchet distance between c and l. Clearly, an arbitrary edge map can be modified to a scaffold map without increasing the Fréchet length. Moreover, property (5) will guarantee that any scaffold map can be extended to a homeomorphism.

5.3 Scaffold Maps and Homeomorphisms

In this section, we show the relation between scaffold maps and homeomorphisms. The following lemma is our goal in this section.

Lemma 5.3.1. *The relation between scaffold maps and homeomorphisms:*

(1) A scaffold map f_1 can be extended to an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism with Fréchet length arbitrarily close to $\delta_F(f_1)$.

(2) A scaffold map f_1 of Fréchet length δ can be obtained from an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism h of Fréchet length δ.

Lemma 5.3.1 will allow us to focus on scaffold maps, because any scaffold map can be extended to a full homeomorphism with the same Fréchet length. Also, the second statement will guarantee that there always exists an optimal scaffold map.

Before proving the first statement, we give the formal definition of the Fréchet length of the scaffold map.

Definition 5.3.2. *The* ***Fréchet length of the scaffold map*** *f_1 is the maximum Fréchet length of all its restrictions to edges $e \in \widetilde{\mathcal{R}}_E$, denoted*

$$\delta_F(f_1) = \max_{e \in \widetilde{\mathcal{R}}_E} \delta_F(f_{1|e}).$$

Now, we will use the following lemma to show that a scaffold map $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ can be extended to an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism of Fréchet length arbitrarily close to $\delta_F(f_1)$.

Lemma 5.3.3 (Cook et al. [12])**.** *Let t be a triangle, p be a folded polygon with n triangles, and $g : \partial(t) \to \partial(p)$ a homeomorphism. For any $\epsilon > 0$, the map g can be extended to a homeomorphism, $h : t \to p$, for which $\delta_F(h) \leq \delta_F(g) + \epsilon$, in polynomial time in n.*

Proof of Lemma 5.3.1 (1). Let f_1 be an arbitrary scaffold map, and let t be a triangle in $\widetilde{\mathcal{R}}_T$, and let (e_1, e_2, e_3) be the edges of t. Images $f_1(e_1), f_1(e_2)$ and $f_1(e_3)$ are piecewise linear curves. Let p be the folded polygon bounded by $f_1(e_1), f_1(e_2)$ and $f_1(e_3)$.

Because f_1 is a homeomorphism from $\partial(t)$ to $\partial(p)$, Lemma 5.3.3 shows that f_1 can be extended to a homeomorphism $h : t \to p$ for which $\delta_F(h) \leq \delta_F(g) + \epsilon$. The one-to-one correspondence of triangles to folded polygons and Lemma 5.3.3 imply that a scaffold map can be extended to an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism of Fréchet length arbitrarily close to $\delta_F(f_1)$. □

To prove the second part, we introduce the shortcutting operation, which is also our main tool to show the property (5) of the scaffold map.

Definition 5.3.4. *Let $\alpha : [0,1] \to \mathbb{R}^d$ be an immersed curve, let $0 \leq t_1 < t_2 \leq 1$, and let $\ell : [t_1, t_2] \to \mathbb{R}^d$ be a line segment with endpoints $\alpha[t_1]$ and $\alpha[t_2]$. Finally, let $\alpha' : [0,1] \to \mathbb{R}^d$ be $\alpha[0,t_1) \cup \ell[t_1,t_2] \cup \alpha(t_2,1]$, that is, α' coincides with α in $[0,t_1) \cup (t_2,1]$, and coincides with the line segment ℓ on $[t_1,t_2]$. We say that α' is obtained from α via a* ***shortcutting operation****.*

Lemma 5.3.5 (Lemma 3 of Buchin et al. [8])**.** *Let $\alpha : [0,1] \to \mathbb{R}^d$ and $\alpha' : [0,1] \to \mathbb{R}^d$ be two curves, and let s be a line segment. If α' is obtained from α via a sequence of shortcutting operations then $\delta_F(\alpha', s) \leq \delta_F(\alpha, s)$.*

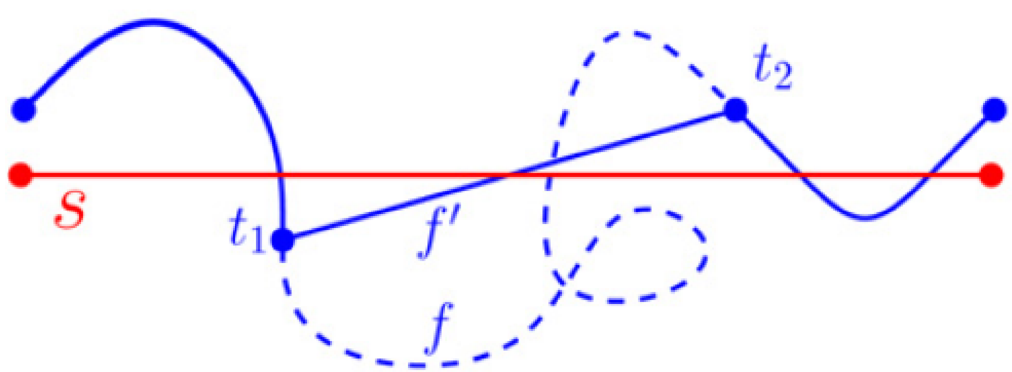


Figure 5.2: The Fréchet distance between a curve and a linear segment is not increased if we replace part of the curve with a line segment.(Figure 3 of Buchin et al. [8])

Lemma 5.3.5 shows that the shortcutting operations do not increase the Fréchet length. This lemma also implies that an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism f can reduce to a scaffold map f_1 of Fréchet length $\delta_F(f)$.

Proof of Lemma 5.3.1 (2). Let $\widetilde{\mathcal{R}}_V = \mathcal{R}_V \cup h^{-1}(\mathcal{S}_V)$, and $f_0 = h|_{\widetilde{\mathcal{R}}_V}$ be the vertex map. Let $\Gamma = \{\gamma_e = h(e) | e \in \widetilde{\mathcal{R}}_E\}$. For any triangle $t \in \widetilde{\mathcal{S}}_T$, if $\Gamma \cap t$ is not a set of line segments, we apply shortcutting operations to $\Gamma \cap t$. Let $\Lambda = \{\lambda_e | e \in \widetilde{\mathcal{R}}_E\}$ be the set of linear piecewise paths obtained from Γ via shortcutting operations. By Lemma 5.3.5, for any $e \in \widetilde{\mathcal{R}}_E$, $\delta_F(e, \lambda_e) \leq \delta_F(e, h(e))$. Then, there exists a map $f_1 : \widetilde{\mathcal{R}}_E \to \Lambda$ such that $\delta_F(f_1) \leq \delta_F(h)$. Clearly, f_1 satisfies property (5). Because shortcutting operations do nothing to the vertex map and the boundary edges, f_1 satisfies property (1) and (3). And applying shortcutting operations in each triangle does not change the embedding, so properties (2) and (4) will be satisfied. Therefore, we obtain a scaffold map f_1 from the homeomorphism h such that $\delta_F(f_1) \leq \delta_F(h)$. □

5.4 Crossing Number and Crossing Bound

In the previous section, we showed that a scaffold map f_1 of Fréchet length $\delta_F(f)$ can be obtained from an $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$-homeomorphism f. However, for each edge $s \in \widetilde{\mathcal{S}}$, $f_1(\widetilde{\mathcal{R}}_E)$ can intersect s an arbitrary number of times. In this section, we show that for any given scaffold map f_1, we can extract a new scaffold map f_1' from f_1 without increasing Fréchet length, such that $f_1'(\widetilde{\mathcal{R}}_E)$ only intersects each edge $s \in \widetilde{\mathcal{S}}$ a bounded number of times. This property enables us to enumerate the combinatorial scaffold maps in our algorithms.

Given a scaffold map f_1, we can split f_1 into two different types of mappings f_{1e} and f_{1p}. f_{1e} is the edge-to-edge map that maps edges in $\widetilde{\mathcal{R}}_E$ to $\widetilde{\mathcal{S}}_E$, and f_{1p} maps edges to piecewise linear paths on $\widetilde{\mathcal{S}}$. f_{1e} maps line segments to line segments, which is not interesting. f_{1p} is what we are interested in. We define $f_{1p}(\widetilde{\mathcal{R}}_E) \cap \widetilde{\mathcal{S}}_E$ as the **crossing points set**. In this section, we show that there exists an optimal scaffold map f_1', such that $|f_{1p}'(\widetilde{\mathcal{R}}_E) \cap \widetilde{\mathcal{S}}_E| \leq 2^{|\widetilde{\mathcal{R}}_V| + |\widetilde{\mathcal{S}}_V|}$. This property allows us to enumerate the number of crossing points on each edge of the optimal scaffold map f_1'.

Definition 5.4.1. *The* ***crossing points set*** $\widetilde{\mathcal{S}}_X$ *is the set of all crossing points between*

$f_1(\widetilde{\mathcal{R}}_E)$ and $\widetilde{\mathcal{S}}_E$. Each element $y \in \widetilde{\mathcal{S}}_X$ is the crossing point of $f_1(e)$ and s for an $e \in \widetilde{\mathcal{R}}_E$ and $s \in \widetilde{\mathcal{S}}_E$; note that two edges may have multiple crossing points. The preimage of y, $x = f_1^{-1}(y)$, is a crossing point between e and $f_1^{-1}(s)$ on $\widetilde{\mathcal{R}}$ that corresponds to y. The preimages of all points in $\widetilde{\mathcal{S}}_X$ comprise the set of crossing points $\widetilde{\mathcal{R}}_X$ on $\widetilde{\mathcal{R}}$ that is in one-to-one correspondence with $\widetilde{\mathcal{S}}_X$.

Definition 5.4.2. *Let f_1 be a scaffold map, and let $e \in \widetilde{\mathcal{R}}_E$. The **crossing sequence** of $f_1(e)$ is the sequence of edges $(s_1, s_2, \ldots, s_k)$ of $\widetilde{\mathcal{S}}_E$ that $f_1(e)$ crosses in order.*

Definition 5.4.3. *For each $s \in \widetilde{\mathcal{S}}_E$, its crossing number $\chi(s)$ is the number of crossing points on s. The **crossing number** of a scaffold map f_1 is the maximum crossing number of edges of $\widetilde{\mathcal{S}}_E$, denoted*

$$\chi(f_1) = \max_{s \in \widetilde{\mathcal{S}}_E} \chi(f_1(s)).$$

The following lemma implies that we always can obtain a scaffold map f_1' from f with at most Fréchet length $\delta_F(f)$ and bounded crossing number via shortcutting operations.

Lemma 5.4.4 (Erickson and Nayyeri [15]). *Let $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ be a set of non-crossing curves on a triangulated surface (of genus zero) $\mathcal{Q} = (\mathcal{Q}_V, \mathcal{Q}_E, \mathcal{Q}_T)$. There exists a set of non-crossing curves $\Gamma' = \{\gamma_1', \gamma_2', \ldots, \gamma_k'\}$ with the following properties.*

(1) For each i, γ_i' is obtained from γ_i via a sequence of shortcutting operations along the edges in $\mathcal{Q}_E$.

(2) For each $\gamma' \in \Gamma'$ and $t \in \mathcal{Q}_T$, each connected component of $\gamma' \cap t$ is

(a) a path with endpoints on different sides of t, or

(b) a path with one point being a vertex of t and the other on its opposite side, or

(c) a side of t; in this case γ' coincides with the side of t.

(3) For each $e \in \mathcal{Q}_E$, if e is crossed by m different curves of Γ' then it is crossed at most 2^m times.

Lemma 5.4.5. *For any $\delta \geq 0$, the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ is at most δ if and only if there is a scaffold map of Fréchet length at most δ and crossing number at most $2^{|\widetilde{\mathcal{R}}_E|+|\widetilde{\mathcal{S}}_E|}$.*

Proof. By Lemma 5.3.1, a homeomorphism h of Fréchet length δ can reduce to a scaffold map f_1 of Fréchet length δ. Let $\Gamma = \{\gamma_e = f_1(e)|e \in \widetilde{\mathcal{R}}_E\}$, and let $\Lambda = \{\lambda_e|e \in \widetilde{\mathcal{R}}_E\}$ be the set of paths obtained from Γ via Lemma 5.4.4 that has properties (1), (2), and (3). Since Λ is obtained via shortcutting operations, for any $e \in \widetilde{\mathcal{R}}_E$, $\delta_F(e, \lambda_e) \leq \delta_F(e, \gamma_e)$. Because the total number of edges is $|\widetilde{\mathcal{R}}_E| + |\widetilde{\mathcal{S}}_E|$, for any $s \in \widetilde{\mathcal{S}}_E$, s is crossed at most $2^{|\widetilde{\mathcal{R}}_E|+|\widetilde{\mathcal{S}}_E|}$ times by Λ. Hence, we get a scaffold map of Fréchet length at most δ and crossing number at most $2^{|\widetilde{\mathcal{R}}_E|+|\widetilde{\mathcal{S}}_E|}$. □

Because the number of edges is bounded by the number of vertices, the crossing number is also bounded by $2^{O}(m + n)$, where m and n are the number of vertices of $\mathcal{R}$ and $\mathcal{S}$.

Chapter 6: Combinatorial Specification and Its Representation

In this chapter, we introduce combinatorial variants of vertex maps, refinements, and scaffold maps.

6.1 Combinatorial Vertex Maps

Definition 6.1.1. *A **combinatorial vertex map** (g, h) is composed of two maps:*

(1) $g : \mathcal{R}_V \to \mathcal{S}_T \cup \mathcal{S}_E$ that maps each internal vertex of $\mathcal{R}_V$ into a triangle of $\mathcal{S}_T$ and each boundary vertex of $\mathcal{R}_V$ into a boundary edge of $\mathcal{S}_E$.

(2) $h : \mathcal{S}_V \to \mathcal{R}_T \cup \mathcal{R}_E$ that maps each internal vertex of $\mathcal{S}_V$ into a triangle of $\mathcal{R}_T$ and each boundary vertex of $\mathcal{S}_V$ into a boundary edge of $\mathcal{R}_E$.

Intuitively, a combinatorial vertex map determines for each internal vertex $u \in \mathcal{R}_V$ the triangle of $\mathcal{S}$ that contains u's image, and for each boundary vertex $b \in \mathcal{R}_V$ the boundary edge of $\mathcal{S}$ that contains b's image. Similarly, for each internal vertex $v \in \mathcal{S}_V$, a combinatorial vertex map specifies the triangle of $\mathcal{R}$ that contains the preimage of v, and for each boundary vertex $c \in \mathcal{S}_V$, it determines the boundary edge of $\mathcal{R}$ that contains the preimage of c.

A vertex map f_0 and a combinatorial vertex map (g, h) are **consistent** if for any $u \in \mathcal{R}_V$, $f_0(u) \in g(u)$ and for any $v \in \mathcal{S}_V$, $f_0^{-1}(v) \in h(v)$.

The following lemma is immediately implied by the definition of combinatorial vertex maps.

Lemma 6.1.2. *There are $(m + n)^{O(m+n)}$ combinatorial vertex maps.*

Proof. For each vertex $v \in \mathcal{R}_V$, there are $O(n)$ possibilities for $g(v)$, and for each vertex $u \in \mathcal{S}_V$, there are $O(m)$ possibilities for $h(v)$. The total number of possible combinatorial vertex maps is $n^{O(m)} \times m^{O(n)}$, which is less than $(m + n)^{O(m+n)}$. □

6.2 Combinatorial Embeddings

If we ignore the exact location of vertices and edges, $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ can be interpreted as combinatorial embeddings of two triangulations. We call them combinatorial embeddings, and use the notation $\widetilde{\mathcal{R}}^c$ and $\widetilde{\mathcal{S}}^c$ to refer to them.

Definition 6.2.1. *Let $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ be a piecewise linear surface. For each vertex $v \in \widetilde{\mathcal{R}}_V$, let c_v be the cyclic order of edges incident on v. The **combinatorial embedding** $\widetilde{\mathcal{R}}^c = \{c_v | v \in \widetilde{\mathcal{R}}_V\}$ is the set of cyclic orders of all vertices.*

Definition 6.2.2. *Let $\widetilde{\mathcal{R}}^c$ be a combinatorial embedding, and $\widetilde{\mathcal{R}}$ be a piecewise linear surface. We say that $\widetilde{\mathcal{R}}^c$ and $\widetilde{\mathcal{R}}$ are consistent if for all $v \in \widetilde{\mathcal{R}}_V$, v has the cyclic order $c_v \in \widetilde{\mathcal{R}}^c$.*

6.3 Combinatorial Scaffold Maps and Normal Coordinates

Now, we are ready to define combinatorial descriptions of scaffold maps, which our algorithm uses to limit its search space to a finite set. We use a technique named **normal coordinates** to represent combinatorial scaffold maps.

Definition 6.3.1. *Let $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ be a scaffold map, and let $t = (s_1, s_2, s_3) \in \widetilde{\mathcal{S}}_T$, where $s_1, s_2, s_3 \in \widetilde{\mathcal{S}}_E$. The intersection of $f_1(\widetilde{\mathcal{R}}_E)$ with t is a collection of **elementary segments**: straight line segments with endpoints on $\partial(t)$. The intersection pattern of $f_1(\widetilde{\mathcal{R}}_E) \cap t$ can be represented (up to continuous deformation) by three numbers, one per edge. For each edge $s \in \mathcal{S}_E$ we define its **normal coordinate**, denoted by $N(s)$, as follows:*

(1) $N(s) = -1$ if $s \in f_1(\widetilde{\mathcal{R}}_E)$,

(2) or $N(s)$ is the number of elementary segments intersecting the interior of e.

Definition 6.3.2. *The **set of normal coordinates** of $f_1(\widetilde{\mathcal{R}}_E)$ is a vector of $|\widetilde{\mathcal{S}}_E| = O(m+n)$ numbers, one per edge in $\widetilde{\mathcal{S}}_E$. Each of these numbers is lower bounded by -1 and upper bounded by the crossing number of f_1, $\chi(f_1)$. If the normal coordinates are specified, there is a unique way of locating the elementary segments inside each $t \in$*

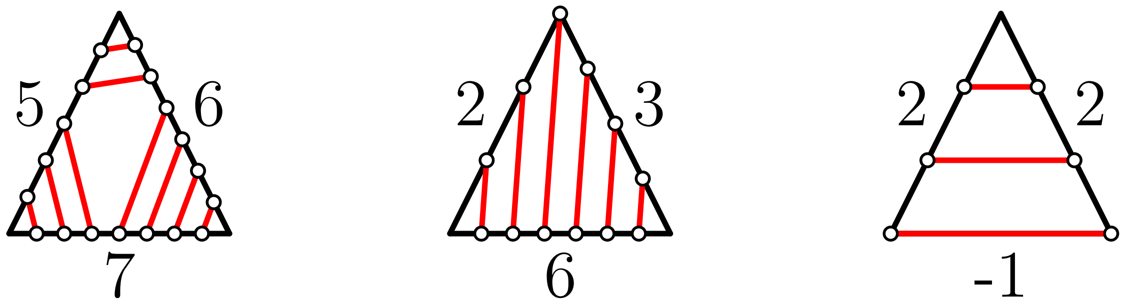


Figure 6.1: Examples for normal coordinates in triangles.

$\widetilde{\mathcal{S}}_T$ (up to a continuous deformation) so that they do not cross. Hence, the normal coordinates specify, for every $e \in \widetilde{\mathcal{R}}_E$, the crossing sequence of $f_1(e)$.

Definition 6.3.3. *A **combinatorial scaffold map** is a triple $\langle (g,h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle$ where (g,h) is a combinatorial vertex map, $(\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c)$ is a combinatorial embedding consistent with (g,h), and N is a set of normal coordinates consistent with $(\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c)$ specifying the crossing sequence for the image of every edge in $\widetilde{\mathcal{R}}_E$ in $\widetilde{\mathcal{S}}$.*

Definition 6.3.4. *A scaffold map $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ is **consistent** with a combinatorial scaffold map $\langle (g,h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle$ if*

(1) $f_0 = f_{1|\widetilde{\mathcal{R}}_V \cup f_1^{-1}(\widetilde{\mathcal{S}}_V)}$ is consistent with (g,h).

(2) f_0 is consistent with $(\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c)$.

(3) for every edge $e \in \widetilde{\mathcal{R}}_E$, the crossing sequence of $f_1(e)$ is consistent with the one implied by the normal coordinates N.

The following corollary immediately follows from Lemma 5.4.5.

Corollary 6.3.4.1. *For any pair of piecewise linear surfaces, $\mathcal{R}$ and $\mathcal{S}$, and any $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$, there is a list of combinatorial scaffold maps L of size $2^{O((m+n)^2)}$ that can be computed in $2^{O((m+n)^2)}$ time that has the following properties:*

(1) There exists a combinatorial scaffold map $\langle (g,h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle \in L$ that is consistent with a scaffold map of Fréchet length at most δ.

(2) For any $\langle (g,h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle \in L$, every normal coordinate of N is at most 2^{m+n}.

This corollary enables us to enumerate a set of combinatorial scaffold maps, one of which can be extended to an optimal scaffold map. Given an optimal combinatorial scaffold map, the problem becomes to find the exact location of the images of vertices and the crossing points. We consider two approaches: an approximation algorithm and an exact algorithm.

Chapter 7: Approximation Algorithm for Surfaces

In this chapter, we describe a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance between two triangulated surfaces $\mathcal{R}$ and $\mathcal{S}$ of diameter at most r.

In the previous section, we have shown that all possible optimal combinatorial scaffold maps are in size $2^{O((n+m)^2)}$. Our goal of this section is to show that any scaffold map is a good approximation if it is consistent with the optimal combinatorial scaffold map.

7.1 Relaxation of Scaffold Maps

If the maximum diameter of the triangles of the two surfaces is at most r, we show that if two scaffold maps are consistent with the same vertex map, refinement and normal coordinates, then the difference between their Fréchet lengths is no more than $2r$.

Lemma 7.1.1. *Let $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ be a scaffold map of Fréchet length δ', and let $e \in \widetilde{\mathcal{R}}_E$. Let $T \subseteq \widetilde{\mathcal{S}}_T$ be the set of all triangles that intersect $f_1(e)$. For any point $x \in e$ and any point $y \in t \in T$, we have $||x - y|| \leq \delta' + 2r$.*

Proof. Let $z \in f_1(e) \cap t$, and let $x' = f_1^{-1}(z)$. Because x and x' are both on e, we have $||x - x'|| \leq r$. Since y and z are in the same triangle t, we have $||z - y|| \leq r$. Therefore,

$$||x - y|| \leq ||x - x'|| + ||x' - z|| + ||z - y|| \leq r + \delta' + r \leq \delta' + 2r$$

□

Lemma 7.1.1 shows that for any scaffold map f_1, for any $e \in \widetilde{\mathcal{R}}_E$, the distance between e and all triangles $f_1(e)$ crosses is not greater than $\delta_F(f_1) + r$. Because the normal coordinates uniquely specify the sequence of triangles each $f(e)$ crosses, we get the following corollary immediately.

Corollary 7.1.1.1. *Let $f_1 : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ and $f_1' : \widetilde{\mathcal{R}}_E \to \widetilde{\mathcal{S}}$ be two scaffold maps with identical sets of normal coordinates N. For each $e \in \widetilde{\mathcal{R}}$, we have $\delta_F(e, f_1(e)) \leq \delta_F(e, f_1'(e)) + 2r$.*

Notice that f_1 and f_1' have the same refinement $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$. In other words, Corollary 7.1.1.1 shows that if two scaffold maps have common vertex map f_0, and combinatorial scaffold map S, then the difference of their Fréchet lengths will be no more than $2r$.

Corollary 7.1.1.2. *Let f_1 and f_1' be two scaffold maps consistent with the same vertex map f_0 and combinatorial scaffold map S. Then $\delta_F(f_1) \leq \delta_F(f_1') + 2r$.*

7.2 Relaxation of Vertex Maps

In the previous section, we show that scaffold maps have similar Fréchet lengths if they share the same vertex map and normal coordinates. In this section, we relax the common vertex map to common combinatorial vertex map. We show that scaffold maps have similar Fréchet lengths if they are consistent with the same combinatorial vertex map and normal coordinates. In other words, once the diameter of triangles is bounded, all scaffold maps with the same combinatorial specification will have similar Fréchet lengths.

First, we consider homeomorphisms from one single triangle to itself. Because the diameter is bounded, it implies the following lemma immediately.

Lemma 7.2.1. *Let t be a triangle with diameter r, and for any homeomorphism $f : t \to t$, $\delta_F(f) \leq r$.*

Proof. For any point $x \in t$, $|x - f(x)| \leq r$. Then $\delta_F(f) \leq r$. □

Corollary 7.2.1.1. *Let t be a triangle with diameter r, and for any two homeomorphisms $f : t \to t$ and $g : t \to t$, $|\delta_F(f) - \delta_F(g)| \leq r$.*

This lemma shows that all homeomorphisms for one triangle have bounded Fréchet length, and it implies that we can move the images of vertices with cost r.

Lemma 7.2.2. *Let t be a triangle with diameter r, and let $P, P' \subseteq int(t)$ be finite point sets with the same cardinality. Also, let $g : P \to P'$ be a bijection. There exists a homeomorphism $h : t \to t$ such that*

(1) $h|_{\partial(t)}$ *is the identity map.*

(2) $h|_P = g$.

(3) $\delta_F(h) \leq r$.

Proof. Let (x, y, z) be the vertices of t. Let $g' : \{x, y, z\} \cup P \to \{x, y, z\} \cup P'$ be a bijection that is the identity map for $\{x, y, z\}$ and g for P. Let H be a triangulation (a plane graph) with vertex set $\{x, y, z\} \cup P$. Let H' be a graph with vertex set $V' = \{x, y, z\} \cup P'$, where $v, v' \in V'$ are adjacent if and only if their corresponding vertices via g' are adjacent in H. The isomorphism between H and H' naturally gives rise to a combinatorial embedding of H' that is equivalent to the embedding of H. The isomorphism between H and H' and their equivalent embedding provides bijections between vertex sets, edge sets, and face sets of H and H'. Let h be any homeomorphism that respects these bijections and that is identity on the boundary. By the construction, h has properties (1) and (2). Additionally, $\delta_F(h) \leq r$, as h maps points within t, which has diameter r. □

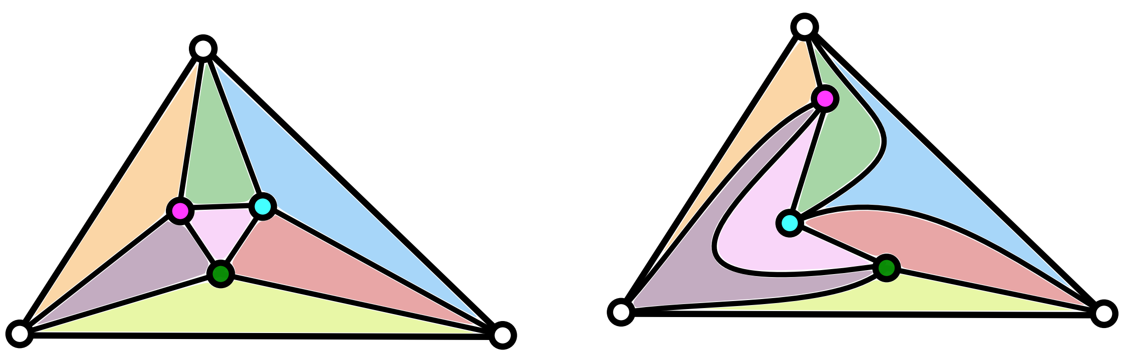


Figure 7.1: H and H'; corresponding faces and vertices have the same colors.

Combining Corollary 7.2.1.1 and Lemma 7.2.2, we show that for any homeomorphism $f : \mathcal{R} \to \mathcal{S}$, if we relax its vertex map to any vertex map with the same combinatorial vertex map (which may break the combinatorial embedding), then the Fréchet length will increase by at most $2r$.

Lemma 7.2.3. *Let $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length δ, and (g, h) be the combinatorial vertex map of f. Any vertex map f_0 that is consistent with (g, h) can be extended to an $(\mathcal{R}, \mathcal{S})$-homeomorphism of Fréchet length $\delta + 2r$.*

Proof. By Lemma 7.2.2, there exist two homeomorphisms of Fréchet length at most r, $h' : \mathcal{R} \to \mathcal{R}$ and $h'' : \mathcal{S} \to \mathcal{S}$, such that for any $v \in \mathcal{S}_V$, $h'(f_0^{-1}(v)) = f^{-1}(v)$, and for any $u \in \mathcal{R}_V$, $h''(f(u)) = f_0(u)$. Then $h'' \circ f \circ h'$ is an extension of f_0, and the Fréchet length is at most $\delta + 2r$. □

Lemma 7.2.3 implies that for any homeomorphism $h : \mathcal{R} \to \mathcal{R}$, for any triangle $t \in \mathcal{S}$ (or $s \in \mathcal{R}$), if we replace the mapping between $h^{-1}(t)$ and t (or s and $h(s)$) with any valid homeomorphism, then the Fréchet length will increase by at most $2r$.

Corollary 7.2.3.1. *Let $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length δ, and (g, h) be the combinatorial vertex map of f. For any vertex map f_0 consistent with (g, h), there exists a scaffold map f_1 and a combinatorial scaffold map $S = \langle (g, h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle$, such that f_1 is consistent with f_0 and S, and $\delta_F(f_1) \le \delta + 2r$.*

7.3 Summing Up

In the previous section, we showed that all scaffold maps with the same combinatorial scaffold map S have bounded Fréchet length. Using Corollary 7.2.3.1, we will show that there exists a combinatorial scaffold map S such that for any scaffold map f_1 consistent with S, $\delta_F(f_1) \le \delta + 4r$, where $\delta_F(f_1)$ is the Fréchet length of f_1.

Lemma 7.3.1. *There exists a combinatorial scaffold map S, such that for any scaffold map f_1 consistent with S, $\delta_F(f_1) \le \delta + 4r$.*

Proof. Let $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length δ, and (g, h) be the combinatorial vertex map of f. By Corollary 7.2.3.1, for any vertex map f_0 consistent with (g, h), there exists a scaffold map f_1 and a combinatorial scaffold map S, such that f_1 is consistent with f_0 and S, and $\delta_F(f_1) \le \delta + 2r$. And by Corollary 7.1.1.2, for any homeomorphism f_1' consistent with f_0 and S, $\delta_F(f_1') \le \delta_F(f_1) + 2r = \delta + 4r$. Hence, there exists a combinatorial scaffold map S, such that for any scaffold map f_1 consistent with S, $\delta_F(f_1) \le \delta + 4r$. □

This lemma only shows the existence of the combinatorial scaffold map. Notice that the images of vertices can have arbitrary locations in the triangle. Once we pick a vertex map, any refinement consistent with the vertex map will be fine. And the crossing points also can have arbitrary locations on each edge. Actually, in the algorithm, two things will decide whether the scaffold map is good approximation: one is the combinatorial vertex map, the other one is the normal coordinates. Any scaffold map that meets these two requirements must be a good scaffold map. Therefore, the proof also works for a stronger version.

Corollary 7.3.1.1. *Let $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length δ, and (g, h) be the combinatorial vertex map consistent with f. For any combinatorial embedding $(\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c)$, there exist normal coordinates N, such that for any scaffold map f_1 consistent with $\langle (g, h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle$, $\delta_F(f_1) \leq \delta + 4r$.*

Corollary 7.3.1.1 also suggests how our algorithm works. First, we search exhaustively for optimal combinatorial vertex map, and arbitrarily select a vertex map and a refinement. Then, we search exhaustively for an optimal normal coordinates N, and arbitrarily select a scaffold map.

Lemma 7.3.2. *Let $\mathcal{R}$ and $\mathcal{S}$ be two triangulated surfaces composed of triangles of diameter at most r. There exists a $2^{O((m+n)^2)}$ time algorithm to compute an $(\mathcal{R}, \mathcal{S})$-homeomorphism of Fréchet length at most $\delta_F(\mathcal{R}, \mathcal{S}) + 4r$, where m and n denote $|\mathcal{R}_V|$ and $|\mathcal{S}_V|$.*

Proof. By Lemma 7.3.1, there exists a combinatorial scaffold map S, such that for any scaffold map f_1 consistent with S, $\delta_F(f_1) \leq \delta + 4r$. To construct the scaffold map f_1, we will search exhaustively for the combinatorial scaffold map S. First, we enumerate all possible combinatorial vertex maps. By Lemma 6.1.2, there are $(m + n)^{O(m+n)}$ combinatorial vertex maps. For each combinatorial vertex map (g, h), we select an arbitrary vertex map f_0 that is consistent with (g, h), and select an arbitrary refinement. Then we enumerate all possible normal coordinates N. By Corollary 6.3.4.1, every coordinate of N is bounded by 2^{m+n}, so the number of possible normal coordinates is bounded by $2^{(m+n)^2}$. For each choice of normal coordinates N, for each $e \in \widetilde{\mathcal{R}}_E \cup \widetilde{\mathcal{S}}_E$, we select $N(e)$ arbitrary points on e as the crossing points. Then, we use the vertex map and the crossing points to construct a scaffold map f_1. By Corollary 7.3.1.1, $\delta_F(f_1) \leq \delta + 4r$. □

In the following sections, we study two cases, general surfaces and terrains. The algorithms for both are based on the algorithm above. We refine the surfaces until they are composed of triangles of bounded diameter, and apply our algorithm. For general surfaces, the running time depends on the area of the surfaces. For terrains, the running time depends on the maximum slope of the surfaces.

7.4 General surfaces

In this section, we describe an algorithm to compute the Fréchet distance between two arbitrary surfaces (of genus zero) $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{R}_T)$. To simplify our running time analysis, we assume that $\mathcal{R}$ and $\mathcal{S}$ are composed of fat triangles, that is, all their angles are larger than a constant $\theta > 0$. In general, our running time would depend on the minimum angle of the constituent triangles of the surfaces.

We define an ***r*-refinement** of a triangulation to be a refinement composed of triangles of diameter at most r. Before refining $\mathcal{R}$ and $\mathcal{S}$, we define triangulated grids, which we find helpful here and in the next section.

Triangulated grids. For any $w \in \mathbb{R}$, let $G_w = (V_w, E_w)$ be a triangulated grid of width w, see figure 7.2. That is

$$V_w = \{(iw, jw) \,|i, j \in \mathbb{Z}\},$$

and

$$E_w = \{((iw, jw), (i'w, j'w)) \,|i, j, i', j' \in \mathbb{Z} \wedge (i - i', j - j') \in \{(0, 1), (1, 0), (1, 1)\}\}.$$

Lemma 7.4.1. *Let $\mathcal{Q} = (\mathcal{Q}_V, \mathcal{Q}_E, \mathcal{Q}_T)$ be a triangulated surface composed of fat triangles, and let $r \in \mathbb{R}^+$. There exists an $O(|\mathcal{Q}_V| + Area(\mathcal{Q})/r^2)$ time algorithm to compute an r-refinement of $\mathcal{Q}$ of size $O(|\mathcal{Q}_V| + Area(\mathcal{Q})/r^2)$.*

Proof. For each triangle $t \in \mathcal{Q}$ with diameter larger than r, we show how to refine it

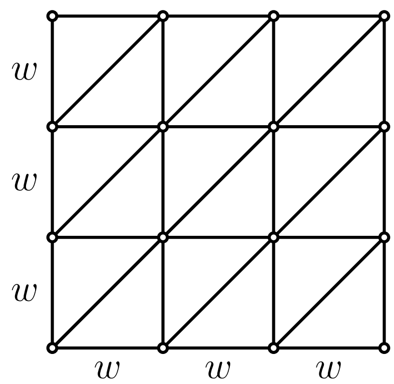


Figure 7.2: Triangulated grid of width w.

into a new triangulation composed of $O(\text{Area}(t)/r^2)$ triangles with diameter r. Note that when we put these triangulations together their vertices do not necessarily match on the border of different triangles. As a result, we may see flat polygons with more than three sides, but we can triangulate them without introducing new vertices.

Let $t \in \mathcal{Q}_T$ with side lengths ℓ, ℓ' and ℓ'', with $\max(\ell, \ell', \ell'') > r$. Place t on $G_{r/\sqrt{2}}$, the triangulated grid of width $r/\sqrt{2}$. Let $\bar{t}$ be the triangulation of t induced by $G_{r/\sqrt{2}}$. The number of triangles in $\bar{t}$ is $|\bar{t}| = O((\ell + \ell' + \ell'')/r + \text{Area}(t)/r^2)$. Since all angles of t are larger than a constant, ℓ, ℓ', and ℓ'' are within a constant factor of each other, and $\text{Area}(t) = \Theta(\ell^2)$. Therefore, $|\bar{t}| = O(\ell/r + \ell^2/r^2) = O(\ell^2/r^2) = O(\text{Area}(t)/r^2)$. $\square$

Our algorithm for general surfaces is implied by Lemma 7.3.2 and Lemma 7.4.1.

Theorem 7.4.2. *Let $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$ be two triangulated surfaces composed of fat triangles, let $n = |\mathcal{R}_V| + |\mathcal{S}_V|$, and let $\varepsilon > 0$. There exists a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance $\delta = \delta_F(\mathcal{R}, \mathcal{S})$ between $\mathcal{R}$ and $\mathcal{S}$ with running time*

$$2^{O\left(\left(|\mathcal{R}_V| + |\mathcal{S}_V| + \frac{\text{Area}(\mathcal{R}) + \text{Area}(\mathcal{S})}{(\varepsilon\delta)^2}\right)^2\right)}.$$

Proof. Let $\delta = \delta_F(\mathcal{R}, \mathcal{S})$, and let $r = (\varepsilon\delta)/4$. Let $\overline{\mathcal{R}}$ and $\overline{\mathcal{S}}$ be r-refinements of $\mathcal{R}$ and $\mathcal{S}$, respectively, obtained by applying the algorithm of Lemma 7.4.1, thus, $|\overline{\mathcal{R}}_V| = O(|\mathcal{R}_V| + \text{Area}(\mathcal{R})/r^2)$, and $|\overline{\mathcal{S}}_V| = O(|\mathcal{S}_V| + \text{Area}(\mathcal{S})/r^2)$. Also, $\overline{\mathcal{R}}$ and $\overline{\mathcal{S}}$ can be computed in linear time with respect to their sizes. Trivially, the number of vertices of $\overline{\mathcal{R}}_V \cup \overline{\mathcal{S}}_V$ in each ball of radius $\max(\delta, r)$ is at most $\overline{n} = |\overline{\mathcal{R}}_V| + |\overline{\mathcal{S}}_V|$. Thus, by Lemma 7.3.2 for $\overline{\mathcal{R}}$ and $\overline{\mathcal{S}}$, there is an $2^{O(\overline{n}^2)} \max(1, \log(\frac{1}{\delta\varepsilon}))$ time algorithm to compute an $(\mathcal{R}, \mathcal{S})$-homeomorphism of Fréchet length at most $\delta + 4r = (1+\varepsilon)\delta$. We have

$$2^{O(\overline{n}^2)} \max(1, \log(\frac{1}{\delta\varepsilon})) = 2^{O(|\overline{\mathcal{R}}_V|+|\overline{\mathcal{S}}_V|)^2} \max(1, \log(\frac{1}{\delta\varepsilon})) = 2^{O\left(\left(|\mathcal{R}_V|+|\mathcal{S}_V|+\frac{\text{Area}(\mathcal{R})+\text{Area}(\mathcal{S})}{(\varepsilon\delta)^2}\right)^2\right)}$$

□

7.5 Terrains

In this section, we describe an algorithm to compute the Fréchet distance between two polyhedral terrains $\mathcal{R}$ and $\mathcal{S}$ over $[0,1]^2$ (i.e. the images of the immersions $\varphi_{\mathcal{R}} : [0,1]^2 \to \mathbb{R}^3$ and $\varphi_{\mathcal{S}} : [0,1]^2 \to \mathbb{R}^3$ are polyhedral terrains over $[0,1]^2$). Let $\delta = \delta_F(\mathcal{R}, \mathcal{S})$, and let D be the maximum slope of $\mathcal{R}$ and $\mathcal{S}$ for any point in their domain, $[0,1]^2$.

7.5.1 Sampling

Let $\mathcal{Q} : [0,1]^2 \to \mathbb{R}$ be a polyhedral terrain, let $1/r \in \mathbb{Z}$, and let $G_r = (V_r, E_r)$ be a grid of width r. Here we use $\mathcal{Q}$ both to refer to the triangulated surface and to the function over $[0,1]^2$. The ***r*-coarse approximation** of $\mathcal{Q}$ is a polyhedral terrain $\overline{\mathcal{Q}}$, whose vertex set is

$$\overline{\mathcal{Q}}_V = \{(x, y, \mathcal{Q}(x,y)) \mid (x,y) \in V_r\},$$

and whose edge set is

$$\overline{\mathcal{Q}}_E = \{((x, y, \mathcal{Q}(x,y)), (x', y', \mathcal{Q}(x',y'))) \mid ((x,y),(x',y')) \in E_r\}.$$

Again, we view $\overline{\mathcal{Q}}$ as a triangulated surface as well as a function $\overline{Q} : [0,1]^2 \to \mathbb{R}$; thus, we use $\overline{\mathcal{Q}}(x,y)$ for a point $(x,y) \in [0,1]^2$.

Lemma 7.5.1. *Let $\mathcal{Q} : [0,1]^2 \to \mathbb{R}$ be a polyhedral terrain with maximum slope D, and let $\overline{\mathcal{Q}}$ be its r-coarse approximation, where $1/r \in \mathbb{Z}$. We have $\delta_F(\mathcal{Q}, \overline{\mathcal{Q}}) \leq 2\sqrt{2} \cdot r(D+$*

1).

Proof. Let $f : \mathcal{Q} \to \overline{\mathcal{Q}}$ be the projection map along the z-axis. That is, for any $(x, y, z) \in \mathcal{Q}$, $f(x, y, z) = (x, y, \overline{\mathcal{Q}}(x, y))$. Let t be the triangle in G_r that contains (x, y), and let (x', y') be any vertex of t. We have $||(x, y) - (x', y')|| \leq \sqrt{2} \cdot r$, which implies

$$||(x, y, \mathcal{Q}(x, y)) - (x', y', \mathcal{Q}(x', y'))|| \leq \sqrt{2} \cdot r(D + 1),$$

and

$$||(x, y, \overline{\mathcal{Q}}(x, y)) - (x', y', \overline{\mathcal{Q}}(x', y'))|| \leq \sqrt{2} \cdot r(D + 1),$$

for the maximum slope of $\overline{\mathcal{Q}}$ is bounded by D too.

On the other hand, since (x', y') is a grid point, $\overline{\mathcal{Q}}(x', y') = \mathcal{Q}(x', y')$. Thus, by the triangle inequality,

$$||(x, y, \mathcal{Q}) - (x, y, \overline{\mathcal{Q}})|| \leq 2\sqrt{2} \cdot r(D + 1).$$

□

Theorem 7.5.2. *Let $\mathcal{R}$ and $\mathcal{S}$ be polyhedral terrains over $[0, 1]^2$ of maximum slope D, and let $n = |\mathcal{R}_V| + |\mathcal{S}_V|$. There exists a $(1 + \varepsilon)$-approximation algorithm for computing the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ with running time*

$$O((D + 1)^2/(\varepsilon\delta)^2) \cdot n + 2^{O((D+1)^4/(\varepsilon^4\delta^2))}.$$

Proof. Let $r' = \min(\varepsilon\delta/(8 + 8\sqrt{2}), \varepsilon\delta/(8\sqrt{2}D))$, let $1/r$ be the smallest integer larger than $1/r'$, and let $\overline{\mathcal{R}}$ and $\overline{\mathcal{S}}$ be r-refinements of $\mathcal{R}$ and $\mathcal{S}$, respectively. Consider any point $p = (x, y, z) \in \mathbb{R}^3$. The number of vertices of $\overline{\mathcal{R}}_V \cup \overline{\mathcal{S}}_V$ in $Ball_{\max(\delta,r)}(p)$ is at most the number of grid points, vertices of V_r, in a disk of radius $\max(\delta, r)$ with center (x, y), which is $O(1 + \delta^2/r^2)$. Thus, Lemma 7.3.2 implies that an $(\overline{\mathcal{R}}, \overline{\mathcal{S}})$-homeomorphism of Fréchet length $\delta + 4r$ can be computed in $2^{O(\delta^2/r^4)} = 2^{O((D+1)^4/(\varepsilon^4\delta^2))}$ time. Composing this homeomorphism with the $(\mathcal{R}, \overline{\mathcal{R}})$-homeomorphism and the $(\overline{\mathcal{S}}, \mathcal{S})$-homeomorphism of Lemma 7.5.1, we obtain a homeomorphism of Fréchet length

$$\delta_F(\mathcal{R}, \mathcal{S}) + 4r + 4\sqrt{2} \cdot r(D + 1) \leq \delta + \varepsilon\delta/2 + \varepsilon\delta/2 = (1 + \varepsilon)\delta.$$

We need to sample $O((D+1)^2/(\varepsilon\delta)^2)$ points from $\mathcal{R}$ and $\mathcal{S}$ to compute $\overline{\mathcal{R}}$ and $\overline{\mathcal{S}}$, which takes $O((D+1)^2/(\varepsilon\delta)^2 n)$ time. Therefore, overall, we obtain the desired asymptotic time bound. $\square$

Chapter 8: Exact Algorithm for Surfaces

In this chapter, we study the decision problem for the Fréchet distance between two piecewise linear surfaces: decide whether the Fréchet distance between two piecewise linear surfaces is less than a query value. We describe an exact algorithm to decide the Fréchet distance between two piecewise linear surfaces. We also show that the Fréchet distance between a piecewise linear surface and a triangle can be decided in PSPACE.

In the previous section, we build a finite-size list of combinatorial scaffold maps that contains at least one optimal combinatorial scaffold map. To obtain the approximation algorithm, we show that for any scaffold map consistent with the optimal combinatorial scaffold map, its Fréchet length is at most $\delta_F(\mathcal{R}, \mathcal{S}) + r$, where r is the maximum length of edges of $\mathcal{R}$ and $\mathcal{S}$. To obtain our exact algorithm, for a given combinatorial scaffold map and δ, we want to check whether the combinatorial scaffold map can be extended to a scaffold map with Fréchet length at most δ. We observe that computing the Fréchet length of a scaffold map can be reduced to computing the maximum distances between (1) vertices and their images, (2) crossing points and their corresponding images (crossing points) on the other surface. For a given combinatorial scaffold map, if we are able to compute the optimal locations for images of vertices and crossing points, minimizing the maximum distance between them, then we can compute an optimal scaffold map consistent with the given combinatorial scaffold map. This observation gives us an idea to build a system of inequalities for the given combinatorial scaffold map and δ. If and only if there is a feasible solution for the system, the given combinatorial scaffold map can be extended to a scaffold map of Fréchet length at most δ.

8.1 System of Polynomial Inequalities

Recall our decision problem: given two surfaces of genus zero $\mathcal{R}$ and $\mathcal{S}$, and $\delta \geq 0$, we want to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$.

The algorithm of Corollary 6.3.4.1 builds a list of combinatorial scaffold maps L that is guaranteed to contain one that extends to a scaffold map of Fréchet length at most δ, if and only if $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$. To decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$, we check every element of L to see if such an extendable map exists. Therefore, we need an algorithm to decide if a combinatorial scaffold map is extendable. To that end, for a given combinatorial scaffold map $S = \langle (g, h), (\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c), N \rangle \in L$, we show how to build a system of inequalities that is feasible if and only if S extends to a scaffold map of Fréchet length at most δ.

As we mentioned, this system tries to compute the exact locations of (i) images of vertices and (ii) crossing points on edges, and these locations are able to extend to a feasible scaffold map. So, there are two types of variables in our system: vertex variables, which represent the locations of images of vertices, and crossing point variables, which represent the crossing points on edges.

Besides these variables, we have three types of constraints to ensure that the scaffold map is consistent with S. (1) For each vertex $u \in \mathcal{R}_V$, let u_v be the variable representing its image. We have constraints on u_v to ensure that the image of u is contained by the triangle $g(v)$. (2) For the crossing points on each edge, the constraints ensure their order on the edge, and they will ensure the images of edges do not cross each other. (3) Constraints ensure that the refinement is valid. If we do not have the third type of constraints, the system will put the image of each vertex on the closest point to that vertex in its corresponding triangle. In other words, these constraints ensure that the images of vertices stay on the correct side of the images of edges corresponding to the given combinatorial scaffold map. Finally, we have constraints to ensure that the Fréchet length of the scaffold map is at most δ. They are inequalities on distance formulas to ensure that the distances between vertices, crossing points and their images are at most δ.

During the rest of this section, for each triangle t fix an ordering of its vertices, and for each edge e fix an ordering of its endpoints. Hence, we can unambiguously denote a triangle by an ordered triple of vertices, such as (a, b, c), and unambiguously denote an edge by an ordered pair of vertices, such as (a, b).

8.2 Vertex Variables

For each vertex $u \in \mathcal{R}_V$, we specify $f_0(u)$ by two variables α_u, and β_u, in addition to $g(u)$. Specifically, if $g(u)$ is a triangle (a, b, c) (with $a, b, c \in \mathcal{S}_V$) then $f_0(u) = a + \alpha_u \overrightarrow{ab} + \beta_u \overrightarrow{ac}$. We can guarantee that $f_0(u) \in g(u)$ by enforcing $\alpha_u \geq 0$, $\beta_u \geq 0$ and $\alpha_u + \beta_u \leq 1$. Additionally, the distance between u and $f_0(u)$ must be at most δ. Hence, for each $u \in \mathcal{R}_V$ and $g(u) = (a, b, c) \in \mathcal{S}_T$, we have the following constraints:

$$\begin{aligned} &\alpha_u, \beta_u \geq 0,\ \alpha_u + \beta_u \leq 1 \\ &||u - f_0(u)|| = ||u - a - \alpha_u \overrightarrow{ab} - \beta_u \overrightarrow{ac}|| \leq \delta \end{aligned} \tag{8.1}$$

Similarly, for each $v \in \mathcal{S}_V$, we specify $f_0^{-1}(v)$ with two variables α'_v and β'_v, in addition to $h(v)$. By a similar argument, for each $v \in \mathcal{S}_V$ and $h(v) = (a', b', c') \in \mathcal{R}_T$, we have the following constraints:

$$\begin{aligned} &\alpha'_v, \beta'_v \geq 0,\ \alpha'_v + \beta'_v \leq 1 \\ &||v - f_0^{-1}(v)|| = ||v - a' - \alpha'_v \overrightarrow{a'b'} - \beta'_v \overrightarrow{a'c'}|| \leq \delta \end{aligned} \tag{8.2}$$

In each case, the first three constraints are linear and the last one is of degree p under the ℓ_p norm; in particular, it is linear for the ℓ_1 norm and quadratic for the ℓ_2 norm.

8.3 Crossing Point Variables

Let $e \in \widetilde{\mathcal{R}}_E$, let $s \in \widetilde{\mathcal{S}}_E$, let x be the crossing point between e and $f_1^{-1}(s)$, and let y be the corresponding crossing point between $f_1(e)$ and s. Note that $f_1(x) = y$. Now, let $e = (a_e, b_e)$ and let $s = (a_s, b_s)$. We specify (the location of) x by one variable α_x, so $x = a_e + \alpha_x \cdot \overrightarrow{a_e b_e}$. We can guarantee that $f_1(x) \in (a, b)$ by enforcing $0 \leq \alpha_x \leq 1$. Similarly, we specify $y = a_s + \alpha_y \cdot \overrightarrow{a_s b_s}$, and enforce $0 \leq \alpha_y \leq 1$. Finally, as $y = f_1(x)$, the x-to-y distance must be at most δ. In summary, for a pair of corresponding crossing points $x \in (a_e, b_e) \in \mathcal{R}_E$, and $y \in (a_s, b_s) \in \mathcal{S}_E$, we obtain the following set of constraints:

$$0 \leq \alpha_x, \alpha_y \leq 1 \tag{8.3}$$
$$||x - y|| = ||(a_e + \alpha_x \cdot \overrightarrow{a_e b_e}) - (a_s + \alpha_y \cdot \overrightarrow{a_s b_s})|| \leq \delta$$

The first two constraints are linear, and the third constraint is of degree $2p$ for the ℓ_p norm. Note a_e, b_e, a_s, and b_s may be described by variables themselves, so the constraint of $||x - y||$ is of degree $2p$, rather than p. In particular the last equation is at most quadratic for the ℓ_1 norm and at most quartic for the ℓ_2 norm.

We also need to guarantee that the sequence of crossing points on an edge is as specified by f_1. Let $e \in \widetilde{\mathcal{R}}_E$, and let $x_1, x_2, \ldots, x_k$ be the set of crossing points on $s = (a, b)$ in the order deduced from the normal coordinates N. Also, for each $1 \leq i \leq k$, let $x_i = a + \alpha_i \cdot \overrightarrow{ab}$, as specified above for crossing points. To ensure that the images of the edges of $\widetilde{\mathcal{R}}_E$ under f_1 do not cross, it is sufficient to force the x_i's to appear in order on s; that is

$$\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_k. \tag{8.4}$$

Similarly, for each $e = (c, d) \in \widetilde{\mathcal{S}}_E$, if $y_1, y_2, \ldots, y_\ell$ is the sequence of crossing points on e deduced from N, and $y_i = c + \beta_i \cdot \overrightarrow{cd}$ (for each $1 \leq i \leq \ell$), the following condition must hold:

$$\beta_1 \leq \beta_2 \leq \ldots \leq \beta_\ell. \tag{8.5}$$

All the constraints in this last category are linear.

8.4 Valid Refinements

Let $\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c$ be combinatorial embeddings, and f_0 be a vertex map. We say that $(\widetilde{\mathcal{R}}^c, \widetilde{\mathcal{S}}^c)$ and f_0 are **consistent** if they give a valid geometric triangulation of every triangle in $\mathcal{R}_T$ and $\mathcal{S}_T$. Specifically, let $t = (a, b, c) \in \mathcal{S}_T$ (a, b, c are in counterclockwise order),

$V = \{f_0(u) \mid u \in \mathcal{R}_V, f_0(u) \in t\}$. Let $v \in V \cup \{a, b, c\}$, and let $v'_0, v'_1, \ldots, v'_{k-1}$ be the neighbors of v inside t in counterclockwise order according to $\widetilde{\mathcal{S}}^c$. We should have:

(1) For any $0 \leq i \leq k-1$, we have $0 \leq \angle v'_i v v'_{i+1} \leq \pi$.

(2) In addition:

  (a) $\sum_{i=0}^{k-1} \angle v'_i v v'_{i+1} = 2\pi$, if $v \notin \{a, b, c\}$.

  (b) $\sum_{i=0}^{k-1} \angle v'_i a v'_{i+1} = \angle bac$, $\sum_{i=0}^{k-1} \angle v'_i b v'_{i+1} = \angle cba$, and $\sum_{i=0}^{k-1} \angle v'_i c v'_{i+1} = \angle acb$.

Additionally, the analogous set of conditions must hold for each triangle of $\mathcal{R}$.

We check the consistency of the combinatorial embeddings $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ and the vertex map f_0. Conditions (1) and (2) of consistency must be verified for each triangle of $\mathcal{S}_T$ and each triangle of $\mathcal{R}_T$. Here we explain how to build a system of inequalities to verify these conditions for a triangle $t \in \mathcal{S}_T$; the other case is symmetric. Let $t = (a, b, c) \in \mathcal{S}_T$ and let V be the set of vertices of $\widetilde{\mathcal{S}}_V$ that are in t. Let $v \in (a, b, c) \cup V$ and let $V' = \{v'_0, v'_1, \ldots, v'_{k-1}\} \subseteq \{a, b, c\} \cup V$ be the set of its neighbors in cyclic (counterclockwise) order according to $\widetilde{\mathcal{S}}$. Condition (1) is equivalent to the following constraint:

$$(v'_i - v) \times (v'_{i+1} - v) \geq 0,$$

where $\times$ is the cross product. Recall that the location of each vertex in the image of $\mathcal{R}_V$ under f_0 is specified by two variables. For any $0 \leq i \leq k-1$, let

$$v'_i = a + \alpha'_i \cdot \overrightarrow{ab} + \beta'_i \cdot \overrightarrow{ac}.$$

Also, let

$$v = a + \alpha \cdot \overrightarrow{ab} + \beta \cdot \overrightarrow{ac}.$$

Therefore, the cross product can be written as follows.

$$(v'_i - v) \times (v'_{i+1} - v) = \big((\alpha'_i - \alpha)(\beta'_{i+1} - \beta) - (\beta'_i - \beta)(\alpha'_{i+1} - \alpha)\big) \cdot (\overrightarrow{ab} \times \overrightarrow{ac})$$

Therefore, since $\overrightarrow{ab} \times \overrightarrow{ac} > 0$, then

$$(v_i' - v) \times (v_{i+1}' - v) \geq 0 \Leftrightarrow \\ (\alpha_i' - \alpha)(\beta_{i+1}' - \beta) - (\beta_i' - \beta)(\alpha_{i+1}' - \alpha) \geq 0 \tag{8.6}$$

We add a quadratic constraint to our system of constraints. Note that if any of v, v_i', or v_{i+1}' corresponds to a, b, or c, the constraint becomes simpler because these vertices have fixed locations, so fewer variables will be involved in our constraints.

Next, we show that if Condition (1) holds for every vertex of $\{a, b, c\} \cup V$, then Condition (2) must hold for all these vertices. Hence, we do not need to check Condition (2) explicitly.

Lemma 8.4.1. *If Condition (1) is satisfied, then*

(1) Condition (2-b) is satisfied, and

(2) for any $v \notin \{a, b, c\}$, there is a $k \in \mathbb{N}$ such that $\sum_{i=0}^{k-1} \angle v_i' v v_{i+1}' = 2k\pi$.

Proof. First, we show Condition (2-b) for vertex a; the arguments for vertices b and c are symmetric. Let $v_0', v_2', \ldots, v_{k-1}'$ be the set of a's neighbors in counterclockwise order. In particular, we have, $v_0' = b$ and $v_{k-1}' = c$. Therefore, the set of angles $\{\angle v_i' a v_{i+1}' \mid 0 \leq i \leq k-2\}$ covers the angle $\angle bac$. Also, for every $0 \leq i \leq k-2$, we have $\angle v_i' a v_{i+1}' \geq 0$ by Condition (1). Therefore, for any $0 \leq i \leq k-3$, $\angle v_i' a v_{i+1}'$ and $\angle v_{i+1}' a v_{i+2}'$ are internally disjoint. Hence, we conclude $\sum_{i=0}^{k-1} \angle v_i' a v_{i+1}' = \angle bac$.

Next, consider an internal $v \notin \{a, b, c\}$. Note that v and V' are in the same plane, and $\angle v_i' v v_{i+1}' \geq 0$ by Condition (1). Therefore, $v_0', v_1', \ldots, v_{k-1}', v_0'$ give a traversal around v that always goes counterclockwise, and starts and ends at the same vertex. We conclude that $\sum_{i=0}^{k-1} \angle v_i' v v_{i+1}'$ is a positive multiple of 2π. □

Lemma 8.4.2. *Condition (1) implies Condition (2).*

Proof. By Lemma 8.4.1, Condition (1) implies Condition (2-b). It remains to show that condition (1) implies condition (2-a).

To that end, let $\tau = (\tau_V, \tau_E, \tau_T)$ be the combinatorial embedding of t according to $\widetilde{\mathcal{S}}$. Further, let $\widetilde{t}$ be the piecewise linear surface that is obtained by identifying sides of geometric triangles according to τ. Therefore, for each combinatorial triangle $(z_1, z_2, z_3) \in \tau_T$ we have a geometric triangle with side lengths $||\overline{z_1} - \overline{z_2}||$, $||\overline{z_1} - \overline{z_3}||$, and $||\overline{z_2} - \overline{z_3}||$, where $\overline{z}$ is used to denote the location of the combinatorial vertex z determined by f_0 if z is internal, or as part of the input of z is a vertex of t. The (dual of the) combinatorial description τ specifies how to identify the sides of these triangles to obtain $\widetilde{\tau}$.

For the topological disk $\widetilde{\tau}$, a discrete form of the Gauss-Bonnet theorem implies $\sum_{v\in\tau_V} \angle v = 2\pi(|\tau_V| - 3) + \pi$, where $\angle v$ denotes the total angle around v on the surface $\widetilde{\tau}$. We include a short proof based on Euler's formula.

Since $\widetilde{\tau}$ is a topological disk, by the Euler formula we have: $|\tau_V| - |\tau_E| + |\tau_T| = 1$. Additionally, as τ has exactly three boundary edges, we have $|\tau_E| = \frac{3(|\tau_T|+1)}{2}$. It follows that

$$
\begin{aligned}
&|\tau_V| - \frac{3(|\tau_T| + 1)}{2} + |\tau_T| = 1 \quad \Rightarrow \\
&|\tau_V| - \frac{|\tau_T|}{2} = \frac{5}{2} \quad \Rightarrow \\
&2|\tau_V| - |\tau_T| = 5 \quad \Rightarrow \\
&|\tau_T| = 2|\tau_V| - 5.
\end{aligned}
$$

Consequently,

$$\sum_{v\in\tau_V} \angle v = \pi \cdot |\tau_T| = 2\pi|\tau_V| - 5\pi = 2\pi(|\tau_V| - 3) + \pi.$$

Because $\angle a + \angle b + \angle c = \pi$, therefore,

$$\sum_{v\in\tau_V\setminus\{a,b,c\}} \angle v_i = 2\pi(|\tau_V| - 3).$$

But, since condition (1) holds, we have $\angle v_i \geq 2\pi$ for all $v \in \tau_V\setminus\{a, b, c\}$. It follows that $\angle v_i$ must be exactly 2π, for all $v \in \tau_V\setminus\{a, b, c\}$, and the proof is complete. $\square$

8.5 Summing Up

Now, we show that given a combinatorial scaffold map we can check whether a low-cost scaffold map with that combinatorial scaffold map exists.

Lemma 8.5.1. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces with m and n vertices, respectively, and let $\delta > 0$. Also, let $S = \langle (g, h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), N \rangle$ be a combinatorial scaffold map, such that the value of every coordinate of N is at most 2^{m+n}. In $2^{O(m+n)}$ time, a system of polynomial constraints of size $2^{O(m+n)}$ can be computed that is feasible if and only if S extends to a scaffold map of Fréchet length at most δ.*

Proof. Our algorithm builds variables and constraints for vertices, crossing points, refinements, and non-crossing images in order as detailed above. First, it builds vertex variables and constraints from (g, h) in $O(m + n)$ time (all constraints of type (8.1) and (8.2). Next, it expands the normal coordinates to compute the crossing sequence of the image of each $e \in \widetilde{\mathcal{R}}_E$. This crossing sequence identifies all the crossing points on e together with their pairs on $\widetilde{\mathcal{S}}$. As the maximum coordinate is at most 2^{m+n}, the crossing sequence of e's image can be computed by just tracing it based on the normal coordinates in $2^{O(m+n)}$ time. Since, there are $O(m+n)$ edges in $\widetilde{\mathcal{R}}_E$, the crossing sequence of all of them can be computed in $2^{O(m+n)}$ time. After computing all crossing sequences we introduce $2^{O(m+n)}$ constraints of type (8.3). Additionally, for each edge we include constraints of type (8.4) or (8.5) to ensure the images of edges do not cross. Finally, to ensure that the refinements are consistent with all feasible solutions of our system, we introduce constraints of type (8.6) for each vertex and its neighbors. Any feasible solution of our system can be extended to a scaffold map of Fréchet length at most δ by interpolating the map between consecutive crossing points. On the other hand, if a scaffold map f_1 of Fréchet length at most δ that is consistent with S exists, then our system has a feasible solution. $\square$

The main theorem of this section is that the Fréchet distance between two surfaces is decidable. Our result follows from Corollary 6.3.4.1, Lemma 8.5.1, and Lemma 3.0.1.

Theorem 8.5.2. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces with m and n vertices, respectively, and let $\delta \geq 0$. There is an algorithm to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$.*

Proof. By Corollary 6.3.4.1, a list L of $2^{O((m+n)^2)}$ combinatorial scaffold maps can be built in $2^{O((m+n)^2)}$ time so that at least one of them extends to a scaffold map of Fréchet length δ if and only if $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$. By Lemma 8.5.1, for any $S \in L$ a system M_S of $2^{O(m+n)}$ polynomial inequalities can be built in $2^{O(m+n)}$ time such that M_S is feasible if and only if S extends to a scaffold map of Fréchet length δ. Finally, by Lemma 3.0.1 the feasibility of M_S can be checked in $2^{O(m+n)}$ space. $\square$

Chapter 9: A Surface and a Triangle: PSPACE

In this chapter, we show that the special case of deciding the Fréchet distance between a surface and a triangle is in PSPACE. This special case has been proved to be NP-hard by Godau [16].

Let $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ be a piecewise linear surface, and let $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$ be a triangle, both immersed into $\mathbb{R}^3$ (by immersions $\varphi_{\mathcal{R}} : \mathcal{R} \to \mathbb{R}^3$ and $\varphi_{\mathcal{S}} : \mathcal{S} \to \mathbb{R}^3$). In particular, $|\mathcal{S}_V| = |\mathcal{S}_E| = 3$, and $|\mathcal{S}_T| = 1$. Also, let $m = |\mathcal{R}_V|$, and let $\delta \geq 0$. We describe a PSPACE algorithm to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$.

9.1 Tight Images

We introduce tight scaffold maps and detailed normal coordinates. We show that enumerating over tight scaffold maps is sufficient when deciding the Fréchet distance between a surface and a triangle. We use detailed normal coordinates to enumerate combinatorial descriptions of tight scaffold maps.

9.1.1 Tight Edge Images

We show that for any $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$ there is a scaffold map f_1 of Fréchet length at most δ that maps each edge $e \in \mathcal{R}_E$ into a homotopic shortest path in $\mathcal{S} \backslash f_1(\mathcal{R}_V)$. This property enables us to reduce the number of constraints for crossing points to a polynomial, and hence facilitates our PSPACE result. We discuss two auxiliary lemmas before stating our main lemma.

The following lemma is implicit in the work of Colin de Verdière and Erickson [13], and it follows from Hass and Scott [19].

Lemma 9.1.1 ([13, 19])**.** *Let γ_1 and γ_2 be two non-crossing paths on a surface of genus zero with boundary components, and let γ_1', and γ_2' be the shortest homotopic paths in the homotopy classes of γ_1 and γ_2, respectively. Then, γ_1' does not cross γ_2'.*

Buchin et al. [8] observe that shortcutting a curve along a line segment cannot increase its Fréchet distance to a line segment. Hass and Scott [19] show that if a curve γ on a surface of genus zero with boundary components is not the shortest path in its homotopy class, then there is an empty bigon whose one side is a subpath of γ and whose other side is a global shortest path. Exploiting this property, Nayyeri and Sidiropoulos [22] show that each curve in a planar domain can be modified to its homotopic shortest path via a finite sequence of shortcuttings along line segments. Taking the observation of Buchin et al. into account they conclude the following lemma.

Lemma 9.1.2 ([22], Corollary 3.8)**.** *Let $t \subseteq \mathbb{R}^3$ be a triangle with point punctures, let $\gamma \in t$ be a path, let γ' be the shortest path homotopic to γ, and let $s \in \mathbb{R}^3$ be a line segment. Then, $\delta_F(\gamma', s) \leq \delta_F(\gamma, s)$.*

Now, we are ready to prove the following lemma.

Lemma 9.1.3. *Let $\mathcal{R}$ be a piecewise linear surface and let $\mathcal{S}$ be a triangle. Let f_0 be a vertex map between them, and let $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ be refinements that are consistent with f_0. Finally, let f_1 be a scaffold map over $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$. There exists a scaffold map f_1' over $\widetilde{\mathcal{R}}$ and $\widetilde{\mathcal{S}}$ with the following properties:*

(1) The maps f_1 and f_1' have the same set of normal coordinates.

(2) For any $e \in \mathcal{R}_E$, $f_1'(e)$ is the shortest homotopic path in $\mathcal{S}\backslash\widetilde{\mathcal{S}}_V$.

(3) $\delta_F(f_1') \leq \delta_F(f_1)$.

Proof. We obtain f_1' by iteratively modifying f_1 as follows. For each $e \in \mathcal{R}_E$, we replace $\gamma = f_1(e)$ with the shortest path γ' in γ's homotopy class in $\mathcal{S}\backslash\widetilde{\mathcal{S}}_V$. We modify f_1 such that (i) $f_1(e) = \gamma'$, and (ii) $\delta_F(f_{1|e}) = \delta_F(e, \gamma')$.

Lemma 9.1.2 implies for every e, $\delta_F(f'_{1|e}) \leq \delta_F(f_{1|e})$, hence, $\delta_F(f_1') \leq \delta_F(f_1)$. Lemma 9.1.1 implies that for each $e, e' \in \mathcal{R}_E$ their images are still non-crossing. In particular, for each vertex $u \in \mathcal{R}_V$ the cyclic orders of edges around $f_1(u)$ and $f_1'(u)$ are the same. Hence, f_1' is a valid scaffold map. Therefore, properties (2) and (3) hold.

Additionally, for each $e \in \mathcal{R}_E$, $\gamma = f_1(e)$ and its homotopic shortest path γ' are homotopic. Therefore, f_1 and f_1' must have the same set of normal coordinates. $\square$

9.1.2 Tight Scaffold Maps

Let f_0 be a vertex map, $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ and $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$ be refinements consistent with f_0, and f_1 a scaffold map consistent with f_0, $\widetilde{\mathcal{R}}$, and $\widetilde{\mathcal{S}}$. We say that f_1 is **tight** over $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ if it has the following two properties.

(1) For each $e \in \widetilde{\mathcal{R}}_E$, $f_1(e)$ is the shortest homotopic path in $\mathcal{S} \backslash \mathcal{S}_V$.

(2) For each $e \in \widetilde{\mathcal{R}}_E$, $f_1(e)$ is composed of edges of $\widetilde{\mathcal{S}}_E$.

Lemma 9.1.4. *For any $\delta \geq 0$, the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ is at most δ if and only if there are refinements $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ and $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$ and a tight scaffold map of Fréchet length at most δ over $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ that has crossing number at most $2m2^m$.*

Proof. By Lemma 9.1.3, there are refinements $\widetilde{\mathcal{R}}' = (\widetilde{\mathcal{R}}'_V, \widetilde{\mathcal{R}}'_E, \widetilde{\mathcal{R}}'_T)$ and $\widetilde{\mathcal{S}}' = (\widetilde{\mathcal{S}}'_V, \widetilde{\mathcal{S}}'_E, \widetilde{\mathcal{S}}'_T)$, and a scaffold map f_1 such that (1) for each $e \in \widetilde{\mathcal{R}}'_E$, $f_1(e)$ is a homotopic shortest path, (2) the normal coordinates of f_1 are bounded by 2^m, and (3) $\delta_F(f_1) \leq \delta$. Therefore, Condition (1) of tight scaffold maps already holds. We modify $\widetilde{\mathcal{S}}'$ to satisfy Condition (2).

Since Condition (1) holds, for each $e \in \widetilde{\mathcal{R}}'_E$, $f_1(e)$ is a sequence of segments $(f_1(x), f_1(x'))$, where $x, x' \in \widetilde{\mathcal{R}}'_V$. Let T be the set of all segments $s = (f_1(x), f_1(x'))$ such that $s \in f_1(e)$ for at least one $e \in \widetilde{\mathcal{R}}'_E$. Since the images of edges of $\widetilde{\mathcal{R}}'_E$ are non-crossing, T is a noncrossing set of segments over $\mathcal{S}'_V$. Complete it to a triangulation $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}'_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$ of $\mathcal{S}$ by adding more segments. By the construction of $\widetilde{\mathcal{S}}$, Condition (2) holds for f_1 and $\widetilde{\mathcal{S}}_E$. It remains to bound the crossing number of each segment in $\widetilde{\mathcal{S}}_E$.

Let $s \in \widetilde{\mathcal{S}}_E$, and let n_s be the number of times (with multiplicity) that the images of edges of $\widetilde{\mathcal{R}}'_E$ use s. We show that $n_s \leq 2^m$. If $s \in \widetilde{\mathcal{S}}'_E$ then the statement follows from the bound on the crossing number according to $\widetilde{\mathcal{S}}'$. Otherwise, s crosses at least one edge $\ell \in \widetilde{\mathcal{S}}'_E$, therefore, $n_S \leq \chi_{\widetilde{\mathcal{S}}'}(\ell) \leq 2^m$, as any traversal of s crosses ℓ.

Now, let $s' = (y, y') \in \widetilde{\mathcal{S}}_E$, and note that any subpath that crosses s' must use an edge that is adjacent to y or y'. But, there are at most $2m$ such edges, and each one is used at most 2^m times by the images of $\widetilde{\mathcal{R}}'_E$ as proved above. $\square$

9.1.3 Detailed Normal Coordinates

Because of Lemma 9.1.4, we can assume that crossings between images of $\widetilde{\mathcal{R}}_E$ and any edge $s \in \widetilde{\mathcal{S}}_E$ happen only at endpoints of s. We call these endpoints **portals**. For each $e \in \widetilde{\mathcal{R}}_E$, in addition to its edge crossing sequence, we define its **portal crossing sequence**, which is the sequence of portals in order that $f_1(e)$ crosses. We refine the normal coordinates to include two numbers for each edge $s \in \widetilde{\mathcal{S}}_E$, $N_1(s)$ and $N_2(s)$: the number of crossings in each endpoint. If $N(s) = -1$ then we set $N_1(s) = N_2(s) = -1$. Otherwise, for each edge $(a, b) \in \widetilde{\mathcal{S}}_E$, $N_1(s)$ and $N_2(s)$ are the number of crossings of s at a and b, respectively. The set of **detailed normal coordinates** of $f_1(\widetilde{\mathcal{R}}_E)$ is composed of two vectors N_1 and N_2 each with $|\widetilde{\mathcal{S}}_E|$ numbers, one per edge in $\widetilde{\mathcal{S}}_E$. Each of these numbers is lower bounded by -1 and upper bounded by the crossing number of f_1, $\chi(f_1)$. Provided the normal coordinates, there is a unique way of locating the elementary segments inside each $t \in \widetilde{\mathcal{S}}_T$. Note that many of these segments may overlap, but no pair of them crosses.

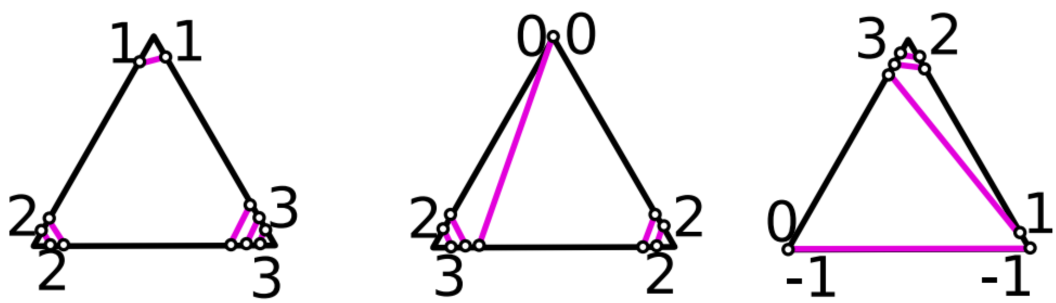


Figure 9.1: Detailed normal coordinates; note in reality the segments intersect each edge only at its endpoints; the figures are slightly modified for demonstration.

A **combinatorial scaffold map with detailed coordinates** is a triple

$$\langle (g, h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), (N_1, N_2) \rangle$$

where (g, h) is a combinatorial vertex map, $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ is a combinatorial embedding over (g, h), and (N_1, N_2) is a set of detailed normal coordinates over $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ specifying the crossing sequence of edges and portals for the image of every edge in $\widetilde{\mathcal{R}}_E$. A combinatorial scaffold map $\langle(g, h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), (N_1, N_2)\rangle$ **extends** a scaffold map f_1 if (i) $f_0 = f_{1|\widetilde{\mathcal{R}}_V \cup f_1^{-1}(\widetilde{\mathcal{S}}_V)}$ is consistent with (g, h), (ii) f_0 is consistent with $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$, and (iii), for every edge $e \in \widetilde{\mathcal{R}}_E$, the portal crossing sequence of $f_1(e)$ is the same as the one implied by the detailed normal coordinates (N_1, N_2). Also, if a vertex map f_0 consistent with (g, h) is given, then we can build the crossing sequences and the scaffold map from $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ and (N_1, N_2). Therefore, our algorithm searches exhaustively for an optimal combinatorial scaffold map with detailed coordinates, and build a system to check whether a vertex map exists, that is consistent with the combinatorial scaffold map and gives the Fréchet length at most δ. The following corollary follows immediately from Lemma 9.1.4.

Corollary 9.1.4.1. *Let $\mathcal{R}$ be a piecewise linear surface with m vertices, $\mathcal{S}$ a triangle, and $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$. There is a list of combinatorial scaffold maps (with detailed normal coordinates) L of size $2^{O(m^2)}$ that can be computed in $2^{O(m^2)}$ time and enumerated in $O(m^2)$ space, with the following properties:*

(1) There is at least one $\langle(g, h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), (N_1, N_2)\rangle \in L$ that extends to a tight scaffold map of Fréchet length at most δ.

(2) For any $\langle(g, h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), (N_1, N_2)\rangle \in L$, every coordinate of N_1 or N_2 is at most $2m2^m$.

9.2 A System of Polynomial Size

If f_1 is a tight scaffold map over $(\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}})$ then the crossing points on $\mathcal{S}$ co-locate with vertices of $\widetilde{\mathcal{S}}_V$. The following lemma uses this property to reduce the number of required constraints in our systems.

Lemma 9.2.1. *Let f_1 be a tight scaffold map over refinements $\widetilde{\mathcal{R}} = (\widetilde{\mathcal{R}}_V, \widetilde{\mathcal{R}}_E, \widetilde{\mathcal{R}}_T)$ and $\widetilde{\mathcal{S}} = (\widetilde{\mathcal{S}}_V, \widetilde{\mathcal{S}}_E, \widetilde{\mathcal{S}}_T)$. Let $e \in \widetilde{\mathcal{R}}_E$, $s = (p_1, p_2) \in \widetilde{\mathcal{S}}_E$. Finally, let $x_1, \ldots, x_k$ be the set of all corresponding points on e in order such that $f_1(x_1) = \ldots = f_1(x_k) = p_1$. For any $1 \leq i \leq k$, we have $||x_i - f_1(x_i)|| \leq \max(||x_1 - f_1(x_1)||, ||x_k - f_1(x_k)||)$.*

Proof. Let $x_i = (1-\lambda)x_1 + \lambda x_k$ for a $0 \leq \lambda \leq 1$. We have:

$$\begin{aligned} ||x_i - f_1(x_i)|| &= ||(1-\lambda)x_1 + \lambda x_k - p_1|| \\ &= ||(1-\lambda)(x_1 - p_1) + \lambda(x_k - p_1)|| \\ &\leq \max(||x_1 - p_1||, ||x_k - p_1||) \\ &= \max(||x_1 - f_1(x_1)||, ||x_k - f_1(x_k)||) \end{aligned}$$

The inequality follows from the convexity of the norm function. □

Lemma 9.2.1 implies that we can disregard all constraints of type (8.3) except $O(m^2)$ of them: two for each choice of $e \in \widetilde{\mathcal{R}}_E$ and a portal in $\widetilde{\mathcal{S}}$. Now, we are ready to show how to build our system of inequalities in polynomial space.

Lemma 9.2.2. *Let $\mathcal{R}$ be a piecewise linear surface with m vertices, $\mathcal{S}$ a triangle, and $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$. Also, let $S = \langle (g,h), (\widetilde{\mathcal{R}}, \widetilde{\mathcal{S}}), (N_1, N_2) \rangle$ be a combinatorial scaffold map (with detailed normal coordinates), such that the value of every coordinate of N_1 and N_2 is $2^{O(m)}$. In $2^{O(m)}$ time, a system of polynomial constraints of size $O(m^2)$ can be computed that is feasible if and only if S extends to a tight scaffold map of Fréchet length at most δ.*

Proof. Our algorithm builds variables and constraints similar to the algorithm of Lemma 8.5.1. The number of variables and constraints for vertices and refinements are polynomial (all constraints of type (8.1), (8.2) and (8.6)). We show that in the new case that $\mathcal{S}$ is a triangle we can reduce the number of variables and constraints for crossing points to $O(m^2)$. Each crossing point of $\widetilde{\mathcal{S}}_X$ is co-located with a vertex; therefore, given the final location of vertices and the detailed normal coordinates, the locations of all crossing points are uniquely determined. Hence, all constraints of type (8.4) and (8.5) can be disregarded. Finally, by Lemma 9.2.1, only $O(m^2)$ constraints of type (8.3) can represent all constraints of this type; all others are redundant. □

9.3 Summing up

Now, we are ready to prove the main theorem of this section: deciding the Fréchet distance between a surface and a triangle is in PSPACE. Our result follows from Corollary 9.1.4.1, Lemma 9.2.2, and Lemma 3.0.1.

Theorem 9.3.1. *Let $\mathcal{R}$ be a piecewise linear surface with m vertices, $\mathcal{S}$ be a triangle, and $\delta \geq 0$. There is an algorithm to decide whether $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$ in PSPACE.*

Proof. By Corollary 9.1.4.1, there is a PSPACE algorithm that enumerates a sequence of $2^{O(m^2)}$ combinatorial scaffold maps with detailed normal coordinates such that at least one of them extends to a scaffold map of Fréchet length at most δ if and only if $\delta_F(\mathcal{R}, \mathcal{S}) \leq \delta$. By Lemma 9.2.2, for any combinatorial scaffold map S (with detailed normal coordinates) a system of a polynomial number of inequalities can be built in PSPACE time that is feasible if and only if S extends to a scaffold map of Fréchet length at most δ. Finally, Lemma 3.0.1 ensures that the feasibility of S can be checked in PSPACE. $\square$

Chapter 10: Improved Approximation Algorithm

10.1 Overview

The approximation algorithm for general surfaces has two limitations. First, the running time depends on the area of the surfaces, not the combinatorial complexity of the input. Second, it works only if the input surfaces are composed of fat triangles. In this chapter, we show how to eliminate these limitations.

In the approximation algorithm for general surfaces, the first step is r-refinement of the input surfaces: refining the surfaces into triangulated surfaces with triangles of diameter at most r. As a result each input triangle t may be decomposed into $\Omega(Area(t)/r^2)$ triangles even if the input is composed of fat triangles. If the input can have skinny triangles, a refinement with may have many triangles; in fact the smallest angle in the input will also show up in the running time. With these problems in mind, in this chapter we modify our algorithm to work without the r-refinement step.

Before we talk about how to remove the r-refinement, let's take a look at the purpose of r-refinement. Recall the last step of our approximation algorithm, which constructs an arbitrary scaffold map f_1 consistent with the given normal coordinates. The error of the Fréchet length of f_1 is bounded by the maximum length of edges of two surfaces, and the r-refinement reduces the maximum length of edges to at most r. In other words, the purpose of r-refinement is to limit the freedom of the scaffold map, and it focuses the images of vertices and crossing points close to the optimal location by breaking the surfaces into small pieces. Therefore, to remove the r-refinement, we need to find an alternative way to guarantee that the images of vertices and crossing points are close to their optimal locations.

Let $v \in \mathcal{R}_V$, and let $t \in \mathcal{S}_T$ be the triangle that contains the optimal image of v. Our first observation is that there is a disc of area at most $\pi\delta^2$ in $t \in \mathcal{S}_T$ that contains the optimal location of v's image. The reason is that the largest possible intersection of the ball $Ball(v, \delta)$ and the triangle t is a circle of radius δ, so $Area(Ball(v, \delta) \cap t) \leq \pi\delta^2$.

By this observation, we are able to build a list of approximate locations for images of vertices. Instead of using refinement to limit the images of vertices, for each vertex $v \in \mathcal{R}_V$ and each triangle $t \in \mathcal{S}_T$, we sample $O(\delta^2/r^2)$ points from $Ball(v, \delta) \cap t$ such that if the optimal image of v is in t then it is r-close to at least one of our sample points.

Once we can approximate the images of vertices, our second task is to approximate the crossing points. To that end, we formulate an optimization problem to find the optimal location for crossing points given a combinatorial scaffold map and an exact vertex map. In turn, the solution of this optimization problem can be extended to a nearly optimal homeomorphism, as described in the previous sections.

Unfortunately, our optimization problem contains quadratic constraints, as the distance between points is computed based on the ℓ_2 norm. First, we observe that if we replace the ℓ_2 norm with ℓ_1, we obtain a linear program. So, we can use existing polynomial time methods to obtain a $(1+\varepsilon)$-approximation algorithm with respect to the ℓ_1 norm. This approximation algorithm immediately implies $\sqrt{2}(1+\varepsilon)$-approximation with respect to ℓ_2. Furthermore, we note that we can add several ℓ_1-type constraints to approximate the ℓ_2 norm arbitrarily closely, thereby obtaining a $(1+\varepsilon)$-approximation algorithm with respect to ℓ_2.

10.2 Approximate vertex map

In the previous section, we showed the intuition for removing the dependence on area in the running time of our approximation algorithm. At a high level, our algorithm runs as previously, using a binary search on the value of δ. In this section, we show, for a given $\delta > \delta_F(\mathcal{R}, \mathcal{S})$, how to obtain a homeomorphism of Fréchet length at most $(1+\varepsilon)\delta$. This fits in the general framework described for our approximation algorithm.

We start with a definition that simplifies the exposition.

Definition 10.2.1. *A bijection $f_0 : \widetilde{\mathcal{R}}_V \to \widetilde{\mathcal{S}}_V$ is a λ-**vertex map** if and only if it has the following properties.*

(1) $\widetilde{\mathcal{R}}_V = \mathcal{R}_V \cup \mathcal{R}'_V$, $\widetilde{\mathcal{S}}_V = \mathcal{S}_V \cup \mathcal{S}'_V$, $f_0(\mathcal{R}_V) = \mathcal{S}'_V$, and $f_0(\mathcal{R}'_V) = \mathcal{S}_V$.

(2) f_0 maps boundary vertices to boundary vertices, and it preserves the cyclic order of boundary vertices on each boundary component.

(3) *There exists a homeomorphism $f : \mathcal{R} \to \mathcal{S}$ of Fréchet length at most λ such that $f(\widetilde{\mathcal{R}}_V) = f_0(\widetilde{\mathcal{R}}_V)$.*

Let $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$. To obtain a $(1+\varepsilon)\delta$-vertex map, we construct a grid-based approximation for the images of vertices. For each triangle t of $\mathcal{S}_T$, we put a grid of width $\sqrt{2}\epsilon\delta/4$ on t. For each $v \in \mathcal{R}_V$, we search for the grid cell that contains its optimal image, and pick an arbitrary point in the grid cell as the approximation of its optimal image. To show this approximate vertex map can be extended to a homeomorphism of Fréchet length $(1+\epsilon)\delta$, we reuse the same technique from Chapter 7.

By replacing the triangle with diameter r by a square s with diagonal of length r in Corollary 7.2.1.1 and Lemma 7.2.2, the following lemma is obtained immediately.

Corollary 10.2.1.1. *Let s be a square with diagonal of length r. For any two homeomorphisms $f : s \to s$ and $g : s \to s$, $|\delta_F(f) - \delta_F(g)| \leq r$.*

Lemma 10.2.2. *Let s be a square with diagonal of length r, and let $P, P' \subseteq int(s)$ be finite point sets with the same cardinality. Also, let $g : P \to P'$ be a bijection. There exists a homeomorphism $h : s \to s$ such that*

(1) *$h|_{\partial(s)}$ is the identity map.*

(2) *$h|_P = g$.*

(3) *$\delta_F(h) \leq r$.*

Combining Corollary 10.2.1.1 and Lemma 10.2.2, we show that for any homeomorphism $f : \mathcal{R} \to \mathcal{S}$, for all $v \in \mathcal{R}_V$, if we relax the images $f(v)$ to any points in the same grid cell of $f(v)$, then the Fréchet length will increase by at most r where r is the length of the diagonal of grid cells.

Lemma 10.2.3. *Let $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length δ. Let $Grid(t)$ be the square grid of diagonal length r on the plane of a triangle $t \in \mathcal{R}_T \cup \mathcal{S}_T$. Let $f_0 : \widetilde{\mathcal{R}} \to \widetilde{\mathcal{S}}$ be a vertex map with the following properties*

(1) *For every $v \in \mathcal{R}_V$:*

(i) $f(v)$ *and* $f_0(v)$ *are in the same triangle* t *of* $\mathcal{S}$*, and*

(ii) $f(v)$ *and* $f_0(v)$ *are in the same grid cell of* $Grid(t)$*.*

(2) For every $u \in \mathcal{S}_V$*:*

(i) $f^{-1}(v)$ *and* $f_0^{-1}(v)$ *are in the same triangle* t *of* $\mathcal{R}$*, and*

(ii) $f^{-1}(v)$ *and* $f_0^{-1}(v)$ *are in the same grid cell of* $Grid(t)$*.*

Then, f_0 *is a* $(\delta + 2r)$*-vertex map.*

Proof. By Lemma 10.2.2, there exist two homeomorphisms of Fréchet length at most r, $h' : \mathcal{R} \to \mathcal{R}$ and $h'' : \mathcal{S} \to \mathcal{S}$, such that for any $v \in \mathcal{S}_V$, $h'(f_0^{-1}(v)) = f^{-1}(v)$, and for any $u \in \mathcal{R}_V$, $h''(f(u)) = f_0(u)$. Then $h'' \circ f \circ h'$ is an extension of f_0, and the Fréchet length is at most $\delta + 2r$. □

The following lemma shows that it is possible to obtain a bounded list of vertex maps, one of which is a nearly δ-vertex map.

Lemma 10.2.4. *Let* $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$*. It is possible to compute a list of vertex maps of size* $((|\mathcal{R}_T| + |\mathcal{S}_T|)/\epsilon)^{2(|\mathcal{R}_V|+|\mathcal{S}_V|)}$ *that contains at least one* $(1+\varepsilon)\delta$*-vertex map in time linear in the size of the list.*

Proof. For each vertex $v \in \mathcal{R}_V$ and triangle $t \in \mathcal{S}_T$, let a be the set of points of t that are at distance at most δ from v. It follows that $Area(a) \leq \pi\delta^2$. We put a grid of width $\sqrt{2}\epsilon\delta/4$ on the plane of t, and consider the grid cells intersecting a as a set of candidate cells for $f_0(v)$. The size of this set is $O(1/\epsilon^2)$. For each candidate cell, choose an arbitrary point in the grid cell as a candidate location for $f_0(v)$. Overall, for each vertex $v \in \mathcal{R}_V$, we have a list $L_V(v)$ of candidate locations for $f_0(v)$ on $\mathcal{S}$ whose size is $O(|\mathcal{S}_T|/\epsilon^2)$. Similarly, for each vertex $u \in \mathcal{S}_V$, we have a list $L_V(u)$ of candidate locations for $f_0^{-1}(u)$ on $\mathcal{R}$ whose size is $O(|\mathcal{R}_T|/\epsilon^2)$.

Now let $f : \mathcal{R} \to \mathcal{S}$ be any homeomorphism of Fréchet length at most δ. Therefore, by Corollary 10.2.1.1 and Lemma 10.2.2, we can obtain a list that contains a $(1+\varepsilon)\delta$-vertex map by enumerating all locations in $L(v)$ for $f_0(v)$ for every vertex $v \in \mathcal{R}_V$, and all locations in $L(u)$ for $f_0^{-1}(u)$ for every vertex $u \in \mathcal{S}_V$. The total number of the vertex maps that we enumerate is

$$O((|\mathcal{R}_T|/\epsilon^2)^{|\mathcal{S}_V|} \times (|\mathcal{S}_T|/\epsilon^2)^{|\mathcal{R}_V|}) \leq O\left(\left(\frac{|\mathcal{S}_T| + |\mathcal{R}_T|}{\epsilon}\right)^{2(|\mathcal{S}_V|+|\mathcal{R}_V|)}\right)$$

□

10.3 Optimization problem

For a given δ, we build a list of vertex maps as described, which is guaranteed to contain a $(1+\varepsilon)\delta$-vertex map provided $\delta \geq \delta_F(\mathcal{S}, \mathcal{R})$. For each vertex map f_0 in our list, we look for a scaffold map f_1 with the minimum Fréchet length that is consistent with f_0. To that end, we enumerate over a restricted set of combinatorial scaffold maps similar to our previous approximation algorithm. For each combinatorial map, we formulate an optimization problem for computing the optimal locations for the crossing points.

We use one variable to specify each crossing point, and set linear constraints to ensure the order of crossing points on each edge. The objective function is to minimize the maximum distance between pairs of crossing points corresponding to the normal coordinates. In this section, we will describe the optimization problem, and give the general form.

For a given vertex map f_0, and normal coordinates N, the optimization problem only has variables for crossing points. Let v be a crossing point on edge $r = (a, b)$ of $\widetilde{\mathcal{R}}_E$; we specify v by one variable x, such that $v = a + x \cdot \overrightarrow{ab}$. For each edge $e \in \widetilde{\mathcal{R}}_E$, the normal coordinate $k = N(e)$ specifies the number of crossing points on e. Let $x_1, x_2, \ldots, x_k$ be the sequence of crossing points on e corresponding to the normal coordinates. To ensure their order, we have the following constraints:

$$0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \leq 1$$

Similarly, let $u = f_1(v)$ be a crossing point on edge $s = (c, d)$ of $\widetilde{\mathcal{S}}_E$. We specify u by $u = c + y \cdot \overrightarrow{cd}$, and we have the same constraints to ensure the order of crossing points on each edge $e \in \widetilde{\mathcal{S}}_R$.

$$0 \leq y_1 \leq y_2 \leq \cdots \leq y_l \leq 1$$

Then, let θ be the variable that specifies the maximum distance between crossing points and their images. Let v be a crossing point on edge $\overline{ab}$ specified by the variable x, and let $u = f_1(v)$ be on edge $\overline{cd}$ specified by the variable y. We have the following constraint

$$||v - u|| = ||(a + x \cdot \overrightarrow{ab}) - (c + y \cdot \overrightarrow{cd})|| \leq \theta$$

Because a, b, c, d are all fixed vertices, the constraint has degree p under the ℓ_p norm.

The objective function is to minimize θ, and the optimization problem has the following form

$$\begin{array}{rcll}
\text{minimize} & & \theta & \\
x_1^e & \geq & 0 & \forall e \in \widetilde{\mathcal{R}}_E \\
y_1^e & \geq & 0 & \forall e \in \widetilde{\mathcal{S}}_E \\
x_k^e & \leq & 1 & \forall e, k : e \in \widetilde{\mathcal{R}}_E, k = N(e) \\
y_k^e & \leq & 1 & \forall e, k : e \in \widetilde{\mathcal{S}}_E, k = N(e) \\
x_i^e & \leq & x_{i+1}^e & \forall e, i : e \in \widetilde{\mathcal{R}}_E, i \in \{1, 2, \ldots, N(e) - 1\} \\
y_i^e & \leq & y_{i+1}^e & \forall e, i : e \in \widetilde{\mathcal{S}}_E, i \in \{1, 2, \ldots, N(e) - 1\} \\
||(a + x_i^e \cdot \overrightarrow{ab}) - (c + y_j^{e'} \cdot \overrightarrow{cd})|| & \leq & \theta & \forall e, i : e \in \widetilde{\mathcal{R}}_E, i \in \{1, 2, \ldots, N(e)\}, \\
 & & & \qquad y_j^{e'} = f_1(x_i^e)
\end{array}$$

If we use the ℓ_1 norm, the distance function will be linear. So, the optimum with respect to ℓ_1 can be approximated using linear programming.

Theorem 10.3.1. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces. There is a $(1 + \varepsilon)$-approximation algorithm for deciding whether the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ no more than δ with respect to the ℓ_1 norm in time*

$$(1/\epsilon)^{O(|\mathcal{R}_V| + |\mathcal{S}_V|)} 2^{O((|\mathcal{R}_V| + |\mathcal{S}_V|)^2)}$$

time.

Proof. By Lemma 10.2.4, we can build a list L of size $((|\mathcal{R}_T| + |\mathcal{S}_T|)/\epsilon)^{2(|\mathcal{R}_V| + |\mathcal{S}_V|)}$ that contains at least one $(1+\varepsilon)$-approximation vertex map in time $((|\mathcal{S}_T|+|\mathcal{R}_T|)/\epsilon)^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}$. For each vertex map in L, we enumerate over a set of combinatorial scaffold maps of size $2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)}$. For each combinatorial scaffold map, we formulate an optimization problem for computing the optimal locations for the crossing points. The optimization problem contains $2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}$ variables and constraints, and it can be solved in $2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}$ time by Lemma 3.0.3. If the algorithm find there exists an optimization problem whose solution is no more than $(1+\varepsilon)\delta$, then the algorithm accepts. Otherwise, it rejects. The total running time of the algorithm is

$$((|\mathcal{S}_T|+|\mathcal{R}_T|)/\epsilon)^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)} = (1/\epsilon)^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)}$$

Correctness: If $\delta \geq \delta_F(\mathcal{S},\mathcal{R})$, then L contains at least one $(1+\varepsilon)\delta$-approximate vertex map, and the algorithm always accepts δ. If $\delta < \delta_F(\mathcal{S},\mathcal{R})/(1+\varepsilon)$, then there doesn't exist a $(1+\varepsilon)\delta$-approximate vertex map, the algorithm always rejects δ.

□

Theorem 10.3.2. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces, and $\delta = \delta_F(\mathcal{R},\mathcal{S}) > 0$. There is a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$ with respect to the ℓ_1 norm in*

$$\log(\delta+1/\delta)(1/\epsilon)^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)}2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)}$$

time.

Proof. The algorithm applies exponential search starting with the real number 1 to find a range (x, y), where $y \leq \sqrt{1+\epsilon}x$, x is rejected and y is accepted by the $1+(\sqrt{1+\epsilon}-1)$-approximation decision algorithm. By Theorem 10.3.1, we know that $x < \delta_F(\mathcal{R},\mathcal{S})$ and $\sqrt{1+\epsilon}y \geq \delta_F(\mathcal{R},\mathcal{S})$. Therefore, $\delta_F(\mathcal{R},\mathcal{S}) \leq \sqrt{1+\epsilon}y \leq (1+\varepsilon)x < (1+\varepsilon)\delta_F(\mathcal{R},\mathcal{S})$. □

This, in turn implies a $\sqrt{2}\cdot(1+\varepsilon)$-approximation algorithm for the ℓ_2 norm. In a sense, we obtain the $\sqrt{2}$ factor because we are estimating an ℓ_2 ball with the smallest ℓ_1 ball that contains it. We can obtain better approximation factors, in fact arbitrarily small factors, by estimating the ℓ_2 ball as the intersection of several ℓ_1 balls. That approach would result in a $(1+\varepsilon)$-approximation algorithm. In the following section, we obtain a $(1+\varepsilon)$-approximation algorithm by directly solving the optimization problem for the ℓ_2 norm.

10.4 Convex Quadratically Constrained Quadratic Programming for ℓ_2 norm

In this section, we show that this optimization problem for the ℓ_2 norm is an instance of convex quadratically constrained quadratic programming. Thus, it can be solved using standard semidefinite programming solvers.

Definition 10.4.1. *A convex quadratically constrained quadratic programming problem has the form*

$$\begin{array}{rl} \textit{minimize} & \frac{1}{2}x^T P_0 x + q_0^T x \\ \textit{subject to} & \frac{1}{2}x^T P_i x + q_i^T x + r_i \leq 0 \quad \textit{for } i = 1, \ldots, m, \\ & Ax = b, \end{array}$$

where $P_0, \ldots, P_M$ are n-by-n positive semidefinite matrices.

First, we re-formulate the optimization problem for the ℓ_2 norm:

$$\begin{array}{rcll} \text{minimize} & & \theta' & \\ x_1^e & \geq & 0 & \forall e \in \widetilde{\mathcal{R}}_E \\ y_1^e & \geq & 0 & \forall e \in \widetilde{\mathcal{S}}_E \\ x_k^e & \leq & 1 & \forall e, k : e \in \widetilde{\mathcal{R}}_E, k = N(e) \\ y_k^e & \leq & 1 & \forall e, k : e \in \widetilde{\mathcal{S}}_E, k = N(e) \\ x_i^e & \leq & x_{i+1}^e & \forall e, i : e \in \widetilde{\mathcal{R}}_E, i \in \{1, 2, \ldots, N(e) - 1\} \\ y_i^e & \leq & y_{i+1}^e & \forall e, i : e \in \widetilde{\mathcal{S}}_E, i \in \{1, 2, \ldots, N(e) - 1\} \\ ||(a + x_i^e \cdot \overrightarrow{ab}) - (c + y_j^{e'} \cdot \overrightarrow{cd})||_2^2 & \leq & \theta' & \forall e, i : e \in \widetilde{\mathcal{R}}_E, i \in \{1, 2, \ldots, N(e)\}, \\ & & & \qquad y_j^{e'} = f_1(x_i^e) \end{array}$$

Instead of minimizing the distance between crossing points, we minimize the square of the distance between them.

Lemma 10.4.2. *The optimization problem for the ℓ_2 norm is a convex quadratically constrained quadratic program.*

Proof. The objective function and all constraints to ensure the order of crossing points are linear. It remains to show that the distance constraints are convex with semidefinite matrix coefficients. To show the convexity, we expand the distance constraints:

$$\begin{aligned}
&||(a + x \cdot \overrightarrow{ab}) - (c + y \cdot \overrightarrow{cd})||^2 \leq \theta' \\
&\Rightarrow ||(a - c) + (x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd})||^2 \leq \theta' \\
&\Rightarrow ||x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}||^2 + 2(a - c) \cdot (x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}) - \theta' + ||a - c||^2 \leq 0
\end{aligned}$$

$||a - c||^2$ is constant, and $2(a - c) \cdot (x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}) - \theta'$ is linear. It remains to show that $||x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}||^2$ can be written as a quadratic form with a semidefinite matrix. We have:

$$||x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}||^2 = Ax^2 + Bxy + Cy^2$$

For constants A, B, and C. Let

$$P = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix}.$$

We have

$$\begin{bmatrix} x & y \end{bmatrix} P \begin{bmatrix} x & y \end{bmatrix}^T = Ax^2 + Bxy + Cy^2 = ||x \cdot \overrightarrow{ab} - y \cdot \overrightarrow{cd}||^2 \geq 0$$

But, the above inequality holds for any point (x, y), that is, P is a positive semi-definite matrix. Further, P is symmetric by its construction. So, the statement of the lemma holds. □

Basu et al. [5] introduced a technique that approximates quadratic constraints by a set of linear constraints. By applying this technique to our problem, the optimization problem becomes a linear programming problem, because the objective is also linear.

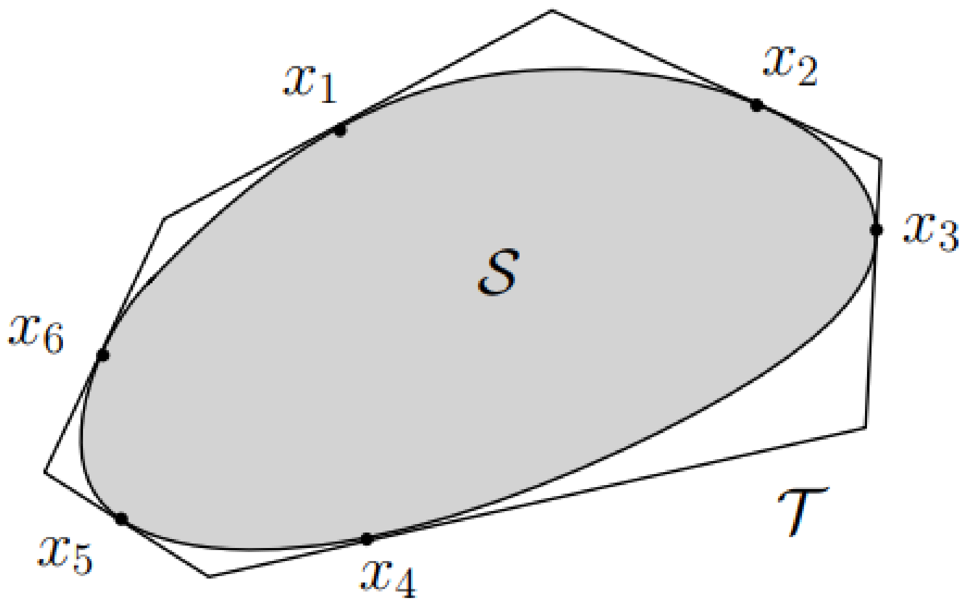


Figure 10.1: Approximating a quadratic constraint by linear constraints. .cFte-basu2017largescale.

Unfortunately, we cannot show that the number of linear constraints needed to get a $(1 + .vFrepsilon)\delta$-approximation of the original quadratic constraints is polynomial in the problem description size and $1/\epsilon$. A trivial construction is adding portals of width $\epsilon\delta$ on each edge of $\widetilde{\mathcal{S}}$ and $\widetilde{\mathcal{R}}$. It is easy to show that for each quadratic constraint, there exists a set of linear constraints of size $O(\ell/(\epsilon\delta))$ that is a $(1+\varepsilon)\delta$-approximation of the quadratic constraint, where ℓ is the length of edges. Hence, this construction builds a linear program of size $O(2^{O((|\mathcal{R}_V|+|\mathcal{S}_V|)^2)}(Length(\mathcal{S}_E) + Length(\mathcal{R}_E))/(\epsilon\delta))$.

On the other hand, the SDP problem can be solved in polynomial time by Lemma 3.0.2. The final theorem of this section follows from Theorem 10.3.2 and Lemma 10.4.2.

Theorem 10.4.3. *Let $\mathcal{R}$ and $\mathcal{S}$ be piecewise linear surfaces, and $\delta = \delta_F(\mathcal{R}, \mathcal{S}) > 0$. There is a $(1+\varepsilon)$-approximation algorithm for computing the Fréchet distance between*

$\mathcal{R}$ and $\mathcal{S}$ with running time

$$\log(1/\delta + \delta)(1/\epsilon)^{O(|\mathcal{R}_V| + |\mathcal{S}_V|)} 2^{O((|\mathcal{R}_V| + |\mathcal{S}_V|)^2)}$$

Chapter 11: Surfaces composed of large triangles

In this chapter, we describe a new restricted class of surfaces, surfaces composed of large triangles. This restricted class of surfaces is inspired by the study on curves with long edges [17]. The definition of surfaces composed of large triangles is as following:

Definition 11.0.1. *Let t be a triangle. If the radius of the incircle of t is no less than θ, then t is a θ-**large triangle**.*

Definition 11.0.2. *Let $\mathcal{S}$ be a piecewise linear surface. If all triangles of $\mathcal{S}$ are θ-large triangles, then $\mathcal{S}$ is a **surface composed of θ-large triangles**.*

Let $\mathcal{R}$ and $\mathcal{S}$ be two surfaces composed of $\delta_F(\mathcal{R}, \mathcal{S})$-large triangles. We show that there is an exact algorithm to compute the Fréchet distance between $\mathcal{R}$ and $\mathcal{S}$.

11.1 Optimal Vertex Map

In the previous chapter, the improved approximation algorithm consists of two stages, searching for an approximate vertex map and computing an optimal scaffold map. If an optimal vertex map can be computed in the first stage, then we obtain an exact algorithm. We show that for surface composed of large triangles, the optimal vertex map can be computed.

Lemma 11.1.1. *Let $\mathcal{R}$ and $\mathcal{S}$ be two surfaces composed of δ-large triangles, and $f : \mathcal{R} \to \mathcal{S}$ be a homeomorphism of Fréchet length no more than δ, and there exists a homeomorphism $f_0 : \mathcal{R} \to \mathcal{S}$ of Fréchet length no more than δ with the following properties:*

(1) For every $v \in \mathcal{R}_V$:

- *(i) $f(v)$ and $f_0(v)$ are in the same triangle of $\mathcal{S}$, and*
- *(ii) $f_0(v)$ is the closest point to v in the triangle.*

(2) For every $u \in \mathcal{S}_V$:

(i) $f^{-1}(u)$ and $f_0^{-1}(u)$ are in the same triangle of $\mathcal{R}$, and

(ii) $f_0^{-1}(u)$ is the closest point to u in the triangle.

Proof. Let v_0 be a vertex in $\mathcal{R}_V$, and $A(v_0) = \{v_1, \ldots, v_k\}$ be the list of adjacent vertices of v_0 in clockwise order, $E(v_0) = \{(v_1, v_2), (v_2, v_3), \ldots, (v_k, v_1)\}$ be the edges between $A(v_0)$. Because f is a homeomorphism of Fréchet length no more than δ, for a point $p \in \mathcal{S}$, the image $f(p) \in Ball(p, \delta)$. The radius of any triangle is no less than δ, the distance from v_0 to the edges between its adjacent vertices $E(v_0)$ is no less than 2δ. Therefore, for each point p on $E(v_0)$, $Ball(v_0, \delta) \cap Ball(p, \delta) = \emptyset$. In other words, for all homeomorphisms f of Fréchet length no more than δ, $Ball(v_0, \delta) \cap Ball(p, \delta) = \emptyset$. Let $T(v_0)$ be the triangle that contains $f(v_0)$, and $Disk(v_0) = Ball(v_0, \delta) \cap T(v_0)$. Consider the scaffold map f_1 of the homeomorphism f. Because $Ball(v_0, \delta) \cap Ball(p, \delta) = \emptyset$, $Disk(v_0)$ doesn't contains any image $f_1(p)$ for any point $p \in E(v_0)$. Therefore, by moving $f_1(v_0)$ anywhere in $Disk(v_0)$, we obtain a new scaffold map f_1' without increasing the Fréchet length. So, there exists a scaffold map that maps each vertex $v \in \mathcal{R}_V$ to the center of $Disk(v)$, that is, the closest point to v in the triangle $T(v_0)$.

Because (1) and (2) are symmetrical, we can obtain a scaffold map f_1'' from f_1' by mapping the closest point in the triangle containing $f_1'^{-1}(u)$ to $u \in \mathcal{S}_V$ without increasing the Fréchet length. Finally, we extend f_1'' to a homeomorphism f'' of Fréchet length no more than δ. □

The following lemma can be obtained from the above lemma.

Lemma 11.1.2. *Let $\mathcal{R} = (\mathcal{R}_V, \mathcal{R}_E, \mathcal{R}_T)$ and $\mathcal{S} = (\mathcal{S}_V, \mathcal{S}_E, \mathcal{S}_T)$ be two surfaces composed of large triangles. There exists a vertex map f_0 that can be extended to an optimal homeomorphism of Fréchet length $\delta_F(\mathcal{R}, \mathcal{S})$, with the following properties:*

(1) $\widetilde{\mathcal{R}}_V = \mathcal{R}_V \cup f_0^{-1}(S_V)$, $\widetilde{\mathcal{S}}_V = \mathcal{S}_V \cup f_0(R_V)$.

(2) f_0 maps boundary vertices of $\widetilde{\mathcal{R}}_V$ to boundary vertices of $\widetilde{\mathcal{S}}_V$, and it preserves the cyclic order of boundary vertices on each boundary component.

(3) For every $v \in \mathcal{R}_V$:

(i) $f_0(v)$ *is a vertex in* $\mathcal{S}_V$*, or*

(ii) $f_0(v)$ *is the closest point in the triangle of* $\mathcal{S}_T$ *containing* $f_0(v)$*, or*

(iii) $f_0(v)$ *is the closest point on the edge of* $\mathcal{S}_E$ *containing* $f_0(v)$*.*

(4) For every $u \in \mathcal{S}_V$*:*

(i) $f_0^{-1}(u)$ *is a vertex in* $\mathcal{R}_V$*, or*

(ii) $f_0^{-1}(u)$ *is the closest point in the triangle of* $\mathcal{R}_T$ *containing* $f_0^{-1}(u)$*, or*

(iii) $f_0^{-1}(u)$ *is the closest point on the edge of* $\mathcal{R}_E$ *containing* $f_0^{-1}(u)$*.*

(5) There exists an optimal homeomorphism $f : \mathcal{R} \to \mathcal{S}$ *of Fréchet length* $\delta_F(\mathcal{R}, \mathcal{S})$ *such that* $f(\widetilde{\mathcal{R}}_V) = f_0(\widetilde{\mathcal{R}}_V)$*. For each vertex* $v \in \widetilde{\mathcal{R}}_V$*,* $f_0(v)$ *is the closest point to* v *in the triangle containing* $f_0(v)$*.*

Clearly, the optimal vertex map between two surfaces composed of large triangles can be found by enumeration. Replacing the approximate vertex map by the optimal vertex map in the first stage of our improved approximation algorithm, an exact algorithm for computing the Fréchet distance between two surfaces composed of large triangles can be obtained.

Theorem 11.1.3. *Let* $\mathcal{R}$ *and* $\mathcal{S}$ *be two surfaces composed of* 2δ*-large triangles, such that* $\delta \geq \delta_F(\mathcal{R}, \mathcal{S})$*. There is an exact algorithm for computing* $\delta_F(\mathcal{R}, \mathcal{S})$ *in*

$$2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)^2}$$

time.

Proof. For each $v \in \mathcal{R}_V$, we enumerate all vertices in $\mathcal{S}_V$ and all closest points to v on triangles in $\mathcal{S}_T$ and all closest points to v on edges in $\mathcal{S}_E$. Also, for each $u \in \mathcal{S}_V$, we enumerate all vertices in $\mathcal{R}_V$ and all closest points to u on triangles in $\mathcal{R}_T$ and all closest points to u on edges in $\mathcal{R}_E$. By Lemma 11.1.2, the enumeration builds a list L of size $O(|\mathcal{R}_V|^{O(|\mathcal{S}_V|)} + |\mathcal{S}_V|^{O(|\mathcal{R}_V|)})$ that contains at least one optimal vertex map. For each vertex map in L, we enumerate over a set of combinatorial scaffold maps of size $2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)^2}$ by enumerating all possible normal coordinates. For each combinatorial scaffold map, we build an optimization problem for computing the optimal locations for the crossing points. Because $\delta(\mathcal{R}, \mathcal{S})$ is the solution of the optimization problem for the optimal combinatorial scaffold map, and the algorithm returns the minimum value over the solutions of all optimization problems, the algorithm computes $\delta_F(\mathcal{R}, \mathcal{S})$ exactly. The optimization problem has size $O(2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)})$, and it can be solved in polynomial time. The total running time for the algorithm is

$$O(|\mathcal{R}_V|^{O(|\mathcal{S}_V|)} + |\mathcal{S}_V|^{O(|\mathcal{R}_V|)}) \times 2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)^2} \times 2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)} = 2^{O(|\mathcal{R}_V|+|\mathcal{S}_V|)^2}.$$

□

11.2 Future work

The running time of the exact algorithm is limited by enumerating all normal coordinates. Therefore, one possible way to improve the algorithm is to show that the crossing number of each edge is linear in the number of vertices. Another possible approach would extend the work of Gudmundsson et al. [17], who gave a greedy algorithm to compute the Fréchet distance between curves with long edges. It is an open problem whether there exists a greedy algorithm for computing the Fréchet distance between surfaces composed of large triangles.

Chapter 12: Discussion

We presented exact and approximation algorithms for computing the Fréchet distance between two surfaces in this thesis. Our results suggest two natural directions for future research that we elaborate here.

Computational complexity of the Fréchet distance. We show for the first time that the Fréchet distance between two surfaces is decidable. Moreover, we show that computing the Fréchet distance between a triangle and a surface is in PSPACE. Determining the computational complexity of the problem remains open.

Our algorithms rely on enumerating all skeleton maps, and in particular edge maps f_1. The number of these maps is bounded by the number of times that the image of an edge of $\mathcal{R}$ can cross an edge of $\mathcal{S}$. So, a natural way to obtain faster algorithms is to find better bounds. In fact, it suffices to show that all edges are crossed in some optimal maps at a polynomial number of points, even if the number of crossings is exponential.

Practical algorithms. Most of the algorithms in this thesis are prohibitively slow for practical purposes. The main reason is that computing the Fréchet distance between general surfaces is a computationally hard problem. We asked if this measure can be computed efficiently for special classes of surfaces that appear in practice. To this end, we studied the class of surfaces with large triangles. We show faster algorithms for this class. A future direction is to find more general classes for which the Fréchet distance can be computed or approximated efficiently.

Bibliography

[1] Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete Fréchet distance in subquadratic time. *CoRR*, abs/1204.5333, 2012. URL: http://arxiv.org/abs/1204.5333, arXiv:1204.5333.

[2] Helmut Alt and Maike Buchin. Semi-computability of the Fréchet distance between surfaces. In *Proceedings of the 21st European Workshop on Computational Geometery*, pages 45 – 48, 2005.

[3] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.

[4] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. In *Proceedings of the 14th Conference on Annual European Symposium - Volume 14*, ESA'06, pages 52–63, London, UK, UK, 2006. Springer-Verlag. URL: http://dx.doi.org/10.1007/11841036_8, doi:10.1007/11841036_8.

[5] Kinjal Basu, Ankan Saha, and Shaunak Chatterjee. Large-scale quadratically constrained quadratic program via low-discrepancy sequences, 2017. arXiv:1710.01163.

[6] Aharon Ben-Tal and Nemirovskiaei. *Lectures on Modern Convex Optimization*. 2019. URL: https://www2.isye.gatech.edu/~nemirovs/LMCO_LN.pdf.

[7] Kevin Buchin, Maike Buchin, and André Schulz. Fréchet distance of surfaces: Some simple hard cases. In *Proceedings of the 18th Annual European Conference on Algorithms: Part II*, ESA'10, pages 63–74, Berlin, Heidelberg, 2010. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1882123.1882132.

[8] Kevin Buchin, Maike Buchin, and Carola Wenk. Computing the Fréchet distance between simple polygons. *Comp. Geom. Theo. Appl.*, 41(1-2):2–20, October 2008. URL: http://dx.doi.org/10.1016/j.comgeo.2007.08.003, doi:10.1016/j.comgeo.2007.08.003.

[9] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Computing the similarity between moving curves. *CoRR*, abs/1507.03819, 2015. URL: `http://arxiv.org/abs/1507.03819`, `arXiv:1507.03819`.

[10] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Computing the Fréchet distance between real-valued surfaces. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2443–2455, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=3039686.3039848`.

[11] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 460–467, New York, NY, USA, 1988. ACM. URL: `http://doi.acm.org/10.1145/62212.62257`, `doi:10.1145/62212.62257`.

[12] Atlas F. Cook IV, Anne Driemel, Jessica Sherette, and Carola Wenk. Computing the Fréchet distance between folded polygons. *Computational Geometry*, 50:1 – 16, 2015. URL: `http://www.sciencedirect.com/science/article/pii/S0925772115000760`, `doi:http://dx.doi.org/10.1016/j.comgeo.2015.08.002`.

[13] Éric Colin de Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM J. Comput.*, 39(8):3784–3813, December 2010. URL: `http://dx.doi.org/10.1137/090761653`, `doi:10.1137/090761653`.

[14] Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. *CoRR*, abs/1003.0460, 2010. URL: `http://arxiv.org/abs/1003.0460`, `arXiv:1003.0460`.

[15] Jeff Erickson and Amir Nayyeri. Shortest non-crossing walks in the plane. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 297–308, 2011.

[16] Michael Godau. *On the Complexity of Measuring the Similarity Between Geometric Objects in Higher Dimensions*. PhD thesis, Freie Universität Berlin, 1998.

[17] Joachim Gudmundsson, Majid Mirzanezhad, Ali Mohades, and Carola Wenk. Fast Fréchet distance between curves with long edges. *CoRR*, abs/1710.10521, 2017. URL: `http://arxiv.org/abs/1710.10521`, `arXiv:1710.10521`.

[18] Sariel Har-Peled and Benjamin Raichel. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms*, 10(1):3, 2014. URL: http://doi.acm.org/10.1145/2532646, doi:10.1145/2532646.

[19] Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel Journal of Mathematics*, 51(1-2):90–120, 1985. URL: http://dx.doi.org/10.1007/BF02772960, doi:10.1007/BF02772960.

[20] Man-Soon Kim, Sang-Wook Kim, and Miyoung Shin. Optimization of subsequence matching under time warping in time-series databases. In *Proceedings of ACM Symposium on Applied Computing*, pages 581–586, 2005.

[21] S. Kwong, Q. H. He, K. F. Man, K. S. Tang, and C. W. Chau. Parallel genetic-based hybrid pattern matching algorithm for isolated word recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(05):573–594, 1998. URL: https://doi.org/10.1142/S0218001498000348, arXiv:https://doi.org/10.1142/S0218001498000348, doi:10.1142/S0218001498000348.

[22] Amir Nayyeri and Anastasios Sidiropoulos. Computing the Fréchet distance between polygons with holes. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 997–1009, 2015. URL: http://dx.doi.org/10.1007/978-3-662-47672-7_81, doi:10.1007/978-3-662-47672-7_81.

[23] E. Sriraghavendra, K. K., and C. Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 461–465, Sept 2007. doi:10.1109/ICDAR.2007.4378752.