Chapter 5

Computability of the Fréchet Distance between Simple Polygons

5.1 Introduction

In this chapter we consider a restricted but important class of surfaces, *simple polygons*, which appear often in applications. For these we show that the set of realizing reparameterizations can be restricted to a set of "nice" maps. This allows us to show that the Fréchet distance between simple polygons can be computed in polynomial time. The results in this chapter where obtained in collaboration with Kevin Buchin and Carola Wenk and have been published in [16].

This chapter is organized as follows: Section 5.2 introduces simple polygons and their Fréchet distance. In Section 5.3 we show that for computing the Fréchet distance between simple polygons, we need to consider only homeomorphisms on the boundary curves which are extended to the diagonals of a convex decomposition of one polygon by mapping the diagonals to shortest paths in the other polygon. In Section 5.4 we use this result to show how to compute the Fréchet distance between simple polygons in polynomial time. For this, we will employ the algorithm for curves, techniques for shortest path in a simple polygon, and dynamic programming.

5.2 Simple Polygons

5.2.1 Fréchet Distance between Simple Polygons

A simple polygon is the area enclosed by a non-self-intersecting closed polygonal curve in a plane. Let P and Q be two simple polygons with n and m vertices, respectively. The two polygons may lie in two different planes. As underlying parameterizations we assume the identity maps $f\colon P \to P$ and $g\colon Q \to Q$. Then the Fréchet distance simplifies to:

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

where σ ranges over all orientation-preserving homeomorphisms.

In the following we assume to be given an orientation of the simple polygons in the form of an ordering of the vertices. For simple polygons, preserving their orientation is equivalent to preserving the orientation on the boundary. We will consider only orientation-preserving homeomorphisms and may refer only to σ or to a homeomorphism if the meaning is clear from the context.

We call a map $\sigma\colon P \to Q$ for which $\max_{t\in P} \|t - \sigma(t)\| = \varepsilon$ an ε-*realizing map*. In particular, we will consider ε-*realizing homeomorphisms*. Recall that the Fréchet distance of simple polygons is defined as the infimum over all homeomorphisms on the polygons. Using the notion of ε-realizing homeomorphism we can reformulate the decision problem for the Fréchet distance as

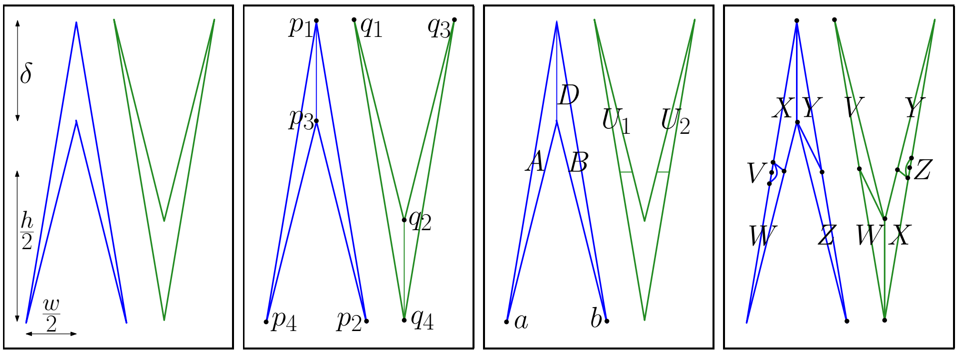
$$\delta_F(P,Q) \le \varepsilon \;\Leftrightarrow\; \text{for all } \varepsilon' > \varepsilon \text{ exists an } \varepsilon'\text{–realizing homeomorphism.}$$

In particular, for a Fréchet distance equal to ε, an ε-realizing homeomorphism need not exist. Instead, a sequence of ε_i-realizing homeomorphisms exists with ε_i tending to ε for i tending to ∞. We call such a sequence an ε-*realizing sequence* for the Fréchet distance.

5.2.2 Fréchet Distance between Boundary Curves

A natural question concerning the Fréchet distance between simple polygons is: does the Fréchet distance between the polygons differ from the Fréchet distance between their boundary curves?

Proposition 5.1. *The Fréchet distance between two simple polygons may be arbitrarily much larger than the Fréchet distance between their boundary curves.*



(a) Polygons with large Fréchet distance but small Fréchet distance between the boundary curves

(b) Assignment of the vertices for a map realizing the Fréchet distance between the boundary curves

(c) Illustration for the proof of the Fréchet distance between the polygons

(d) Illustration of a map realizing the Fréchet distance between the polygons

Figure 5.1: Example for differing Fréchet distance between boundary curves and polygons.

Proof. Figure 5.1 (a) shows two polygons for which the Fréchet distance between the boundary curves may be arbitrarily much smaller than the Fréchet distance between the polygons. We show that, if the two polygons are placed on top of each other and the distances δ and w are both infinitesimally small, then the Fréchet distance between the boundary curves is zero whereas the Fréchet distance between the polygons is at least half the height of the polygons.

Figure 5.1 (b) indicates a homeomorphism which realizes the Fréchet distance between the boundary curves, assuming that both w and δ are much smaller than

h and the two polygons are placed on top of each other such that their bounding boxes coincide. In the figure, p_i is mapped to q_i for all i. The Fréchet distance equals the maximal point-to-point distance, which is the distance between p_3 and q_3 in Figure 5.1 (b). For $\delta \to 0$ the distance $\|p_3 - q_3\|$ converges to $w/2$, if also $w \to 0$ the Fréchet distance becomes zero.

Figure 5.1 (c) illustrates that the Fréchet distance between the polygons cannot be smaller than $h/2$ provided that $4\delta < h$. Consider the diagonal D in the left polygon. For the Fréchet distance to be less than half the height of the polygons, D must be mapped to a path that lies completely either in U_1 or U_2. Then also either A or B must be completely mapped to U_1 or U_2, in particular the vertex a or b. But both a and b have a distance more than $h/2$ to both U_1 and U_2. □

In the proof of Proposition 5.1 we showed that the Fréchet distance between the polygons in Figure 5.1 is at least $h/2$. It, in fact, equals $h/2$ if as above both w and δ tend to zero, and the two polygons are placed on top of each other such that their bounding boxes coincide. This is indicated in Figure 5.1 (d), which shows isomorphic decompositions of P and Q. The Fréchet distance of $h/2$ can be realized by a homeomorphism mapping regions with the same label onto each other. The reader may verify that such a mapping can be realized by a homeomorphism with maximum point-to-point distance $h/2$, for w and δ tending to zero.

5.2.3 Fréchet Distance between Convex Polygons

For the special case of convex polygons Proposition 5.2 below states that the Fréchet distance between the polygons equals the Fréchet distance between their boundary curves. Boundary curves of convex polygons are closed convex curves and for these it is known that the Fréchet distance equals the Hausdorff distance [3, 9, 29]. Thus Proposition 5.2 implies that the Fréchet distance between convex polygons equals the Hausdorff distance between their boundary curves, which can be computed in polynomial time. In two dimensions the Hausdorff distance between convex polygons can be computed in $O((m+n)\log(m+n))$ time [2] and for disjoint convex polygons or convex polygons where one contains the other in $O(m+n)$ time [12].

Proposition 5.2 will follow also from Corollary 5.1, see Section 5.3. The proof we give now can be generalized to convex polytopes in arbitrary dimension.

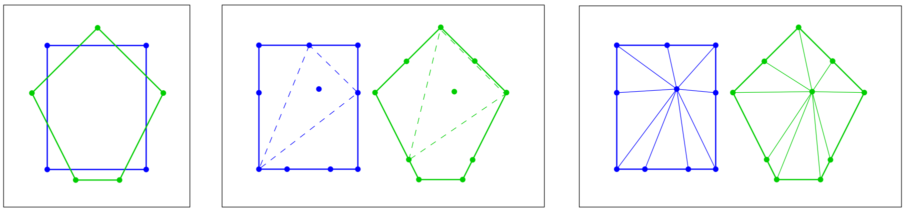
Proposition 5.2. *The Fréchet distance between convex polygons equals the Fréchet distance between their boundary curves.*

Proof. The Fréchet distance between two polygons is at least as large as the Fréchet distance between their boundary curves, since a homeomorphism on the polygons restricted to the boundary curves is again a homeomorphism. We show that it is also not larger by showing that for any homeomorphism σ on the boundary curves there exists a homeomorphism σ' between the polygons such that $\max\|t-\sigma'(t)\| \leq \max\|t-\sigma(t)\|$.

Let P, Q be two convex polygons, and $\sigma : \partial P \to \partial Q$ an arbitrary homeomorphism on the polygon boundaries. We construct a piecewise linear homeomorphism $\sigma' : P \to Q$ on the polygons which equals σ on all boundary vertices of P and Q and attains its maximum on one of these vertices. The construction of σ' is illustrated in Figure 5.2.

First we add to the boundary vertices of P and Q all (inverse) images under σ of the boundary vertices of Q and P, respectively. I.e., we add the vertex $\sigma^{-1}(q)$ to the boundary of P for all boundary vertices q of Q and $\sigma(p)$ to the boundary of Q for all boundary vertices p of P.

Next we choose a point p_0 in the interior of P and a point q_0 in the interior of Q. We choose p_0 and q_0 s.t. their distance is bounded by a distance $\|p-q\|$ where p is



(a) convex polygons (b) interior vertices (c) triangulations

Figure 5.2: The Fréchet distance of convex polygons is attained on the boundary curves.

a boundary vertex of P and $q = \sigma(p)$ a boundary vertex of Q. For this, we choose p_0 and q_0 as the barycenters of two triangles (p_1, p_2, p_3) and (q_1, q_2, q_3) where the p_i and q_i are boundary vertices of P and Q, respectively, for $i = 1, 2, 3$.

If both P and Q are not triangles, i.e., both have at least four boundary edges, we can choose triangles (p_1, p_2, p_3) and (q_1, q_2, q_3) as illustrated in Figure 5.2 (b). We choose two boundary vertices p_1 and p_2 of P which do not share a boundary edge of P. Let $q_1 = \sigma(p_1)$ and $q_2 = \sigma(p_2)$. Then we choose a boundary vertex q_3 of Q such that q_1, q_2, q_3 are not collinear. Let $p_3 = \sigma^{-1}(q_3)$. Then also p_1, p_2, p_3 are not collinear because p_1 and p_2 were chosen such that they do not share a boundary edge of P. The barycenters of the triangles p_1, p_2, p_3 and q_1, q_2, q_3 lie inside P and Q, respectively, because of the convexity of P and Q. Furthermore, the distance between the barycenters is bounded by the distances between the vertices of the triangles.

If P is the triangle (p_1, p_2, p_3), then we choose these vertices and the vertices $q_1 = \sigma(p_1), q_2 = \sigma(p_2)$, and $q_3 = \sigma(p_3)$ in Q. If q_1, q_2, q_3 are not collinear then we can proceed as before. If they are collinear, we assume without loss of generality that q_2 lies between q_1 and q_3 on the line. This implies that the edges (p_1, p_2) and (p_2, p_3) are mapped by σ to the same boundary segment of Q and the edge $e = (p_1, p_3)$ is mapped to the rest of the boundary of Q. We choose a boundary vertex q_3' of Q which does not lie on the same line, and therefore it lies on $\sigma(e)$. We replace q_3 by q_3' and p_3 by $p_3' = \sigma^{-1}(q_3')$. Then, because p_3' lies on the edge $e = (p_1, p_3)$, also p_1, p_2, p_3' are not collinear.

Now we triangulate both P and Q into a "circular fan" by adding for all vertices v on the boundary of P the diagonal (v, p_0) and for all vertices w on the boundary of Q the diagonal (w, q_0). See Figure 5.2 (c) for an example. This yields isomorphic triangulations of P and Q with the property that σ is an isomorphism on the vertices. Thus, we can define σ' to be the piecewise linear homeomorphism between these two triangulations. □

5.2.4 Shortest Paths in a Simple Polygon

Our algorithm for computing the Fréchet distance between simple polygons involves shortest paths. We therefore review an important concept for shortest paths in a simple polygon which was introduced by Guibas et al. [30]: *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 p_1 on S_1 and any point p_2 on S_2. It can be described by the (possibly degenerate) polygon given by the two segments and the two shortest paths between neighboring endpoints of the segments (i.e., if a_1, a_2 and b_1, b_2 are the endpoints of S_1 and S_2, respectively, and their order along the boundary of the polygon is a_1, a_2, b_1, b_2, then the hourglass is the polygon with boundary S_1, the shortest path between a_2 and b_1, S_2, and the shortest path between b_2 and a_1). See Figure 5.3 for examples.

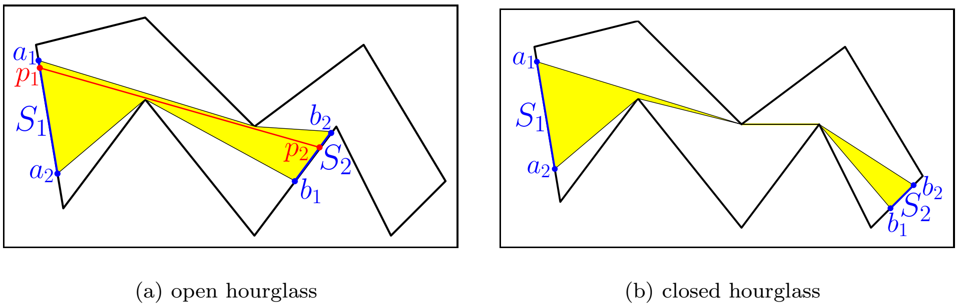


Figure 5.3: Hourglasses in a simple polygon.

There are two kinds of hourglasses: *open* and *closed* as depicted in Figure 5.3. An hourglass is called *open* if there are two points p_1 on S_1 and p_2 on S_2 that directly see each other, i.e., the shortest path from p_1 to p_2 is the segment (p_1, p_2). Otherwise the hourglass is called *closed*. In this case the segments S_1 and S_2 are mutually invisible, i.e., there are no two points on S_1 and S_2 that directly see each other.

We will use hourglasses which are not given by two segments but more generally by two chains of consecutive segments. That is, instead of two segments S_1, S_2 we are given two polygonal chains C_1, C_2 where each C_i, $i = 1, 2$, consists of consecutive edge segments. We call these chains the *end chains* of the *generalized hourglass* . This generalization is straightforward and the notions open and closed of hourglasses remain the same.

5.2.5 Convex Decompositions of Simple Polygons

For computing the Fréchet distance between simple polygons we will decompose the polygons into convex parts. We will decompose the polygons without allowing additional vertices, i.e., we allow diagonals only between existing vertices. For achieving a better run time we will use an (approximate) *minimum convex decomposition*. A minimum convex decomposition is a convex decomposition with a minimal number of components.

A minimum convex decomposition without additional vertices can be computed in $O(n+r^2 \min(n, r^2))$ time [35], where n is the number of vertices of the polygon and r the number of reflex vertices. A reflex vertex is a vertex with interior angle larger than π. For computing a constant factor approximation of the minimum convex decomposition several $O(n \log n)$ time algorithms exist. See the survey of Keil [34] for more details and references. In our algorithm we will compute an approximate minimum convex decomposition of each polygon and choose the smaller one.

Note that it is also possible to use a triangulation instead of a minimum convex decomposition. However, since the combinatorial complexity of a triangulation is typically much larger than that of a minimum convex decomposition, this would increase the run time of our algorithm.

5.3 Restricting the Set of Homeomorphisms

In this section we show that for the Fréchet distance between simple polygons P, Q it suffices to consider a small well-behaved class of realizing maps. These are homeomorphisms on the boundaries of P and Q which are extended to the diagonals of a convex decomposition of P by mapping the diagonals of the decomposition to shortest paths in Q. Each diagonal is mapped homeomorphically to the shortest path between the images of its endpoints. We call these maps *shortest path maps*. If the homeomorphism on the boundaries is orientation preserving, we call the shortest path map orientation preserving.

For a convex decomposition C of P we denote with E_C the set of all points lying on some edge of C. The edges of C are the boundary edges and diagonals of P. Thus, a shortest path map is a non-surjective map $\sigma' : E_C \to Q$. If the shortest paths in Q overlap (with each other or the boundary), then a shortest paths map is not a homeomorphism. However, restricted to the boundary or to any diagonal of C a shortest path map is a homeomorphism.

A homeomorphism on the polygons can be restricted to the polygon boundary which yields a homeomorphism between the polygon boundaries. We can obtain a shortest path map from a homeomorphism on the polygons by restricting it to the boundary and then extending it to the diagonals of a convex decomposition by mapping these to shortest paths. We call these *induced shortest path maps*.

First, we show a preliminary result, Lemma 5.1, which states that simplifying a curve by replacing a part of the curve by a segment does not increase its Fréchet distance to another curve. Then, we show that for any homeomorphism σ on the polygons the induced shortest path map σ' realizes a value for the Fréchet distance which is not larger than the one realized by σ. This will follow from Lemma 5.2. Next we show in Lemma 5.3 that for a given shortest path map σ' there are homeomorphisms on the polygons that realize a Fréchet distance arbitrarily close to the value realized by σ'. Combining these two results, we see in Proposition 5.3 that for the Fréchet distance between simple polygons it suffices to consider shortest paths maps.

5.3.1 Simplifying a Curve

Given a curve f and a line segment s, Lemma 5.1 shows that simplifying the curve f by replacing a part of it with a line segment does not increase its Fréchet distance to the line segment s. See Figure 5.4 for an illustration.

Lemma 5.1. *Let $f : [0,1] \to \mathbb{R}^d$ be a curve, let $s : [0,1] \to \mathbb{R}^d$ be a parameterized line segment, and let $0 \leq t_1 < t_2 \leq 1$. Define $f' : [0,1] \to \mathbb{R}^d$ to coincide with f on $[0, t_1] \cup [t_2, 1]$ and to coincide with a parameterized line segment from $f(t_1)$ to $f(t_2)$ on $[t_1, t_2]$. Then $\delta_F(f', s) \leq \delta_F(f, s)$.*

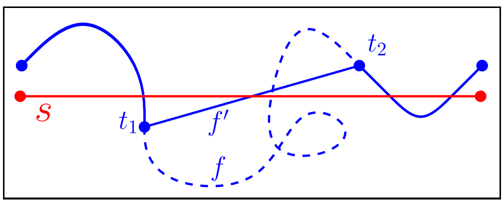


Figure 5.4: The Fréchet distance between a diagonal and a curve is not increased if the curve is simplified by substituting part of it by a line segment.

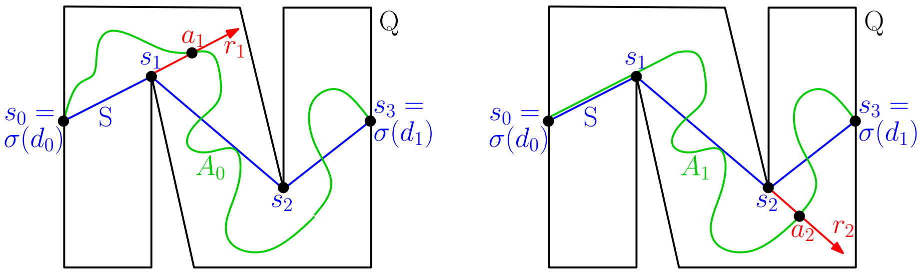


Figure 5.5: We recursively simplify the curve $A_0 = \sigma(D)$ to the polygonal curve S which is the shortest path from $\sigma(d_0)$ to $\sigma(d_1)$ in Q.

Proof. Given a homeomorphism $\sigma\colon [0,1] \to [0,1]$, let $\sigma'\colon [0,1] \to [0,1]$ be the homeomorphism that equals σ on $[0,t_1] \cup [t_2,1]$ and maps (t_1,t_2) linearly to $(\sigma(t_1),\sigma(t_2))$. Then it holds

$$\begin{aligned} \max_{t\in[0,1]} \|f(t) - s(\sigma(t))\| &\geq \max\nolimits_{t\in[0,t_1]\cup[t_2,1]} \|f'(t) - s(\sigma'(t))\| \\ &= \max\nolimits_{t\in[0,1]} \|f'(t) - s(\sigma'(t))\|. \end{aligned}$$

The first inequality holds because f and f' as well as σ and σ' coincide on $[0,t_1] \cup [t_2,1]$. The following equality holds because on the missing interval (t_1,t_2) we are taking the maximum distance between two parameterized segments, which is attained at the segment endpoints.

And thus, for a sequence of homeomorphisms σ_i realizing $\delta_F(f,s)$, the sequence of homeomorphisms σ_i' yields $\delta_F(f',s) \leq \delta_F(f,s)$. □

5.3.2 Mapping Diagonals to Shortest Paths

Lemma 5.2. *Given two simple polygons P and Q, a diagonal D of P and a homeomorphism $\sigma\colon P \to Q$. Let $\sigma'\colon P \to Q$ map the diagonal D homeomorphically to the shortest path between the images of its endpoints under σ. Then*

$$\delta_F(D,\sigma'(D)) \leq \delta_F(D,\sigma(D)).$$

Proof. Let d_0, d_1 be the starting and endpoint of the diagonal D in P. Let $A = \sigma(D)$ be the curve which is the image of D under σ and let S be the shortest path in Q between $\sigma(d_0)$ and $\sigma(d_1)$. We want to show

$$\delta_F(D,S) \leq \delta_F(D,A).$$

The shortest path S is a polygonal path in Q with starting point $\sigma(d_0)$ and endpoint $\sigma(d_1)$. We denote the vertices of this polygonal path by $s_0,\ldots,s_l$, where $s_0 = \sigma'(d_0) = \sigma(d_0)$, $s_l = \sigma'(d_1) = \sigma(d_1)$, and l is the number of edges of the polygonal path as shown in Figure 5.5.

We iteratively shoot rays along the edges of the shortest path and simplify the curve A using Lemma 5.1 as follows: Let $A_0 = A = \sigma(D)$. For each $i = 1,\ldots,l$ we do the following (cf. Figure 5.5): Let r_i be the ray in direction $s_i - s_{i-1}$ starting at s_{i-1}. By construction s_{i-1} lies on A_{i-1}. The ray r_i cuts the polygon into two parts such that the points s_0 and s_l lie in different parts. Hence, the curve A_{i-1}, which is a continuous curve from s_0 to s_l, intersects r_i inside the polygon. Let a_i be the first intersection of r_i with A_{i-1}. Define A_i to be A_{i-1} simplified by exchanging the part of A_{i-1} from s_{i-1} to a_i with the line segment (s_{i-1},a_i). By Lemma 5.1, $\delta_F(D,A_i) \leq \delta_F(D,A_{i-1})$. Note that s_i lies on the line segment (s_{i-1},a_i). Starting with $A_0 = A = \sigma(D)$ we end with $A_l = S = \sigma'(D)$ and therefore the iteration shows $\delta_F(D,S) \leq \delta_F(D,A)$. □

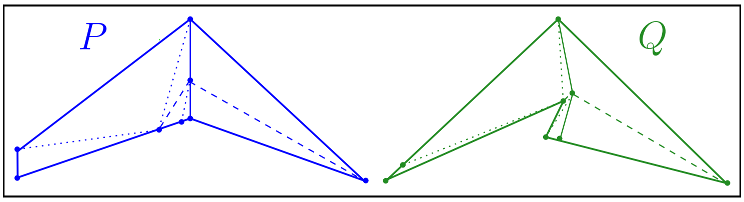


Figure 5.6: Constructing isomorphic triangulations of P and Q. The initial decomposition is shown in solid lines and is initially refined by the dashed edges and furthermore refined by the dotted edges.

5.3.3 Approximating Shortest Path Maps

Lemma 5.3. *Given two simple polygons P and Q, a convex decomposition C of P and a shortest path map $\sigma'\colon E_C \to Q$. Then for all $\delta > 0$ there exists a homeomorphism $\sigma_\delta\colon P \to Q$ that realizes a Fréchet distance not larger than δ plus the Fréchet distance realized by σ'. That is,*

$$\max_{t\in P} \|t - \sigma_\delta(t)\| \;\le\; \max_{t\in E_C} \|t - \sigma'(t)\| + \delta.$$

Proof. Given a shortest path map σ' and $\delta > 0$, we construct a piecewise linear homeomorphism $\sigma_\delta\colon P \to Q$ fulfilling the claim of the lemma. We construct σ_δ by constructing isomorphic triangulations of P and Q. On the vertices of the triangulations we construct σ_δ as a graph isomorphism and then extend σ_δ piecewise linear inside triangles. The main technical part of the proof is constructing the isomorphic triangulations which we defer to the end of the proof. First, we give the proof without explicit construction of the triangulations.

The shortest path map induces isomorphic decompositions of the polygons P and Q (after perhaps slight perturbation of vertices). Namely, these are the given convex decomposition C of P and the decomposition of Q induced by the shortest paths to which the diagonals in P are mapped. We refine these decompositions to isomorphic triangulations of P and Q by introducing additional vertices and diagonals. While constructing the isomorphic triangulations, we construct σ_δ to be an isomorphism on the vertices. Furthermore, σ_δ equals σ' on these vertices possibly after slight perturbation. We can then extend σ_δ piecewise linearly inside triangles to obtain a homeomorphism on the polygons. By the piecewise linearity, σ_δ achieves its maximum value for the Fréchet distance at a vertex. Since it equals σ' on vertices, the claim of the lemma follows.

During the construction of the isomorphic triangulations, we slightly perturb vertices. This always means that we move the vertices a distance of at most δ. This ensures that the value achieved for the Fréchet distance is increased by at most δ as claimed. Note that the sequence of homeomorphisms σ_δ that we construct in this way are an ε-realizing sequence for the Fréchet distance, where $\varepsilon = \max_{t\in E_C} \|t - \sigma'(t)\|$, i.e., the value for the Fréchet distance achieved by the shortest path map σ'.

Constructing Isomorphic Triangulations of P and Q We construct isomorphic triangulations of P and Q in three steps (cf. Figure 5.6). First, we define initial isomorphic decompositions of P and Q. These decompositions are refined twice to obtain isomorphic triangulations. In the first refinement, we refine the decomposition of Q to a triangulation and in the second we refine the decomposition of P to a triangulation.

During the construction we construct σ_δ as an isomorphism on the vertices of the triangulations. For this, we let σ_δ equal σ' on the boundary vertices of P and Q, that is, $\sigma_\delta(p) = \sigma'(p)$ for all boundary vertices p of P and for all inverse images

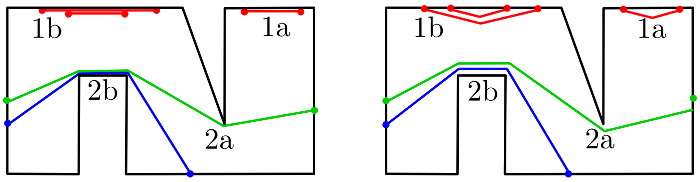


Figure 5.7: Degeneracies that may occur in the initial decomposition of Q (left) and their perturbation (right).

$p = \sigma'^{-1}(q)$ of boundary vertices q of Q. During the construction, we refine the isomorphic decompositions by adding new edges (and their endpoints). Whenever we add an edge e_P in P and corresponding edge e_Q in Q, we always let σ_δ map the corresponding endpoints onto each other, i.e., $\sigma_\delta(p) = q$ for corresponding endpoints p of e_P and q of e_Q.

Initial Decompositions. The initial decomposition of P is the convex decomposition C. The initial decomposition of Q is obtained by connecting for each diagonal in C the images of its endpoints under σ in Q by a shortest path. If all shortest paths in Q are diagonals, i.e., line segments between different boundary edges of Q, then these are isomorphic decompositions of P and Q. However, two kinds of degeneracies may occur (cf. Figure 5.7 left): In the resulting decomposition of Q, a shortest path may lie completely on the boundary or it may consist of more than one edge. In the first case, a face in the decomposition is degenerated to a line segment (1). The second case causes two faces to be connected either only by a vertex (2a) or an edge (2b). Also, several shortest paths may share an inner vertex or edge chain, as in the cases 2b and 1b in Figure 5.7.

For both kinds of degeneracies we can add and perturb vertices (cf. Figure 5.7 right) to obtain isomorphic decompositions. If a shortest path lies completely on the boundary, we add the midpoints of the diagonal in P and on the shortest path in Q. The midpoint of the shortest path is moved slightly into the interior of the polygon which results in a triangular face (1a in Figure 5.7). If several shortest paths lie on top of each other on the boundary (1b in Figure 5.7) the longer diagonals are perturbed more than the shorter diagonals.

If a shortest path consists of more than one edge, we add each inner vertex of the shortest path as a new vertex, both on the shortest path in Q and on the diagonal in P. On the diagonal in P we add the vertices $\tau^{-1}(v)$ where v is an inner vertex of the shortest path and τ is an ε-realizing homeomorphism for the diagonal and shortest path and ε is the value for the Fréchet distance achieved by σ'. If only ε-realizing sequences exist, we slightly perturb the construction by using an ε'-realizing homeomorphism for $\varepsilon < \varepsilon' \leq \varepsilon + \delta$. We also move the inner vertices of the shortest path slightly into the interior of the polygon. If several shortest paths share a vertex (see Figure 5.7 (2b)), we create a new vertex for each shortest path, and move these slightly into the interior in an order which ensures that the shortest paths do not cross. This is possible since the corresponding diagonals in P are non-crossing. Now we have isomorphic decompositions of P and Q.

Idea of Refinements. The idea of refining the isomorphic decompositions to isomorphic triangulations is the following: One of the initial isomorphic decompositions is convex, namely the one of P. We first triangulate the possibly non-convex decomposition of Q and insert corresponding diagonals in P. Then we have a (nearly) convex decomposition of P and a (nearly) triangulation of Q (both only nearly because of perturbation in case of degeneracies). Then we triangulate the convex decomposition of P and insert corresponding diagonals in Q to obtain isomorphic triangulations (again we have to handle degenerate cases).

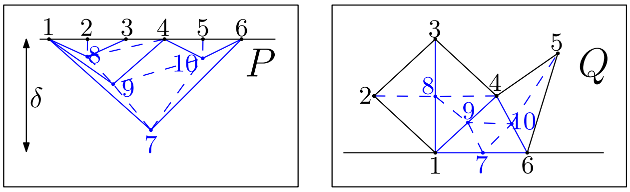


Figure 5.8: Handling degeneracies in the first refinement. Vertex labels indicate the isomorphism on the vertices.

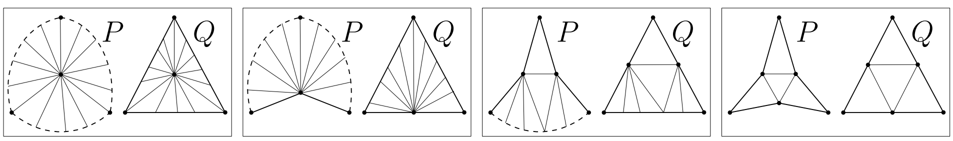
In the following, we give these refinements in more detail. The main effort is handling the degeneracies. We give one (of many) possible solution for this.

First Refinement. We first triangulate the initial decomposition of Q. For all diagonals (q, q') that we add to Q for this, we also want to add the segment $(\sigma'^{-1}(q), \sigma'^{-1}(q'))$ in P. However, a segment $(\sigma'^{-1}(q), \sigma'^{-1}(q'))$ may lie completely on the boundary of P. If it does not, we can add the segment as diagonal to P. If it does, we need to perturb the segment as follows.

Let $(\sigma'^{-1}(q), \sigma'^{-1}(q'))$ be a segment that lies on the boundary of P. There may be several such segments on top of each other, i.e., the corresponding diagonals are mapped to the same boundary segment of P. In this case, we perturb the set of segments simultaneously, as illustrated in Figure 5.8. We add the midpoints to all these segments in P and the midpoints to the corresponding diagonals in Q. Then we move the midpoints of the segments in P slightly into the interior. The midpoints of longer segments are moved further than those of shorter segments. We will call these *bent diagonals* in P. Next, we isomorphically triangulate both constructions by connecting midpoints and boundary vertices as in Figure 5.8.

The resulting decomposition of P is nearly convex in the sense that it may contain bent diagonals. The resulting decomposition of Q is nearly a triangulation in the sense that some edges contain extra midpoints. Furthermore, these two cases exactly coincide, i.e., each bent diagonal in P exactly corresponds to an edge with an additional midpoint in Q.

Second Refinement. Now we triangulate the remaining faces of P which are not yet triangulated. Because each bent edge in P corresponds exactly to a midpoint edge in Q, there are only four possible cases: zero, one, two, or three of the edges are irregular. In each case, we can isomorphically triangulate the two faces without introducing new degeneracies (cf. Figure 5.9) . If all sides are regular, we add the midpoints of the triangle in Q and the corresponding three points in P and triangulate into a star (Figure 5.9 (a)). If one side is irregular, we triangulate from the single midpoints (Figure 5.9 (b)). If two sides are irregular, we connect the two midpoints and triangulate the rest (Figure 5.9 (c)). If all three sides are irregular, we connect the three midpoints into a triangle (Figure 5.9 (d)). □



(a) (b) (c) (d)

Figure 5.9: All cases that may occur in the second refinement. Dashed lines indicate convex edge chains.

5.3.4 Fréchet distance using Shortest Paths Maps

Lemmas 5.2 and 5.3 imply the following proposition.

Proposition 5.3. *The Fréchet distance between simple polygons P and Q equals*

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

where C is an arbitrary convex decomposition of P. The map σ' ranges over all orientation-preserving shortest path maps σ' from E_C to Q.

Proof. Lemmas 5.2 and 5.3 together give the following equivalence.

$$\begin{aligned} \delta_F(P,Q) \le \varepsilon \quad &\Leftrightarrow \quad \text{for all } \varepsilon' > \varepsilon \text{ exists an } \varepsilon'\text{−realizing homeomorphism} \\ &\Leftrightarrow \quad \text{for all } \varepsilon' > \varepsilon \text{ exists an } \varepsilon'\text{−realizing shortest path map.} \end{aligned}$$

The first equivalence holds by definition of the Fréchet distance, see Section 5.2.1.Now consider the direction from left to right of the second equivalence. Let σ be an ε'-realizing homeomorphism. Using Lemma 5.2 the induced shortest paths map realizes a Fréchet distance of at most ε', as well. The other direction of the second equivalence follows from Lemma 5.3. □

A convex decomposition of a convex polygon is the polygon itself. Therefore a shortest path map on a convex polygon is a homeomorphism on the boundary curves. Thus, for the Fréchet distance of two polygons where one is convex, it suffices, by Proposition 5.3, to map the boundary of the convex polygon to the boundary of the other polygon, i.e., to compute the Fréchet distance of the boundary curves.

Corollary 5.1. *The Fréchet distance between two simple polygons, of which one polygon is convex, equals the Fréchet distance between their boundary curves.*

5.4 Deciding the Fréchet Distance

In this section we give a polynomial time algorithm for deciding the Fréchet distance between simple polygons. For this, we first characterize in Section 5.4.1 which paths in the free space diagram correspond to solutions of the Fréchet distance between simple polygons. In Section 5.4.2 we describe the *combined reachability graph.* This graph extends the reachability structure for closed polygonal curves, reviewed in Section 2.3.2, by adding the information of diagonals in one polygon mapped to shortest paths in the other polygon.

As a subproblem in the algorithm we need to solve the decision problem for the Fréchet distance between a diagonal in one polygon and all shortest paths between two consecutive chains of boundary segments of the other polygon. These shortest paths make up the generalized hourglass of the two chains, which was introduced in Section 5.2.4. We show how to solve the decision problem for the Fréchet distance between a diagonal and one hourglass in Section 5.4.3 and for a diagonal and multiple hourglasses at once in Section 5.4.4.

In the following P and Q always denote two simple polygons with n and m vertices, respectively, and ε a real value greater than or equal to zero. C denotes a convex decomposition of P. The decision problem for the Fréchet distance is to decide whether $\delta_F(P,Q) \le \varepsilon$. We will consider the free space diagram of a diagonal and a shortest path. Such a free space diagram consists of $1 \times k$ cells where k is the number of edges of the shortest path. We will always assume in this case that the free space diagram is a column of cells, i.e., the diagonal corresponds to the bottom boundary of the diagram and the shortest path corresponds to the left boundary of the diagram.

5.4.1 Feasible Path in the Free Space Diagram

By Proposition 5.3, the Fréchet distance between simple polygons is realized by an ε-realizing sequence of shortest path maps. That is, an ε-realizing sequence of homeomorphisms on the boundary curves which are extended to the diagonals of a convex decomposition by mapping these to shortest paths. We can find ε-realizing sequences of homeomorphisms on the boundary curves by searching for monotone paths in the double free space diagram [6]. For a monotone path we can check whether the corresponding homeomorphism (or limit of homeomorphisms) maps the diagonals to shortest paths with Fréchet distance at most ε. If it does we call it a *feasible path* in the free space.

We check the condition on the diagonals for a monotone path as follows. A strictly monotone path in the free space diagram F_ε corresponds to a homeomorphism on the boundary curves. Thus, we know where the diagonal endpoints are mapped to and can decide if the Fréchet distances between diagonals and corresponding shortest paths are at most ε. A monotone path may, however, contain horizontal and vertical segments. If a vertical segment corresponds to a diagonal endpoint, then the monotone path does not map this endpoint to a unique point but it maps the diagonal endpoint to a connected chain of segments on the boundary of Q instead. For any $\varepsilon' > \varepsilon$ there are strictly monotone paths in the free space arbitrary close to the monotone path. For these, the vertical segment is slightly tilted which uniquely maps the diagonal endpoint. We will see in Section 5.4.3, that for the Fréchet distance between a diagonal and a shortest path it does not matter which points on a vertical segment in the free space are chosen as endpoints of the shortest paths. Therefore, in the case of vertical segments corresponding to diagonal endpoints, we can choose arbitrary points on the segments as endpoints of the shortest paths.

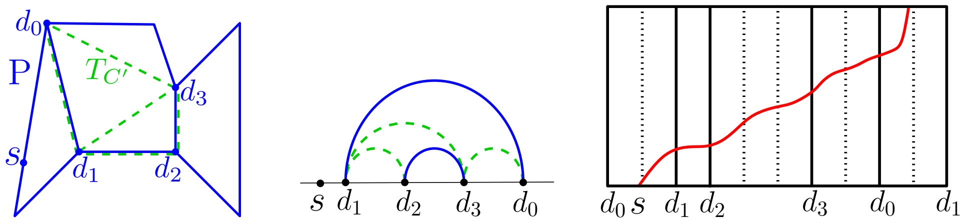
The above considerations and Proposition 5.3 yield the following corollary.

Corollary 5.2. *The Fréchet distance between simple polygons is less than or equal ε if and only if there is a feasible path in the free space diagram F_ε.*

Searching for such a path seems difficult at first because there typically are infinitely many monotone paths realizing the Fréchet distance of the boundary curves. Of course we cannot check for infinitely many paths whether they map the diagonals to shortest paths with Fréchet distance less than or equal ε. Again we use the result of Section 5.4.3, by which it suffices to specify for each diagonal endpoint not the exact point it is mapped to but only the segment it is mapped to in the following sense.

For each point p on the boundary of P, i.e., in particular for each diagonal endpoint, consider a pre-image of the point p in parameter space. Consider the vertical line in free space corresponding to this pre-image. On this line there are at most m intervals which lie in the free space, because each row can contribute at most one interval. We call these intervals *free intervals*. Note that such intervals may span several rows.

In Section 5.4.3 we will show that the solution of the decision problem of the Fréchet distance between a diagonal and a shortest path depends only on the free intervals to which the diagonal endpoints are mapped. We call an assignment of the diagonal endpoints to free intervals a *placement* of the diagonals. As stated before, there are at most m placements for each diagonal endpoint. We call a placement of a diagonal *valid* if the Fréchet distance between the diagonal and the shortest path given by the placement is at most ε.



(a) convex decomposition of P

(b) diagonals in the order visited from s

(c) path in free space starting at s

Figure 5.10: Diagonals in a convex deomposition are mapped to shortest paths by a monotone path in the free space.

5.4.2 Combined Reachability Graph

The *combined reachability graph* combines the reachability information in the free space with valid diagonal placements and thus allows us to search for feasible paths. First, we define the *reachability graph* to be the reachability structure (see Section 2.3.2) represented as a graph: its vertices are the reachable intervals of the reachability structure with an edge between two intervals if one can reach the other. The combined reachability graph is a subgraph of the reachability graph. Its vertices are also the reachable intervals of the reachability structure. Its edges are between intervals that can be reached by feasible paths (cf. Section 5.4.1). Since the reachability structure contains $O(mn)$ intervals, both the reachability graph and the combined reachability graph contain $O(mn)$ vertices and $O((mn)^2)$ edges.

Let $d_0, d_1, \ldots, d_{l-1}$ be the l endpoints of the k diagonals of a convex decomposition C of P where $l \leq 2k$. In the following we use $+_l$ to denote addition modulo l and $-_l$ to denote subtraction modulo l.

Consider the *simplified* convex decomposition C' which is obtained from C by (i) removing duplicate edges which occur for diagonals $(d_i, d_{i+_l 1})$ and by (ii) contracting every path of boundary edges between two diagonal endpoints, without any interior diagonal endpoints, to one edge. The vertices of C' are $d_0, d_1, \ldots, d_{l-1}$. Let $T_{C'}$ be a triangulation of C' (cf. Figure 5.10 (a)). Let a *c-diagonal* be a diagonal of the convex decomposition C. Let a *t-diagonal* be all other edges of the triangulation $T_{C'}$. In Figure 5.10, (d_0, d_1) and (d_2, d_3) are c-diagonals and (d_1, d_3), (d_1, d_2), and (d_3, d_0) are t-diagonals.

For any c-diagonal or t-diagonal (d_i, d_j) let $RG(i, j)$ denote the reachability graph and let $CRG(i, j)$ denote the combined reachability graph spanning the part of the free space between d_i and d_j. Thus $CRG(i, j)$ takes into account the diagonals that have both endpoints in between d_i and d_j, using wraparound modulo l when j is smaller than i. We assume that $RG(i, j)$ and $CRG(i, j)$ use the refinement of the reachability structure of the double free space diagram. That is, we assume the intervals occurring as vertices in $RG(i, j)$ and $CRG(i, j)$, respectively, to be the (possibly) refined inervals of the larger reachability structure. This refinement can easily be computed in $O(mn \log mn)$ time by projecting all vertical (horizontal) intervals of the free space onto the vertical (horizontal) boundaries of the free space.

The following lemma shows that a combined reachability graph $CRG(i, j)$ can be recursively constructed from reachability graphs and combined reachability graphs that span smaller parts of the free space. This construction uses two functions: Combine and Merge. The Merge function "concatenates" adjacent reachability graphs (some of which may be combined reachability graphs) by taking the union of the graphs, computing the transitive closure, and discarding intermediate vertices. The Combine function computes the combined reachability graph from the input reachability graph by keeping only those edges that encode valid placements of diagonals. In the following, a pair (d_i, d_j) will denote either a c-diagonal or a t-diagonal. A triple (d_i, d_h, d_j) denotes a triangle in $T_{C'}$ with the endpoints in the given order.

Lemma 5.4. *For any c-diagonal or t-diagonal (d_i, d_j) of $T_{C'}$ holds:*

(C1) *If (d_i, d_j) is a c-diagonal with $j = i +_l 1$ then*

$$CRG(i, j) = \text{Combine}(RG(i, j))$$

(T1) *If (d_i, d_j) is a t-diagonal with $j = i +_l 1$ then*

$$CRG(i, j) = RG(i, j)$$

(C2) *If (d_i, d_j) is a c-diagonal with $j \neq i +_l 1$ then*

$$CRG(i, j) = \text{Combine}(\text{Merge}(CRG(i, h), CRG(h, j))),$$

where (d_i, d_h, d_j) is a triangle in $T_{C'}$.

(T2) *If (d_i, d_j) is a t-diagonal with $j \neq i +_l 1$ then*

$$CRG(i, j) = \text{Merge}(CRG(i, h), CRG(h, j)),$$

where (d_i, d_h, d_j) is a triangle in $T_{C'}$.

Proof. (C1) and (C2) cover the cases where (d_i, d_j) is a c-diagonal, and (T1) and (T2) cover the cases where (d_i, d_j) is a t-diagonal. In the following, we show how the combined reachability graph can be computed by partitioning the free space, computing sub-reachability graphs on each part, and then merging and combining the subgraphs.

(C1) and (T1) follow directly from the definition of combined reachability graphs. Now consider case (T2) in which (d_i, d_j) is a t-diagonal with $j \neq i +_l 1$. There are at most two triangles in $T_{C'}$ incident to (d_i, d_j): $\Delta = (d_i, d_h, d_j)$ and $\Delta' = (d_j, d_{h'}, d_i)$ for some h and h'. Since $CRG(i, j)$ contains the reachability information for the part of the free space between d_i and d_j, we only need to consider the triangle Δ, which has to exist because $j \neq i +_l 1$. (Δ' is considered for $CRG(j, i)$.) A path in the free space between d_i and d_j is feasible if and only if the sub-path between d_i and d_h is feasible, the sub-path between d_h and d_j is feasible, and in case of (C2) it also places (d_i, d_j) correctly. Thus, the first sub-path has to lie in $CRG(i, h)$ and the second in $CRG(h, j)$. Merging these two graphs yields (T2). If (d_i, d_j) is a c-diagonal then an additional call to the Combine function ensures that only edges are kept that encode valid placements of (d_i, d_j), which yields (C2). □

Corollary 5.3 below follows directly from Lemma 5.4.

Corollary 5.3. *There is a feasible path in the free space diagram starting in $[d_{i -_l 1}, d_i)$ if and only if there is an edge in $G = \text{Merge}(RG(i -_l 1, i), CRG(i, i -_l 1), RG(i -_l 1, i))$ connecting an interval in $[d_{i -_l 1}, d_i)$ to the same interval translated by n on the second half of the top boundary.*

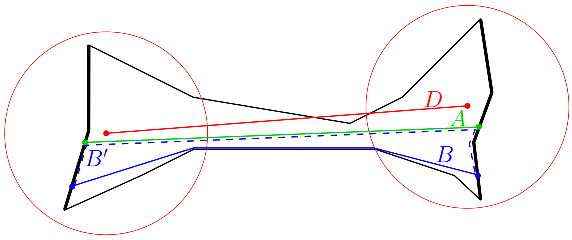


Figure 5.11: If one shortest path has Fréchet distance at most ε to the line segment, then so does every other shortest path in the hourglass.

By Corollary 5.3, a feasible path in free space corresponds to an edge in the graph $G =$ MERGE$(RG(i -_l 1, i), CRG(i, i -_l 1), RG(i -_l 1, i))$. Note that for computing G it would suffice to consider the subgraph of $CRG(i, i -_l 1)$ of only vertical segments in free space. However, this would not improve the asymptotic complexity of (the computation of) G.

5.4.3 Fréchet Distance between a Diagonal and an Hourglass

The following lemma shows how to solve the decision problem for the Fréchet distance between a diagonal and all shortest paths in an hourglass. As introduced in Section 5.2.4, we consider generalized hourglasses given by two end chains, i.e., two chains of consecutive edge segments. With a *shortest path in the hourglass* we always refer to a shortest path between two points on the two end chains defining the hourglass.

Lemma 5.5. *Let an hourglass and a diagonal be given such that the end chains of the hourglass are contained in the ε-disks around the endpoints of the diagonal. 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.*

Proof. Let D be the diagonal, $A = a_1, \ldots, a_l$ be a shortest path in the hourglass with $\delta_F(A, D) \leq \varepsilon$, and let $B = b_1 \ldots, b_k$ be another shortest path in the hourglass, as in Figure 5.11. Because the Fréchet distance between D and A is at most ε, each ε-disk around the endpoints of D must contain at least one endpoint of A. Without loss of generality we assume that a_1 and b_1 lie in one ε-disk and a_l and b_k lie in the other ε-disk.

We define $B' = b_1, v_1, \ldots, v_i, a_1, \ldots, a_l, w_1, \ldots, w_j, b_k$, where $v_1, \ldots, v_i$ and $w_1, \ldots, w_j$ $(i, j \geq 0)$ are the vertices of the end chains that (possibly) lie on a shortest path between b_1, a_1 and b_2, a_2, respectively. See Figure 5.11 for an illustration. Because a_1, b_1 have distance at most ε to one endpoint of D, a_l, b_k have distance at most ε to the other endpoint, and $\delta_F(A, D) \leq \varepsilon$, it follows that $\delta_F(B', D) \leq \varepsilon$.

If B' is not the shortest path from b_1 to b_k there exist two points on B' such that simplifying B' by replacing the part of B' between the two points by the line segment between them yields a shorter path B'' from b_1 to b_k in the hourglass. By Lemma 5.1, $\delta_F(B'', D) \leq \delta_F(B', D)$. Repeating this process and observing that the simplified paths converge to B shows that $\delta_F(B, D) \leq \delta_F(B', D) \leq \varepsilon$. ☐

Note that in an open hourglass there always exists (by definition) a shortest path S between the end chains which is a segment. Thus, if the end chains of an open hourglass lie in ε-disks around the endpoints of the diagonal – as in Lemma 5.5 – then the Fréchet distance between the segment S and the diagonal is at most ε.

This holds because the Fréchet distance between two segments equals the maximum distance of the endpoints.

5.4.4 Fréchet Distances between a Diagonal and Multiple Hourglasses

In Section 5.4.5 we need to solve the decision problem for the Fréchet distances between a diagonal and multiple hourglasses that have a common end chain. This can be done in linear time by choosing an arbitrary vertex on each end chain of the hourglasses and then using Lemma 5.5 and the following Lemma 5.6.

Lemma 5.6. *Given a diagonal, a polygon with m vertices, and a set of m points $w_1, \ldots, w_m$ on the boundary of the polygon. The decision problems for the Fréchet distances between the diagonal and the $m-1$ shortest paths $\pi(w_1, w_i)$ between w_1 and w_i for $i = 2, \ldots, m$ can be solved in total $O(m)$ time.*

Proof. We add the points $w_1, \ldots, w_m$ to the vertices of the polygon. Then we run 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. [30]. During the algorithm we decide whether the Fréchet distance between the diagonal and the shortest path from w_1 to w_i for $i = 2, \ldots, m$ is less than or equal to ε using the free space diagram and the reachability structure for that problem. This structure can be updated in amortized constant time as follows.

The algorithm by Guibas et al. [30] computes the shortest paths starting at w_1 such that when a new vertex is processed all other vertices on its shortest path to w_1 have already been processed and the previous vertex on the shortest path is known. Thus, for deciding the Fréchet distance between the shortest path to the new vertex, we only need to discard some of the last cells and compute the new last cell of the free space diagram. One cell can be computed in constant time and the discarding can be done in amortized constant time.

If the previous vertex was not reachable in free space, the new vertex is not either, and we store this for the new vertex. If the previous vertex was reachable, we compute the top boundary of the new cell of the free space diagram. For an original vertex of the polygon which is not one of the w_i, we then test and store whether the top boundary is reachable from the last cell and store the leftmost reachable point on the boundary. For a vertex w_i we also test and store whether the right corner of the top boundary is reachable which means that the Fréchet distance is less than or equal ε. □

5.4.5 Decision Algorithm

Now we can give our algorithm for solving the decision problem for the Fréchet distance between simple polygons: Algorithm 4. The objective of the algorithm is to search for a feasible path in the free space diagram, which by Corollary 5.2 is equivalent to solving the decision problem for the Fréchet distance between simple polygons. We first explain the details of the algorithm and then show in Theorem 5.1 that it solves the decision problem for the Fréchet distance between simple polygons in polynomial time.

Algorithm 4: DecideFréchet(P, Q, ε)

Input: Simple Polygons P, Q, $\varepsilon > 0$
Output: Is $\delta_F(P, Q) \leq \varepsilon$?

```
1  Compute approximate minimum convex decompositions of P and Q. Let C
   be the smaller of these. Assume without loss of generality that C is a convex
   decomposition of P. Let l be the number of diagonal endpoints in C.
2  Let C' be the simplified convex decomposition obtained from C. Compute a
   triangulation T_{C'} of C'.
3  Compute a single free space diagram of the boundary curves.
4  forall i = 0, ..., l − 1 do
5      Compute RG(i, i +_l 1)
6  end
7  forall diagonals in the convex decomposition C do
8      forall placements in the free space do
9          Decide δ_F(diagonal, shortest path) ≤ ε? for a shortest path in the
           hourglass of the placement.
10     end
11 end
12 forall i = 0, ..., l − 1 do
13     Compute CRG(i, i −_l 1) based on T_{C'} using memoization.
14     Compute G = MERGE(RG(i −_l 1, i), CRG(i, i −_l 1), RG(i −_l 1, i)).
15     Query G for a feasible path starting in [d_{i −_l 1}, d_i).
16 end
17 Output true if a feasible path has been found, else output false.
```

Lines **1**–**11** consist of preprocessing. In line **1** we compute a convex decomposition C of P or Q. The size of the convex decomposition will be a multiplicative factor in the runtime of the algorithm. Therefore we compute an approximate minimum convex decomposition of both P and Q and choose the smaller. Then the size k of the chosen convex decomposition fulfills $k \in O(\min(c(P), c(Q)))$, where $c(P)$ and $c(Q)$ denote the size of a minimum convex decomposition of P and Q, respectively. The number l of endpoints of diagonals in C fulfills $l \leq 2k$. We assume without loss of generality that the chosen convex decomposition is a decomposition of P. If Q has the smaller convex decomposition, we swap the roles of P and Q for the rest of the algorithm.

Let $d_0, \ldots, d_{l-1}$ be the endpoints of the diagonals in C in the given order. In line **2** we compute a triangulation $T_{C'}$ of the simplified convex decomposition C' (see Section 5.4.2). In line **3** a single free space diagram of the boundary curves is computed. In line **4**–**6** the reachability graphs $RG(i, i +_l 1)$ for $i = 0, \ldots, l-1$ are computed. The graph $RG(i, i +_l 1)$ is computed by first computing the reachability structure of the part of the free space between d_i and and $d_{i +_l 1}$. The intervals of this reachability structure are refined to the refinement of the reachability structure of the double free space diagram. Then the reachability structure is converted to the reachability graph.

In lines **7**–**11** we test for all diagonals in C which of their possible placements in the free space are valid, i.e., map the diagonal to a shortest path with Fréchet distance at most ε. For this we consider all free intervals on the boundary of Q and pick one arbitrary point in each interval. Then we apply Lemma 5.6 using these points and by Lemma 5.5 this solves the decision problem for the Fréchet distance for any shortest path determined by the placement.

In lines **12**–**16** we loop over all diagonal endpoints d_i and search for a feasible path that starts in $[d_{i-_l 1}, d_i)$ on the bottom boundary of the free space diagram. For this, we compute $CRG(i, i -_l 1)$ using the recursive formula of Lemma 5.4 based on the diagonals of $T_{C'}$.

In order to avoid recomputation of the same combined reachability graphs we employ the dynamic programming technique of memoization in which the first recursive call to $CRG(i, j)$ computes and stores the graph, and subsequent calls simply access the stored graph without any further computation. Hence, we compute and store all $CRG(i, j)$ for all c-diagonals and t-diagonals. The COMBINE procedure used in Lemma 5.4 checks for each edge in the input graph whether the hourglass between the two intervals contains (only) valid placements for the diagonal (d_i, d_j). The validity of the placements can be looked up in constant time from the results precomputed in lines **7–11** for the corresponding hourglass. If a placement is not valid, then the edge is deleted, otherwise it is kept.

In line **14** we merge $RG(i -_l 1, i)$ to the front and to the end of $CRG(i, i -_l 1)$. This results in the graph G whose edges, by Corollary 5.3, encode feasible paths starting in $[d_{i-_l 1}, d_i)$. Then, in line **15**, we query for feasible paths in G by checking whether any interval in $[d_{i-_l 1}, d_i)$ on the bottom boundary is connected by an edge to the same interval translated by n on the second half of the top boundary. Finally in line **17**, we output "true" if a feasible path has been found starting in one of the intervals, else we output "false".

Theorem 5.1. *Algorithm 4 solves the decision problem for the Fréchet distance between two simple polygons P, Q. Its runtime is $O(k\,T(mn))$, where $T(N)$ is the time needed to multiply two $N \times N$ matrices (using $O(N^2)$ space), n and m are the number of vertices of P and Q, and k is the minimum size of a minimum convex decomposition of P or Q.*

Note that $T(N) = \Omega(N^2)$ and the currently fastest known matrix multiplication algorithm is the algorithm of Coppersmith and Winograd [18] which is an improvement of Strassen's algorithm [49] and has a runtime of $T(N) = O(N^{2.376})$ using $O(N^2)$ space.

Proof. We first show the correctness of the algorithm and then analyze its time complexity.

Correctness By Corollary 5.2 the Fréchet distance between simple polygons is less than or equal to a given value ε if and only if there is a feasible path in the free space diagram. Thus it suffices to show that Algorithm 4 correctly determines whether such a feasible path exists.

Lines **1–11** of the algorithm are preprocessing steps. Lemma 5.5 proves that the loop in lines **7–11** correctly solves the decision problem for the Fréchet distance between a diagonal and any shortest path in a certain hourglass.

In the main part of the algorithm, lines **12–16**, we search for a feasible path in the free space. A feasible path has to start in in one of the intervals $[d_{i-_l 1}, d_i)$, where d_i is a diagonal endpoint, and we loop over all such diagonal endpoints. For each d_i we compute $CRG(i, i -_l 1)$, merge $RG(i -_l 1, i)$ to the front and to the end, and query for a feasible path in the resulting graph G. By Corollary 5.3, edges in G encode feasible paths.

Time Complexity Computing the approximate minimum convex decompositions in line **1** takes $O(m \log m + n \log n)$ time. The triangulation $T_{C'}$ in line **2** can be computed in $O(k)$ time by fan-triangulating every convex face in the convex decomposition C' which is of size $O(k)$.

The free space diagram in line **3** can be computed in time $O(mn)$. For computing all reachability graphs in lines **4–6**, we first compute the reachability structures for all parts of the free space between neighboring diagonal endpoints. These reachability structures can be computed in $O(\sum_{i=1}^{l} mn_i \log(mn_i)) = O(mn \log mn)$ time, because each free space part consists of $m \times n_i$ cells for $i = 1, \ldots, l$ and $\sum_{i=1}^{l} n_i = n$. The refinement of the reachability structure of the double free space diagram can be computed in $O(mn \log mn)$ time. From this refinement and the partial reachability structures, the reachability graphs $RG(i, i{+_l}1)$ can be computed in total $O(k(mn)^2)$ time.

In lines **7–11** we test for all diagonals of C which of their possible placements in the free space are valid. For a fixed diagonal endpoint this can be done in $O(m)$ time using Lemma 5.6. Thus, testing the placements for one diagonal can be done in $O(m^2)$ and for all diagonals in $O(km^2)$ time.

In the loop in lines **12–16** each $CRG(i, j)$ is computed exactly once using a recursive call as described in Lemma 5.4. A $CRG(i, j)$ is computed for all c- and t-diagonals of the triangulation $T_{C'}$ which has $l = O(k)$ vertices. Thus, $O(k)$ combined reachability graphs are computed and stored. Each recursive call involves at most one MERGE and one COMBINE, and line **14** adds another MERGE, for a total of $O(k)$ MERGE and COMBINE operations. The triangles incident to (d_i, d_j) that are needed during a recursive call to $CRG(i, j)$ (cf. Lemma 5.4) can be found in constant time, assuming that $T_{C'}$ is given in an appropriate graph-representation, such as a doubly-connected edge list.

We merge the combined reachability graphs by multiplying their adjacency matrices (using boolean operations). Thus, merging two combined reachability graphs takes the time to multiply two $O(mn) \times O(mn)$ matrices.

A COMBINE operation involves exactly one diagonal and the correct placing of this diagonal is checked for each of the $O((mn)^2)$ edges by looking up the precomputed results of lines **7–11** which takes $O((mn)^2)$ time. In total, the complexity of the loop in lines **12–16** is therefore $O(k\,T(mn))$, where $T(N)$ is the time to multiply two $N \times N$ matrices and k is the size of the convex decomposition. Note that looping line **13** over all diagonal endpoints (lines **12–16**) does not increase the complexity because we store and re-use the combined reachability graphs of subresults. In line **15** we query for a feasible path, which can be done in amortized $O(mn)$ time for the loop in lines **12–16**.

In total, the time complexity of the algorithm is dominated by the time complexity of the main loop in lines **12–16** which is $O(k\,T(mn))$. □

The space complexity of Algorithm 4 is $O(k(mn)^2)$ since it stores $O(k)$ combined reachability graphs and $O(k)$ reachability graphs which each have complexity $O((mn)^2)$.

5.5 Computing the Fréchet Distance

For computing the Fréchet distance we proceed as in the case of polygonal curves [6]: we search a set of critical values for ε using the decision algorithm in each step. The set of critical values is a candidate set for the Fréchet distance, i.e., a set of values that the Fréchet distance may attain. For this we first give the set of critical values in Lemma 5.7 and use this set of critical values to compute the Fréchet distance in Theorem 5.2.

By Corollary 5.2 the Fréchet distance between simple polygons equals a value ε if and only if there is a feasible path in the free space for parameter ε and there is no feasible path in the free space for any parameter $\varepsilon' < \varepsilon$. Such a value of ε is an example of a *critical value*.

Alt and Godau [6] distinguish three kinds of critical values for polygonal curves. Each type of critical value corresponds to a combinatorial change in the free space diagram and can be described geometrically. For their Algorithm 2 these are:

(a) The critical value is the distance between corresponding endpoints – the corners of the free space diagram become free.

(b) The critical value is the distance of a vertex on one curve to a segment of the other curve – a cell boundary becomes non-empty.

(c) Two vertices on one curve have the same distance, the critical value, to a point on the other curve – a passage through several cells may open.

For closed curves the critical values of type (c) refer to passages in either one of the double free space diagrams (with either curve doubled). Now we can give the critical values for simple polygons.

Lemma 5.7. *Given two simple polygons with m and n vertices, respectively. Their Fréchet distance equals one of $O(m^2n + mn^2)$ critical values. The set of critical values can be computed in $O(m^2n + mn^2)$ time.*

Proof. Recall from Section 5.4.1 that a feasible path has two properties: (1) it realizes a Fréchet distance less than or equal ε on the boundary curves and (2) it maps diagonals to shortest paths with Fréchet distance less than or equal ε. A critical value for the Fréchet distance between simple polygons is therefore either (1) a critical value for the Fréchet distance between the closed polygonal boundary curves, or (2) a critical value for the Fréchet distance between a diagonal and a shortest path, or both.

Equivalently, the critical values for simple polygons can be derived by comparing the decision algorithm for the Fréchet distance between the polygons with the decision algorithm for the Fréchet distance between the boundary curves (Algorithm 1 in [6]). In addition to the critical values for the boundary curves, critical values for simple polygons can occur in lines **7-11** of our Algorithm 4. In these lines we test whether the Fréchet distance between a diagonal and a shortest path is less than or equal ε.

There are $O(m^2n + mn^2)$ critical values for the Fréchet distance between the closed polygonal boundary curves, each of which can be computed in constant time [6]. It remains to compute the critical values for the Fréchet distance between a diagonal and a shortest path. Even though there are infinitely many shortest paths in a simple polygon, we will see that a polynomial number of these critical values suffice.

Each type of critical value for polygonal curves involves at least one vertex of one of the curves. For the Fréchet distance between a diagonal and a polygonal curve it has to involve a vertex of the curve. This is true because the free space of a diagonal and a polygonal curve consists only of one column. Thus a path in the free space does not pass through any vertical cell boundaries, and the critical values of type (b) and (c) can be restricted to horizontal cell boundaries.

Thus, we can restrict the critical values for the Fréchet distance between a diagonal and a shortest path to the critical values of type (a) and the critical values of type (b) and (c) for inner vertices of the shortest path. The critical values of type (a), i.e., the distances between endpoints, are distances between the boundary curves, which we already consider in the critical values of type (b) for the boundary curves. Therefore, as critical values for the Fréchet distance between a diagonal and a shortest path, it suffices to compute the critical values of type (b) and (c) for inner vertices of the shortest path.

A shortest path in a simple polygon is a polygonal curve with arbitrary first and last vertex on the boundary and all inner vertices are reflex vertices of the polygon boundary, i.e., they have an interior angle larger than π. There are $k \leq n$ diagonals and at most m reflex vertices in Q.

For a fixed diagonal we compute the critical values for all shortest paths as follows. The critical values of type (b) are the distances of a reflex vertex to the diagonal. These are $O(m)$ values each of which can be computed in constant time. The critical values of type (c) occur when two reflex vertices have the same distance to a point on the diagonal. This point is the intersection of the diagonal with the bisector between the two reflex vertices. These are $O(m^2)$ critical values, each of which can again be computed in constant time. In total, for each diagonal we compute $O(m)$ critical values of type (b) and $O(m^2)$ critical values of type (c) in total $O(m^2)$ time. This yields $O(km^2)$ critical values for the Fréchet distance between a diagonal and a shortest path which can be computed in time $O(km^2)$. □

Remark. As critical values for the second property of a feasible path we use critical values for the Fréchet distance between a diagonal and a shortest path. We could reduce these critical values to the Fréchet distances between a diagonal and a shortest path which are attained at an inner vertex of the shortest path. This, however, would increase the time for computing these critical values and would not reduce the number of critical values asymptotically. For computing these critical values we would compute for all of the k diagonals in P and a shortest path between reflex vertices $(r_1, \ldots, r_l)$ in Q the Fréchet distance of the diagonal (d_i, d_j) and the polygonal path $(d_i, r_1, \ldots, r_l, d_j)$. This would also yield $O(m^2 k)$ critical values but to compute them would take $O(m^3 k \log m)$ time.

Using the above lemma we can compute the Fréchet distance between simple polygons in polynomial time.

Theorem 5.2. *The Fréchet distance between two simple polygons P and Q can be computed in $O(k\,T(mn)\log(mn))$ time, where $T(N)$ is the time needed to multiply two $N \times N$ matrices, (using $O(N^2)$ space), n and m are the number of vertices of P and Q, and k is the minimum size of a minimum convex decomposition of P or Q.*

Proof. We use Algorithm 1 with Algorithm 4 as decision algorithm and the critical values given in Lemma 5.7. The correctness of the resulting algorithm follows from Theorem 5.1 and Lemma 5.7.

The run time of computing the critical values is $O(m^2 n + mn^2)$ and of sorting them is $O((m^2 n + mn^2)\log(mn))$. The binary search has an $O(k\,T(mn)\log(mn))$ run time which dominates the total run time. □

The space complexity of this algorithm is dominated by the space complexity of the decision algorithm in the binary search, which is $O(k(mn)^2)$.

Note that the algorithm for polygonal curves [6] searches the set of critical values using Cole's [17] technique for parametric search [39] based on sorting. In their case this yields a better run time than a binary search. For curves the run time for a binary search is dominated by the time for computing and sorting the critical values. In our case the run time is always dominated by the actual binary search, due to the higher run time of the decision algorithm. Thus, parametric search would not improve the run time of our algorithm.

5.6 Discussion

In this chapter we have given an algorithm for computing the Fréchet distance between simple polygons. This is the first polynomial time algorithm for computing the Fréchet distance for a non-trivial class of surfaces. For this we showed that the set of realizing maps can be restricted to a set of "nice" maps which we can handle algorithmically.

As discussed in Section 3.6, an interesting open problem is whether such a restriction of the feasible set of homeomorphisms is possible for the general case of triangulated surfaces. In particular, we would like to find other classes of surfaces for which the set of feasible homeomorphisms can be restricted such that computability or even polynomial time computability can be shown. A promising next step might be to extend the results for simple polygons to more general polygons such as polygons with holes, self-intersections, or folds.

The NP-hardness proof [29] shows that it is NP-hard to decide the Fréchet distance between a triangle and a selfintersecting polygon. The selfintersecting polygon is given by a non-injective parameterization $f\colon [0,1]^2 \to \mathbb{R}^2$. It can be expanded to a non-selfintersecting surface in $\mathbb{R}^4$. For this, consider the function graph of the parameterization with scaled parameter value, i.e., consider $f'(t) = (\delta t, f(t))$ where δ is the scaling parameter. The function f' is injective and for δ tending to 0, f' comes arbitrary close to an embedding of f. Thus, the NP-hardness holds also for non-selfintersecting surfaces in $\mathbb{R}^4$. In this chapter, we have shown that the Fréchet distance is polynomial time computable if the parameterizations are required to be injective in $\mathbb{R}^2$, i.e., the polygons are simple. It remains open, whether it is NP-hard to decide whether the Fréchet distance between non-selfintersecting surfaces in $\mathbb{R}^3$ is less than or equal ε.