Computing the Fréchet Distance Between Simple Polygons

Kevin Buchin*†, Maike Buchin*†, Carola Wenk‡

Abstract

We present the first polynomial-time algorithm for computing the Fréchet distance for a non-trivial class of surfaces: simple polygons. For this, we show that it suffices to consider homeomorphisms that map an arbitrary triangulation of one polygon to the other polygon such that diagonals of the triangulation are mapped to shortest paths in the other polygon.

1 Introduction

The Fréchet distance is a distance measure used in shape matching. It is defined for continuous shapes such as curves and surfaces using reparametrizations of the shapes.

The Fréchet distance between polygonal curves can be computed in polynomial time [2], however computing the Fréchet distance distance for (two-dimensional) surfaces is NP-hard [5]. Except for the NP-hardness very little is known so far about the Fréchet distance of surfaces. It is known to be semi-computable [1], but it is unknown whether it is computable, and there are no approximation algorithms.

We address this problem by considering a restricted but important class of surfaces, simple polygons, and show that their Fréchet distance can be computed in polynomial time. This is the first polynomial-time algorithm for computing the Fréchet distance for a non-trivial class of surfaces.

The rest of this abstract is organized as follows: First we introduce in Section 2 notations and preliminary lemmas. Then we show in Section 3 that it suffices to look at a small well-behaved class of homeomorphisms. We use this to develop a polynomial time algorithm for deciding the Fréchet distance in Section 4 which we to a computation algorithm by searching over a set of critical values.

Due to space restrictions we omit detailed proofs in this extended abstract but provide the main ideas.

*Freie Universität Berlin, Institute of Computer Science, Takustr. 9, 14195 Berlin, Germany

†This research was supported by the Deutsche Forschungsgemeinschaft within the European graduate program 'Combinatorics, Geometry, and Computation' (No. GRK 588/2).

‡University of Texas at San Antonio, Department of Computer Science, 6900 N. Loop 1604 West, San Antonio, TX 78249-0667, USA

2 Preliminaries

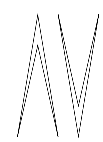
Simple Polygons

Let P and Q be two simple polygons in the plane with m and n vertices, respectively. A simple polygon is the area enclosed by a non-selfintersecting closed polygonal curve in the plane. The two polygons may lie in two different planes. We assume the underlying parametrizations $f : P \to P$ and $g : Q \to Q$. Their Fréchet distance is:

$$\delta_F(P,Q) = \inf_{\sigma: P \to Q} \max_{t \in P} ||t - \sigma(t)|| \, ,$$

where σ ranges over all orientation-preserving homeomorphisms and $||.||$ is the Euclidean norm[1]. In the remainder we will only consider orientation-preserving homeomorphisms and might refer only to σ or to a homeomorphism when the meaning is clear from the context.

The first question that comes to mind is: Is the Fréchet distance of polygons different from the Fréchet distance of their boundary curves?



Observation 1 *The Fréchet distance of two polygons may be arbitrarily larger than the Fréchet distance of their boundary curves.*

This observation can be proved by showing that it holds for the two polygons on the right.

We will compute the Fréchet distance between simple polygons using shortest paths and for this use an important concept which was introduced by Guibas et al. [6]: *hourglasses*. If s_1 and s_2 are two segments in a simple polygon, the hourglass of s_1 and s_2 represents all shortest paths between any point on s_1 and any point on s_2.

Simplifying a Curve

Given a curve f and a line segment s, Lemma 1 shows that simplifying f by replacing a part of it with a line segment does not increase the Fréchet distance to s.

Lemma 1 *Let $f : [0,1] \to \mathbb{R}^d$ be a curve, let $s : [0,1] \to \mathbb{R}^d$ be a line segment, and let $0 \le t_1 < t_2 \le 1$. Define $f' : [0,1] \to \mathbb{R}^d$ to be equal to f for $t \in [0,t_1] \cup$*

[1]Of course any other metric can be considered as well.

$[t_2, 1]$. *And for* $t \in [t_1, t_2]$ *it equals the line segment from* $f(t_1)$ *to* $f(t_2)$. *Then* $\delta_F(f', s) \leq \delta_F(f, s)$.

For proving this lemma, we show that a homeomorphism σ for computing $\delta_F(f, s)$ can be modified to a homeomorphism σ' for computing $\delta_F(f', s)$ that does not yield a larger value. For this we use that the maximum distance between two segments is attained at segment end points.

3 Shortest Paths Lemma

The next lemma states that it suffices to look at homeomorphisms that map the diagonals of a triangulation of P to shortest paths in Q and are piecewise linear inside triangles. For a triangulation T of P we denote with E_T the set of points lying on all edges of T, i.e., all points on all boundary edges and diagonals (and vertices) of T.

Lemma 2 *Given two simple polygons* P *and* Q*, a triangulation* T *of* P *and homeomorphism* $\sigma : P \to Q$. *Then there is a map* $\sigma' : P \to Q$ *which fulfills*

(1) σ' *is a limit of homeomorphisms from* $P \to Q$

(2) σ' *maps the diagonals of the triangulation* T *to shortest paths in* Q *and is piecewise linear inside triangles without introducing interior vertices. Thus* $\max_{t \in P} ||t - \sigma'(t)|| = \max_{t \in E_T} ||t - \sigma'(t)||$.

(3) σ' *is "at least as good as* σ*", i.e.,* $\max_{t \in P} ||t - \sigma'(t)|| \leq \max_{t \in P} ||t - \sigma(t)||$.

Proof. Let σ' be as follows: σ' equals σ on the boundary, it maps diagonals of T to the shortest paths between the corresponding boundary points of Q, and is extended piecewise linearly inside triangles (without introducing any interior vertices).

Then (2) holds by definition of σ' and (1) holds because the shortest paths may be overlapping but noncrossing. We can show (3) by iteratively shooting rays along the edges of the shortest path and simplifying the curve $\sigma(D)$ using Lemma 1. □

From this lemma we get the following corollaries:

Corollary 3 *The Fréchet distance between simple polygons* P *and* Q *equals*

$$\inf_{\sigma' : P \to Q} \max_{t \in T} ||t - \sigma'(t)||$$

where T *is an arbitrary triangulation of* P. σ' *ranges over all homeomorphisms from the boundary of* P *to the boundary of* Q *which are extended to* T *by mapping the diagonals of* T *to the shortest paths between the boundary vertices and which are extended piecewise linearly inside the triangles (without introducing interior vertices).*

Corollary 4 *The Fréchet distance between two simple polygons, of which one polygon is convex, equals the Fréchet distance between their boundary curves.*

The first corollary follows immediately. For the second corollary we triangulate the possibly non-convex polygon and map the diagonals of the triangulation to shortest paths in the convex polygon. Then the shortest paths in the convex polygon are also diagonals and the Fréchet distance between two line segments equals the maximum distance of its endpoints.

4 Computing the Fréchet distance

The main result of this section is a polynomial time algorithm for computing the Fréchet distance between simple polygons. But first we have to show some preliminary results and introduce some more notation.

In the following P and Q always denote two simple polygons, n and m the number of vertices of the boundaries of P and Q, respectively, and ε a real value greater 0. T is a triangulation of P. The decision problem is to decide whether $\delta_F(P, Q) \leq \varepsilon$.

Free Space Diagram and Reachability Structure

The *free space diagram* is the data structure developed by Alt and Godau [2] for computing the Fréchet distance between polygonal curves. For $\varepsilon > 0$ and two parametrized curves $f, g : [0, 1] \to \mathbb{R}^d$ it is defined as $\{(s, t) \in [0, 1]^2 \,|\, ||f(s) - g(t)|| \leq \varepsilon\}$. If f and g are polygonal curves of complexity n and m, respectively, then the free space diagram can be represented in a rectangle $[0, n] \times [0, m]$ consisting of n columns and m rows of a total of mn *cells*. The *double free space diagram* is the free space diagram of f concatenated f and g and can be represented in $[0, 2n] \times [0, m]$.

The decision problem for closed polygonal curves is solved by computing the *reachability structure* [2]. It is a partition of the boundary of the double free space diagram into $O(mn)$ intervals, where each intervals is assigned pointers containing information on the reachability in free space. The reachability structure has complexity $O(mn)$ and can be computed in $O(mn \log mn)$ time.

Feasible Path in the Free Space Diagram

By Lemma 2 it suffices to consider homeomorphisms on the boundary curves of the simple polygons P and Q and extend these by mapping the diagonals of a triangulation of P to the corresponding shortest paths in Q. In other words for the decision problem we search for a homeomorphism $\sigma : P \to Q$ fulfilling:

(1) σ maps ∂P onto ∂Q and $\max_{t \in \partial P} ||t - \sigma(t)|| \leq \varepsilon$

(2) σ maps all diagonals of a triangulation to shortest paths in Q, and all diagonals D must fulfill $\max_{t \in D} ||t - \sigma(t)|| \leq \varepsilon$

For condition (1) we use the algorithm and data structure developed by Alt and Godau [2] for closed curves which searches for a monotone path in the double free space diagram. We handle condition (2) as follows: a path in the free space diagram determines the shortest paths that the diagonals are mapped to because it maps the end points of the diagonals. We call these *diagonal placements*. A path in the free space diagram which places all diagonals correctly, i.e., fulfills (2), we call *feasible path*.

Order of Diagonals in a Triangulation

The edge set of a triangulation T of P consists of edges on the boundary and in the interior of P, the latter of which are *diagonals*. We define an order of the diagonals which we will later use for dynamic programming over the diagonals. For a fixed starting point s on the boundary of P we order the diagonals as follows: If we write the diagonals as ordered tuples of their end points (d_i, d_j) with $i < j$, we define $(d_i, d_j) < (d_k, d_l) :\Leftrightarrow (j < l) \vee (j = l \wedge i > k)$.

For two starting points that lie in between the same two diagonal end points the resulting diagonal order is the same. Hence there are at most $n - 2$ different orders of diagonals in total, each characterized by the next diagonal endpoint (in counterclockwise order) to the starting point. We will call areas of the free space diagram that induce the same diagonal order *blocks*. Blocks consists of one or two columns in the diagram that lie between the vertical lines corresponding to two neighboring diagonal end points.

Combined Reachability Graph

The *combined reachability graph* combines the reachability information in the free space with valid diagonal placements. First we define a *reachability graph* which is the reachability structure represented as a graph: its vertices are the reachable intervals of the reachability structure with an edge between two intervals if they can reach each other. The combined reachability graph is a subgraph of the reachability graph. Its vertices are all vertical interval-vertices of the reachability graph with edges between intervals that can be reached by feasible paths. For a fixed order of diagonals the combined reachability graph can be computed by recursively merging the reachability graphs of the blocks in the order of diagonals. Since the reachability structure contains $O(mn)$ intervals the reachability graph and the combined reachability graph contain $O(mn)$ vertices and $O((mn)^2)$ edges.

We will use the combined reachability graph as follows: When searching for feasible paths starting in block B_1 we compute the combined reachability graph for the order of diagonals starting in B_1 by merging blocks B_2 through B_l (where l is the number of blocks). A feasible path starting in block B_1 consists of an edge in the reachability graph of B_1 from its lower to its right boundary, an edge in the combined reachability graph between the left boundary of B_2 to the right boundary of B_l, and an edge in the reachability graph of B_1 from its left to its upper boundary.

Fréchet Distance of a Diagonal and an Hourglass

The following lemma shows how to decide the Fréchet distance between a diagonal and a whole set of shortest paths, namely the hourglass of two segments. With a *shortest path in the hourglass* we always refer to a shortest path between two points on the two segments defining the hourglass.

Lemma 5 *Let an hourglass and a diagonal be given such that each end segment of the hourglass is contained in one of the ε-disks around the endpoints of the diagonal (and not both in the same disk). If there exists one shortest path in the hourglass with Fréchet distance at most ε to the diagonal, then all shortest paths in the hourglass have Fréchet distance at most ε to the diagonal.*

The idea of the proof of this lemma is the following: If $A = a_1, \ldots, a_l$ is a shortest path in the hourglass with Fréchet distance at most ε to the diagonal and $B = b_1 \ldots, b_k$ another shortest path in the hourglass. Define $B' = b_1, a_1, \ldots, a_l, b_k$. B' can be simplified to B by repeatedly using Lemma 1.

Fréchet Distances of a Diagonal and many Hourglasses

In Section 4 we need to decide all Fréchet distances between a diagonal and several hourglasses that have a common end segment. This can be done in $O(m)$ time by choosing an arbitrary vertex on each end segment of the hourglasses and using Lemma 5 and Lemma 6.

Lemma 6 *Given a diagonal, a polygon with m vertices, and a set of m vertices $w_1, \ldots, w_m$ on the boundary of the polygon. Then we can decide all Fréchet distances between the diagonal and the m shortest paths $\pi(w_1, w_i)$ between w_1 and w_i for $i = 1, \ldots, m$ in total $O(m)$ time.*

For proving this lemma we use the linear time algorithm for computing the lengths of all shortest paths from one vertex of a simple polygon to all others by Guibas et al. [6]. During the algorithm we store additional information about reachability in free space, which can be updated in constant time while processing each vertex.

Decision Algorithm

Algorithm 1: DecideFréchet(P, Q, ε)

```
Input: Simple Polygons P, Q, ε > 0
Output: Is δ_F(P, Q) ≤ ε?
Compute a triangulation of P
Compute all orders of diagonals in the triangulation of P
Compute a single free space diagram of the boundary curves
Compute the reachability graph for all blocks in the free space diagram
forall diagonals in the triangulation do
    forall placements in the free space do
        test δ_F(diagonal, shortest path) ≤ ε for a corresponding shortest path
    end
end
forall blocks do
    forall diagonals, in the order given by the block do
        if combined reachability graph is not yet computed then
            if previous diagonal nested then
                compute the combined reachability graph merged with the combined
                reachability graph of the nested diagonal
            end
            else
                compute the combined reachability graph
            end
            store the combined reachability graph
        end
        if diagonal has a left neighbor then
            merge the combined reachability graphs
        end
    end
    Query for a feasible path starting at the lower boundary of the block
end
Answer "yes" if a feasible path has been found, else "no"
```

Theorem 7 *Algorithm* DecideFréchet(P, Q, ε) *decides whether the Fréchet distance between simple polygons* P, Q *is at most* ε*. The runtime is* $O(nT(mn))$*, where* $T(N)$ *is the time to multiply two* $N \times N$ *matrices, and* n *and* m *are the number of vertices of* P *and* Q*.*

Note that $T(N) = \Omega(N^2)$ and the currently fastest known matrix multiplication algorithm has a runtime of $T(N) = O(N^{2.376})$ [4]. Due to space restrictions we have to completely omit the proof of this theorem.

Critical Values for Computation

For computing the Fréchet distance we apply the same technique as for curves [2]: We search a set of critical values using parametric search. Additionally to the critical values of the boundary curves we consider critical values for the Fréchet distance between diagonals and shortest paths.

A shortest path is always a polygonal curve where the first and last vertex are arbitrary points on the boundary of Q and all other vertices are vertices of Q. The distances between the diagonal end points and the boundary of Q are already contained in the critical values for the boundary curves. Additional critical values can only occur if the Fréchet distance between a diagonal and a shortest path is attained in the interior of the diagonal and the shortest path, i.e., it is attained at a vertex of Q. For the parametric search we sort the "spikes" in the free space diagram. We get $m \cdot n$ such spikes, one for any pair of a diagonal and a vertex of Q. In total we get:

Theorem 8 *The Fréchet distance between two simple polygons can be computed in time* $O(nT(mn)\log(mn))$*, where* $T(N)$ *is the time to multiply two* $N \times N$ *matrices, and* n *and* m *are the number of vertices on the boundary of* P *and* Q*.*

Acknowledgements

We thank Boris Aronov for suggesting the problem, and we thank Peter Brass and Helmut Alt for fruitful discussions.

References

[1] H. Alt and M. Buchin. Semi-computability of the Fréchet distance between surfaces. In *Proc. 21st Europ. Workshop on Comp. Geom.*, pages 45–48, 2005.

[2] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.

[3] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[4] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.

[5] M. Godau. *On the complexity of measuring the similarity between geometric objects in higher dimensions.* PhD thesis, Freie Universität Berlin, Germany, 1998.

[6] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear time algorithms for visibility and shortest path problems inside simple polygons. In *Proc. 2nd Ann. Symp. on Comput. Geom.*, pages 1–13, New York, NY, USA, 1986. ACM Press.