How to Walk Your Dog in the Mountains with No Magic Leash*

Sariel Har-Peled† Amir Nayyeri‡ Mohammad Salavatipour§
Anastasios Sidiropoulos¶

January 29, 2014

Abstract

We describe a $O(\log n)$-approximation algorithm for computing the homotopic Frechet distance between two polygonal curves that lie on the boundary of a triangulated topological disk. Prior to this work, algorithms were known only for curves on the Euclidean plane with polygonal obstacles.

A key technical ingredient in our analysis is a $O(\log n)$-approximation algorithm for computing the minimum height of a homotopy between two curves. No algorithms were previously known for approximating this parameter. Surprisingly, it is not even known if computing either the homotopic Frechet distance, or the minimum height of a homotopy, is in NP.

1 Introduction

Comparing the shapes of curves – or sequenced data in general – is a challenging task that arises in many different contexts. The Frechet distance and its variants (e.g. dynamic time-warping [KP99]) have been used as a similarity measure in various applications such as matching of time series in databases [KKS05], comparing melodies in music information retrieval [SGHS08], matching coastlines over time [MDBH06], as well as in map-matching of vehicle tracking data [BPSW05, WSP06], and moving objects analysis [BBG08a, BBG +08b]. See [AB05, AG95] for algorithms for computing the Frechet distance.

Informally, for a pair of such curves $f, g : [0,1] \to D$, for some ambient metric space $(D, d)$, their Frechet distance is the minimum length leash needed to traverse both curves in sync. To this end, imagine a person traversing $f$ starting from $f(0)$, and a dog traversing $g$ starting from $g(0)$, both traveling continuously along these curves without ever moving backwards. Then, the Frechet distance is the infimum over all possible traversals, of the maximum distance between the person and the dog. Specifically, given a bijective continuous parameterization $\phi : [0,1] \to [0,1]$, the width of this reparameterization, i.e., the longest leash needed by this reparameterization, is $\text{width}(\phi) = \sup_{t \in [0,1]} d\left(f(t), g(\phi(t))\right)$, where

---

*A preliminary version of this paper appeared in SoCG 2012 [HNSS12].
†Department of Computer Science, University of Illinois, Urbana-Champaign; sariel@illinois.edu. Work on this paper was partially supported by a NSF AF award CCF-0915984.
‡School of Electrical Engineering & Computer Science, Oregon State University; nayyeria@eecs.oregonstate.edu. Part of this work was done while visiting Toyota Technological Institute at Chicago; nayyeri2@illinois.edu.
§Department of Computing Science, University of Alberta; Supported by NSERC and Alberta Innovates. Part of this work was done while visiting Toyota Technological Institute at Chicago; mreza@cs.ualberta.ca.
¶Department of Computer Science & Engineering, and Department of Mathematics, The Ohio State University; sidiropoulos.1@osu.edu.
$d(x, y)$ is the Euclidean distance between $x$ and $y$. Consequently, the Frechet distance between $f$ and $g$ is defined to be

$$d_F(f, g) = \inf_{\phi: [0,1] \rightarrow [0,1]} \text{width}(\phi),$$

where $\phi$ ranges over all orientation-preserving homeomorphisms.

While this measure captures similarities between two curves when the underlying space is Euclidean, it is not as informative for more complicated underlying spaces such as a surface. For example, imagine walking a dog in the woods. The leash might get tangled as the dog and the person walk on two different sides of a tree. Since the Frechet distance cares only about the distance between the two moving points, the leash would “magically” jump over the tree. In reality, when there is no “magic” leash that jumps over a tree, one has to take into account the extra length needed (for the leash) to pass over such high points.

**Homotopic Frechet distance.** To address this shortcoming, a natural extension of the above notion called *homotopic Frechet distance* was introduced by Chambers et al. [CCE+10]. Informally, revisiting the above person-dog analogy, we consider the infimum over all possible traversals of the curves, but this time, we require that the person is connected to the dog via a leash, i.e., a curve that moves continuously over time. Furthermore, one keeps track of the leash during the motion, where the purpose is to minimize the maximum leash length needed.

To this end, consider a homotopy $h : [0,1]^2 \rightarrow D$ that also specifies a morph of a unit square to $D$. For parameters $\sigma, \tau \in [0,1]$ consider the one dimensional functions $\ell(\tau) = h(\tau, \cdot) : [0,1] \rightarrow D$ and $\mu(\sigma) = h(\cdot, \sigma) : [0,1] \rightarrow D$. These are parameterized curves that are the natural restrictions of $h$ into one dimension. We require that $\mu(0) = f$ and $\mu(1) = g$. The homotopy width of $h$ is $\text{width}(h) = \sup_{\tau \in [0,1]} \|\ell(\tau)\|$, and the homotopic Frechet distance between $f$ and $g$ is

$$d_H(f, g) = \inf_{h:[0,1]^2 \rightarrow D} \text{width}(h),$$

where the infimum is over all such mappings, and $\|\cdot\|$ denotes the length of a curve.

Clearly, $d_H(f, g) \geq d_F(f, g)$ and, furthermore, $d_H(f, g)$ can be arbitrary larger than $d_F(f, g)$. We remark that $d_H(f, g) = d_F(f, g)$ for any pair of curves in the Euclidean plane, as we can always pick the leash to be a straight line segment between the person and the dog. In other words, the map $h$ in the definition of $d_H$ can be obtained from the map $h$ in the definition of $d_F$ via an appropriate affine extension. However, this is not true for general ambient spaces, where the leash might have to pass over obstacles, hills, etc. In particular, in the general settings, usually, the leash would not be a geodesic (i.e., a shortest path) during the motion.

The homotopic Frechet distance is referred to as the morph width of $f$ and $g$, and it bounds how far a point on $f$ has to travel to its corresponding point in $g$ under the morph of $h$ [EGH+02]. The length of $\mu(\sigma)$ is the height of the morph at time $\sigma$, and the height of such a morph is $\text{height}(\mu) = \sup_{\sigma \in [0,1]} \|\mu(\sigma)\|$. The homotopy height between $f$ and $g$, bounded by $\ell(0)$ and $\ell(1)$, is

$$h(f, g, \ell(0), \ell(1)) = \inf_{\mu} \text{height}(\mu),$$

where $\mu$ varies over all possible morphs between $f$ and $g$, such that each curve in $\mu$ has one end on $\ell(0)$ and one end on $\ell(1)$. See Figure 1.1 for an example. Note that if we do not constrain the endpoints of the curves during the morph to stay on $\ell(0)$ and $\ell(1)$, the problem of computing the minimum height
Figure 1.1: (i) Two curves \(f\) and \(g\), and (ii) the parameterization of their homotopic Frechét distance.

Homotopy is trivial. One can contract \(f\) to a point, send it to a point in \(g\), and then expand it to \(g\). To keep the notation simple, we use \(h(f, g)\) when \(f\) and \(g\) have common endpoints.

Intuitively, the homotopy height measures how long the curve has to become as it deforms from \(f\) to \(g\), and it was introduced by Chambers and Letscher [CL09, CL10] and Brightwell and Winkler [BW09]. Observe that if we are given the starting and ending leashes \(\ell(0)\) and \(\ell(1)\) then the homotopy height of \(f\) and \(g\), is the homotopic Frechét distance between \(\ell(0)\) and \(\ell(1)\).

Here, we are interested in the problems of computing the homotopic Frechét distance and the homotopy height between two simple polygonal curves that lie on the boundary of an arbitrary triangulated topological disk.

**Why are these measures interesting?** For the sake of the discussion here, assume that we know the starting and ending leash of the homotopy between \(f\) and \(g\). The region bounded by the two curves and these leashes, form a topological disk, and the mapping realizing the homotopic Frechét distance is a mapping of the unit square to this disk \(D\). This mapping specifies how to sweep over \(D\) in a geometrically “efficient” way (especially if the leash does not sweep over the same point more than once), so that the leash (i.e., the sweeping curve) is never too long [EGH+02]. As a concrete example, consider the two curves as enclosing several mountains between them on the surface – computing the homotopic Frechét distance corresponds to deciding which mountains to sweep first and in which order.

Furthermore, this mapping can be interpreted as surface parameterization [Flo97, SdS00] and can thus be used in applications such as texture mapping [BVIG91, PB00]. In the texture mapping problem, we wish to find a continuous and invertible mapping from the texture, usually a two-dimensional rectangular image, to the surface.

Another interesting interpretation is when \(f\) is a closed curve, and \(g\) is a point. Interpreting \(f\) as a rubber band in a 3d model, the homotopy height between \(f\) and \(g\) here is the minimum length the rubber band has to have so that it can be collapsed to a point through a continuous motion within the surface. In particular, a short closed curve with large homotopy height to any point in the surface is a “neck” in the 3d model.

To summarize, these measures seem to provide us with a fundamental understanding of the structure of the given surface/model.

**Continuous vs. discrete.** Here we are interested in two possible models. In the continuous settings, as described above, the leash moves continuously in the interior of the domain. In the discrete settings, the leash is restricted to the triangulation edges. As such, a transition of the leash corresponds to the leash “jumping” over a single face at each step. The two versions are similar in nature, but technically require somewhat different tools and insights. This issue is discussed more formally in Section 2.
Previous work. The problem of computing the (standard) Frechét distance between two polygonal curves in the plane has been considered by Alt and Godau [AG95], who gave a polynomial time algorithm. Eiter and Mannila [EM94] studied the easier discrete version of this problem. Computing the Frechét distance between surfaces [Fre24], appears to be a more difficult task, and its complexity is poorly understood. The problem has been shown to be NP-hard by Godau [God99], while the best algorithmic result is due to Alt and Buchin [AB05], who showed that it is upper semi-computable.

Efrat et al. [EGH+02] considered the Frechét distance inside a simple polygon as a way to facilitate sweeping it efficiently. They also used the Frechét distance with the underlying geodesic metric as a way to obtain a morph between two curves. For recent work on the Frechét distance, see [CW10, CLJL11, HR11, CDH+11, DHW12, CW12] and references therein.

Chambers et al. [CCE+10] gave a polynomial time algorithm to compute the homotopic Frechét distance between two polygonal curves on the Euclidean plane with polygonal obstacles. Chambers and Letscher [CL09, CL10] and Brightwell and Winkler [BW09] considered the notion of minimum homotopy height, and proved structural properties for the case of a pair of paths on the boundary of a topological disk. We remark that in general, it is not known whether the optimum homotopy has polynomially long description. In particular, it is not known whether the problem is in NP.

Variants of the Frechét distance for curves that are known to be computationally hard, include (i) the problem of finding the most similar simple (i.e., no self crossings) curve to a given curve on a surface [SW13], and (ii) computing the optimal Frechét distance when allowing shortcuts anywhere on one of the curves [BDS13].

Our results. In this paper, we consider the problems of computing the homotopic Frechét distance and the homotopy height between two simple polygonal curves that lie on the boundary of a triangulated topological disk $D$ that is composed of $n$ triangles.

We give a polynomial time $O(\log n)$-approximation algorithm for computing the homotopy height between $f$ and $g$. Our approach is based on a simple, yet delicate divide and conquer approach.

We use the homotopy height algorithm as an ingredient for a $O(\log n)$-approximation algorithm for the homotopic Frechét distance problem. Here is an high-level description of our algorithm for approximating the homotopic Frechét distance: We first guess (i.e., search over) the optimum (i.e., $d_H(f,g)$). Using this guess, we classify parts of $D$ as “obstacles”, i.e., regions over which a short leash cannot pass. Consider the punctured disk obtained from $D$ after removing these obstacles. Intuitively, two leashes are homotopic if one can be continuously deformed to the other one within the punctured disk, while its endpoints remain on the boundary during the deformation. Observe that the leashes of the optimum solution are homotopic. We describe a greedy algorithm to compute a “small” number of homotopy classes out of infinitely many choices. The homotopic Frechét distance constrained to paths inside one of these classes is a polynomial approximation to the homotopic Frechét distance in $D$. We can then perform a binary search over this interval to acquire a better approximation. An extended version of the homotopy height algorithm is used in this algorithm in several places.

The $O(\log n)$ factor shows up in the homotopic Frechét distance algorithm only because it uses the homotopy height as a subroutine. Thus, any constant factor approximation algorithm for the homotopy height problem implies a constant factor approximation algorithm for the homotopic Frechét distance.

Organization. We continue the paper by providing basic definitions in Section 2. Then, we consider the discrete version of the homotopy height problem in Section 3. This is how Chambers and Letscher formulated the problem. Later, in Section 4, we describe an algorithm to approximately find the shortest homotopy in continuous settings. In Section 5, we address the homotopic Frechét distance, for both the
2 Background

2.1 Planar Graphs

An embedding of a graph $G$ on the plane maps the vertices of $G$ to different points on the plane and its edges to disjoint paths except for the endpoints. The faces of an embedding are maximal connected subsets of the plane that are disjoint from the image of the graph. We use $\partial f$ to refer to the boundary of a single face $f$. The term plane graph refers to a graph together with its embedding on the plane.

The dual graph $G^*$ of a plane graph $G$ is the (multi-)graph whose vertices correspond to the faces of $G$, where two faces are joined by a (dual) edge if and only if their corresponding faces are separated by an edge of $G$. Thus, any edge $e$ in $G$ corresponds to a dual edge $e^*$ in $G^*$, any vertex $v$ in $G$ corresponds to a face $v^*$ in $G^*$ and any face $f$ in $G^*$ corresponds to a vertex $f^*$ in $G^*$.

Let $G = (V,E)$ be a simple undirected plane graph with edge weights $w : E \rightarrow \mathbb{R}^+$. A walk $W$ in $G$ is a sequence of vertices $(v_1,v_2,\ldots,v_k)$ such that each adjacent pair $e_i = (v_i,v_{i+1})$ is an edge in $G$. The length of $W$ is $||W|| = \sum_{i=1}^{k-1} w(e_i)$.

Let $v_i$ and $v_j$ be two vertices that appear on $W$. By $W[v_i,v_j]$ we mean the sub-walk of $W$ that starts at the first appearance of $v_i$ and ends at the first appearance of $v_j$ after $v_i$ on $W$. For two walks, $W_1 = (v_1,v_2,\ldots,v_i)$ and $W_2 = (v_i,v_{i+1},\ldots,v_j)$, we define their concatenation to be $W_1 \cdot W_2 = (v_1,v_2,\ldots,v_i,v_{i+1},\ldots,v_j)$.

A walk with distinct vertices is called a path. We use the terms $(u,v)$-walk to refer to a walk that starts at $u$ and ends in $v$; $(u,v)$-path is defined similarly. A walk is closed if its first and last vertices are identical. A closed path is a cycle. Two walks cross if and only if their images cross on the plane. That is, no infinitesimal perturbation makes them disjoint.

2.2 Piecewise Linear Surfaces and Geodesics

A piecewise linear surface is composed of a finite number of Euclidean triangles by identifying pairs of equal length edges. In this paper we work with piecewise linear surfaces that can be embedded in three dimensional space such that all triangles are flat and the surface does not cross itself. Equivalently, the surface can be presented by a set of edges and three dimensional coordinates of the vertices.

We say that a triangulated surface is non-degenerate if no interior point has curvature 0, i.e., when for every non-boundary vertex $x$, the sum of the angles of the triangles incident to $x$ is not equal to $2\pi$. We can turn any triangulated surface into a non-degenerate one by perturbing all edge lengths by a factor of at most $1 + \varepsilon$, for some $\varepsilon = O(1/n^2)$. This changes the metric of the surface by at most a factor of $1 + 1/n$, and thus the minimum height of a homotopy. Since such a factor will be negligible for our approximation guarantee, we can assume that the input surface is always non-degenerate.

A path $\gamma$ on the surface $\mathcal{D}$ is a function $\gamma : [0,1] \rightarrow \mathcal{D}$; $\gamma(0)$ and $\gamma(1)$ are the endpoints of the path. We use $||\gamma||$ to denote the length of $\gamma$. The path $\gamma$ is simple if and only if it maps the interval $[0,1]$ to distinct points on $\mathcal{D}$. A path is a geodesic if and only if it is locally a shortest path; i.e., it cannot be shortened by an infinitesimal perturbation. In particular, global shortest paths are geodesics. We use the term curve as an alternative for path. A path or a curve is polygonal if it is composed of a finite number of line segments.

Mitchell et al. [MMP87] describe an algorithm to compute the shortest path distance from a single source to the whole surface in $O(n^2 \log n)$ time. The same algorithm can be adapted to compute the
shortest path distance from an edge to the whole surface in the same running time. It follows that the shortest path from a set of $O(n)$ edges to the whole surface can be computed in $O(n^3 \log n)$.

Any shortest path in $D$ is a polygonal line that intersects every edge at most once and passes through a face along a segment. Moreover, the shortest path crossing an edge looks locally like a straight segment, if one rotates the adjacent faces so that they are coplanar. See [MMP87] for more details.

Let $S$ be a set of edges of $D$ and let $\pi$ be a shortest path from a point $p \in D$ to $S$. We define the signature of $\pi$, to be the ordered set of edges (crossed or used) by $\pi$. Since $\pi$ is locally a straight segment, we can rotate all faces that intersect $\pi$ one by one so that $\pi$ becomes a straight line. It follows that two geodesics with the same signature from $p$ are identical. A point $p$ on the surface is a medial point with respect to $S$ if there are more than one shortest paths (with different signatures) from $p$ to $S$.

### 2.3 Homotopy and Leash Function

Let $L$ and $R$ be two paths with the same endpoints $s$ and $t$ on a surface $D$. A homotopy $h : [0,1] \times [0,1] \to D$ is a continuous function, such that $h(0, \cdot) = L$, $h(1, \cdot) = R$, $h(\cdot, 0) = s$ and $h(\cdot, 1) = t$. So, for each $\tau \in [0,1]$, $h(\tau, \cdot)$ is an $(s,t)$-path. The height of such a homotopy (as defined previously) is defined to be $\sup_{\tau \in [0,1]} \|h(\tau, \cdot)\|$.

Let $A$ and $B$ be two disjoint curves. A curve connecting a point in $A$ to a point in $B$ is called an $(A,B)$-leash. We define a $(A,B)$-leash function to be a function $f$ that maps every $\tau \in [0,1]$ to a leash with endpoints $a(\tau) \in A$ and $b(\tau) \in B$ such that $a : [0,1] \to A$ and $b : [0,1] \to B$ are reparametrizations of $A$ and $B$, respectively. We say that an $(A,B)$-leash function $f$ is continuous if the leash $f(\tau)$ varies continuously with $\tau$. The height of a leash function $f$ is defined to be $\sup_{\tau \in [0,1]} \|f(\tau)\|$. Recall that the Frechet distance between $A$ and $B$ is the height of the minimum height $(A,B)$-leash function. The homotopic Frechet distance between $A$ and $B$ is the height of the minimum height continuous $(A,B)$-leash function.

### 2.4 Discrete Problems

Let $W_1$ be an $(s,t)$-walk and $f$ be a face in an embedded planar graph $G$. Assume that $\alpha_1$ is a subwalk of $W_1$ and $\partial f = \alpha_1 \cup \alpha_2$, where $\alpha_1$ and $\alpha_2$ are walks that share endpoints $u$ and $v$, such that $u$ is closer to $s$ on $W_1$. We define the face flip operation as follows. The walk $W_2 = W_1[s,u] \cdot \alpha_2 \cdot W_1[v,t]$ is the result of flipping $W_1$ over $f$. In this case, we say that $W_1$ and $W_2$ are one face flip operation apart. See Figure 2.1.

Now, let $W_1$ be an $(s,t)$-walk and $e = (u,v)$ be and edge in $G$. Assume that $u \in W_1$. We obtain the
walk \( W_2 = W_1[s,u] \cdot (u,v) \cdot (v,u) \cdot W_1[u,t] \) after applying a spike operation on \( W_1 \) along \( e \). In this case, we can obtain \( W_1 \) from \( W_2 \) by applying a reverse spike operation along \( e \). We say that \( W_1 \) and \( W_2 \) are a spike operation apart. In general, we say that \( W_1 \) and \( W_2 \) are one operation apart if we can transform one to the other using a single face flip, spike, or reverse spike. Letscher and Chambers introduce the same set of operations with the names: face lengthening, face shortening, spike and reverse spike.

Let \( L \) and \( R \) be two \((s,t)\)-walks on the outer face of \( G \). We define the sequence of walks \((L = W_0, W_1, \ldots, W_m = R)\) to be a \((L,R)\)-discrete homotopy if and only if for all \( 1 \leq i \leq m, W_i \) and \( W_{i-1} \) are one operation apart. We may use the word homotopy as a short form of discrete homotopy when it is clear from the context. A homotopy is monotonic (or equivalently it avoids backward moves) if \( W_{i-1} \) is inside the disc with boundary \( L \cup W_i \) for every \( 1 \leq i \leq m \). The height of the homotopy is defined to be the length of the longest walk in its sequence. The homotopy height between \( L \) and \( R \), is the height of the shortest possible \((L,R)\)-homotopy.

Let \( A = (a_0, a_1, \ldots, a_k) \) and \( B = (b_0, b_1, \ldots, b_l) \) be walks of \( G \). The walk \( W_1 = (a_i = w_1, w_2, \ldots, w_k = b_j) \) changes to the walk \( W_2 = (a_{i+1}, a_i = w_1, w_2, \ldots, w_k) \) after a person move. Similarly, the walk \( W_1 = (a_i = w_1, w_2, \ldots, w_k = b_j) \) changes to the walk \( W_2 = (w_1, w_2, \ldots, w_k = b_j, b_{j+1}) \) after a dog move. We say that the walk \( W_1 \) \( A \) leash operation is a dog move, a person move, a face flip, a spike or a reverse spike.

An \((A,B)\)-walk is a walk that has one endpoint on \( A \) and one endpoint on \( B \). A sequence of \((A,B)\)-walks, \((W_1, W_2, \ldots, W_q)\) is called an \((A,B)\)-leash sequence if

(i) \( W_1 \) is a \((a_0, b_0)\)-walk,
(ii) \( W_q \) is a \((a_k, b_l)\)-walk, and
(iii) we have that, for \( i = 1, \ldots, q-1, W_i \) changes to \( W_{i+1} \) by a set of leash operations that contains at most one move.

The height of a leash sequence is the length of its longest walk. The discrete Frechét distance of \( A \) and \( B \) is the height of the minimum height \((A,B)\)-leash sequence. The leash sequence \((W_1, W_2, \ldots, W_q)\) contains no gap if \( W_i \) changes to \( W_{i+1} \) by exactly one leash operation. The homotopic discrete Frechét distance of \( A \) and \( B \) is the height of the minimum height \((A,B)\)-leash sequence that contains no gap.

### 3 Approximating the height – the discrete case

In this section, we give an approximation algorithm for finding a homotopy of minimum height in a topological disk \( D \), whose boundary is defined by two walks \( L \) and \( R \) that share their endpoints \( s \) and \( t \). We start with the discrete case, i.e., when the disk is a triangulated edge-weighted planar graph. We use the ideas developed here in the continuous case, see Section 4.

#### 3.1 Settings

We are given an embedded planar graph \( G \) all of whose faces (except possibly the outer face) are triangles. Let \( s, t \in \partial G \) and \( L \) and \( R \) be the two non-crossing \((s,t)\)-walks on \( \partial G \) in counter-clockwise and clockwise order, respectively. We use \( D \) to denote the topological disk enclosed by \( L \cup R \). We refer to vertices of \( G \) (inside or on the boundary of \( D \)) as vertices of \( D \). Our goal is to find a minimum height homotopy from \( L \) to \( R \) of non-crossing walks. Recall that a homotopy is a sequence of walks, where every two consecutive walks differ by either a triangle, or an edge (being traversed twice).

**Lemma 3.1.** Let \( x \) and \( y \) be vertices of \( G \) that are at graph distance \( \rho \). Then any homotopy between \( L \) and \( R \) has height at least \( \rho \).
Proof: Fix a homotopy of height $\delta$. This homotopy contains an $(s, t)$-walk $\omega$ that passes through $x$, and an $(s, t)$-walk $\chi$ that passes through $y$. Let $d_G(x, y)$ be the distance between $x$ and $y$ in $G$. We have, by the triangle inequality, that $\rho = d_G(x, y) \leq \|s, x\| + \|s, y\|$, and, similarly, $\rho \leq \|x, t\| + \|y, t\|$. Therefore, $\rho \leq (\|\omega\| + \|\chi\|)/2 \leq \max(\|\omega\|, \|\chi\|) \leq \delta$, as required.  

Lemma 3.2. Suppose $d_1$ is the maximum distance of a vertex of $G$ from $L$, $d_2$ is the largest edge weight, and let $d_L = \max\{d_1, d_2\}$. Furthermore, let $D$, $L$, and $R$ be defined as above. Then any homotopy between $L$ and $R$ has height at least $d_L$.

Proof: For every homotopy between $L$ and $R$, and for every edge $e$, there exists a walk in the homotopy that passes through $e$. Therefore, the height of the homotopy is at least $d_2$. Moreover, the height is at least $d_1$ by Lemma 3.1.  

3.2 The algorithm

Theorem 3.3. Let $D$ be an edge-weighted triangulated topological disk with $n$ faces such that its boundary is formed by two walks $L$ and $R$ that share endpoints $s$ and $t$. Then, one can compute, in $O(n \log n)$ time, a homotopy from $L$ to $R$ of height at most $\|L\| + \|R\| + O(d_L \log n)$, where $d_L$ is the largest among (i) the maximum distance of a vertex of $D$ from $L$, and (ii) the maximum edge weight.

In particular, the generated homotopy has height $O(h_{\text{opt}} \log n)$, where $h_{\text{opt}}$ is the minimum homotopy height between $L$ and $R$.

Proof: Let $b(\delta, d_L, n)$ be the maximum possible height of a homotopy obtained by our algorithm for any disk of perimeter $\delta$ that is composed of $n$ faces and has $d_L$ as defined in the statement of theorem. We prove that $b(\delta, d_L, n) = \delta + O(d_L \log n)$, which in particular implies the theorem statement.

The base case $n = 0$ is easy. Indeed, if we have two edges $(u, v)$ and $(v, u)$ consecutive in $R$ (or in $L$) we can retract these two edges. By repeating this we arrive at both $L$ and $R$ being identical, and we are done. The case $n = 1$ is handled in a similar fashion. After one face flip, the problem reduces to the case $n = 0$. Hence, $b(\|L\| + \|R\|, d_L, 1) \leq \|L\| + \|R\| + d_L$.

For $n > 1$, compute for each vertex of $G$ its shortest path to $L$, and consider the set of edges $E$ used by all these shortest paths. Clearly, these shortest paths can be chosen so that $L \cup E$ form a tree. We consider each edge of $R$ to be “thick” and have two sides (i.e., we think about these edges as being corridors – this is done to guarantee that in the recursive scheme, done below, there are exactly two subproblems to each instance). If $E$ uses an edge of $R$ then it uses the inner copy of this edge, while $R$ uses the outer side. Similarly, we consider each original vertex of $R$ to be two vertices (one inside and the other one on the boundary $R$). The set $E$ would use only the inner vertices of $R$, while $R$ would use only the outer vertices. To keep the graph triangulated we also arbitrarily triangulate inside each thick edge of $R$ by adding corridor edges. Observe that, each corridor edge either connects two copies of a single vertex (thus has weight zero) or copies of two neighbors on $R$ (and so has the same weight as the original edge).

Clearly, if we cut $D$ along the edges of $E$, what remains is a simple triangulated polygon (it might have “thin” corridors along the edges of $R$). One can find a diagonal $uv$ such that each side of the diagonal contains at least $\lfloor n/3 \rfloor$ triangles and at most $\lfloor 2/3 \rfloor$ triangles of the original $G$. We emphasize that we count only the “real” triangles of $G$ – we conceptually assign zero weights to the faces within corridors. Observe that, because the faces inside corridors have weight zero, we can ensure that if the separating edge $uv$ is a corridor edge (i.e., corresponding to an edge $e$ of $R$) then $u$ and $v$ are copies of the same vertex. Indeed,
if not, we can change the separating edge so this property holds, and the new separating edge separates regions with the same weight, see the figure on the right. We use this property in the following case analysis, see Section 4.

(A) Consider the case that \( u \) and \( v \) are both vertices of \( R \). In this case, let \( R[u,v] \) be the portion of \( R \) in between \( u \) and \( v \), and let \( D_2 \) be the disk having \( R[u,v] \cup uv \) as its outer boundary. Let \( D_1 \) be the disk \( D \setminus D_2 \). Let \( M = R[s,u] \cup uv \cup R[v,t] \).

Clearly, the distance of any vertex of \( D_1 \) from \( L \) is at most \( d_L \). By induction, there is a homotopy of height \( b(||L|| + ||M||, d_L, (2/3)n) \) from \( L \) to \( M \). Similarly, the distance of any vertex of \( D_2 \) from \( uv \) is at most its distance to \( L \). Therefore, by induction, there is a homotopy between \( uv \) and \( R[u,v] \) of height at most \( f(||R[u,v]|| + d_L, d_L, (2/3)n) \). Clearly, we can extend this to a homotopy of \( M \) to \( R \) of height \( ||R[s,u]|| + f(||R[u,v]|| + d_L, d_L, (2/3)n) + ||R[v,t]|| \).

Putting these two homotopies together results in the desired homotopy from \( L \) to \( R \).

(B) Consider the case that \( v \) is a vertex of \( E \) and \( u \) is a vertex of \( R \). So, \( v \) is an inner vertex of \( R \) (that belongs to \( E \)) and \( u \) is an outer vertex of \( R \). Recall that we can assume that \( v \) and \( u \) are inner and outer copies of the same vertex of \( R \). Let \( \pi_v \) be the shortest path in \( D \) from \( v \) to \( L \), and let \( v' \) be its endpoint on \( L \).

Consider the disk \( D_1 \) having the “left” boundary \( L_1 = L[s,v'] \cup \pi_v \cup vu \) and \( R_1 = R[s,u] \) as its “right” boundary. This disk contains at most \((2/3)n \) triangles, and by induction, it has a homotopy of height \( b(||L_1|| + ||R_1||, d_L, (2/3)n) \). To see why we can apply the recursion, observe that \( u \) and \( v \) are copies of the same vertex of \( R \). That is, all shortest paths of vertices inside \( D_1 \) to \( L \) are completely inside \( D_1 \). Particularly, the distance of all vertices in \( D_1 \) to \( L_1 \) are at most \( d_L \).

Similarly, the topological disk \( D_2 \) with the left boundary \( L_2 = uv \cup \pi_v \cup L[v',t] \) and the right boundary \( R_2 = R[u,t] \) has a homotopy of height \( f(||L_2|| + ||R_2||, d_L, (2/3)n) \).

Starting with \( L \), extending a tendril from \( v' \) to \( v \), from \( v \) to \( u \), and then applying the homotopy to the first part of this walk (i.e., \( L_1 \)) to move to \( R_1 \), and then the homotopy of \( D_2 \) to the second part, results in a homotopy of \( L \) to \( R \) of height

\[
\max \left( \frac{||L|| + 2d_L}{f(||L_1|| + ||R_1||, d_L, (2/3)n) + ||L_2||}, \frac{||R_1|| + f(||L_2|| + ||R_2||, d_L, (2/3)n)}{3d_L} \right).
\]

If the first number is the maximum, we are done. Otherwise, the above value is at most \( b(||L|| + ||R|| + 2d_L, d_L, (2/3)n) \).

(C) Here we handle the case that \( u \) and \( v \) are both vertices of \( L \cup E \). Then, as before, let \( u' \) and \( v' \) be the closest points on \( L \) to \( u \) and \( v \), respectively. Now, let \( \pi_u \) (resp. \( \pi_v \)) be the shortest path from \( u \) (resp. \( v \)) to \( u' \) (resp. \( v' \)). Note that we might have \( u' = v' \).

Consider the disk \( D_1 \) having \( L_1 = [u',v'] \) as left boundary, and \( R_1 = \pi_u \cup uv \cup \pi_v \) as right boundary. This disk contains between \( n/3 \) and \( 2n/3 \) triangles of the original surface. The distance of any vertex of \( D_1 \) to \( L_1 \) (when restricted to \( D_1 \)) is at most \( d_L \), and therefore by induction, there is a homotopy from \( L_1 \) to \( R_1 \) of height at most \( \alpha = f(||L_1|| + ||R_1||, d_L, (2/3)n) \leq b(||L[u',v']|| + 3d_L, d_L, (2/3)n) \). This yields
a homotopy of height $\alpha_1 = \|L[s, u']\| + \alpha + \|L[v', t]\|$, from $L$ to $L_2 = L[s, u'] \cup \pi_u \cup uv \cup \pi_v \cup L[v', t]$. It is straightforward to check that $\alpha_1 \leq b(\|L\| + 3d_L, d_L, (2/3)n)$.

Next, let $D_2$ be the disk with its left boundary being $L_2$ and its right boundary being $R_2 = R$. Observe, that as before, the maximum distance of any vertex of $D_2$ to $L_2$ is at most $d_L$. As before, by induction, there is a homotopy from $L_2$ to $R_2$ of height $\alpha_2 = b(\|L_2\| + \|R_2\|, d_L, (2/3)n)$. Since $\|L_2\| \leq \|L\| + 3d$, we have $\alpha_2 \leq b(\|L\| + \|R\| + 3d_L, d_L, (2/3)n)$.

In all cases the length of the homotopy is at most

$$f(\|L\| + \|R\| + 3d_L, d_L, (2/3)n).$$

In general,

$$f(\delta, d_L, n) \leq f(\delta + 3d_L, d_L, (2/3)n).$$

Thus, a simple inductive argument shows that $b(\delta, d_L, n) = \delta + O(d_L \log n)$, as desired. The final guarantee of approximation follows as $d_L \leq h_{opt}$, by Lemma 3.2.

We can compute the shortest path tree in linear time using the algorithm of Henzinger et al. [HKRS97]. The separating edge can also be found in linear time using DFS. So, the running time for a graph with $n$ faces is $T(n) = T(n_1) + T(n_2) + O(n)$, where $n_1 + n_2 = n$ and $n_1, n_2 \leq (2/3)n$. It follows that $T(n) = O(n \log n)$.

**Remark 3.4.** (A) In the algorithm of Theorem 3.3, it is not necessary that we have the shortest paths from $L$ to all the vertices of $D$. Instead, it is sufficient if we have a tree structure that provides paths from any vertex of $D$ to $L$ of distance at most $d_L$ in this tree. We will use this property in the continuous case, where recomputing the shortest path tree is relatively expensive.

(B) A more careful analysis shows that the height of the homotopy generated by Theorem 3.3 is at most $\max(\|L\|, \|R\|) + O(d_L \log n)$.

(C) Note, that if $d_L = O(\max(\|L\|, \|R\|)/\log n)$ then Theorem 3.3 provides a constant factor approximation. This is the situation when $L$ and $R$ are close to each other compared to their relative length.

(D) Note, that the $O(n \log n)$ running time algorithm cannot explicitly output the list of paths in the homotopy. Indeed, that list requires $O(n^2)$ space to be stored and so $O(n^2)$ time to output. The output of the algorithm of the above lemma is a shortest path tree together with an ordered list of edges, each representing an $(s, t)$-walk.

### 4 Approximating the height – The continuous case

In this section we extend the algorithm to the continuous case. Here we are given a piecewise linear triangulated topological disk, $D$, with $n$ triangles. The boundary of $D$ is composed of two paths $L$ and $R$ with shared endpoints $s$ and $t$. Observe that the distance of any point $x$ in $D$ from $L$ and $R$ is not longer than the homotopy height as there is a $(s, t)$-path that contains $x$. Here, we build a homotopy of height at most $\|L\| + \|R\| + O(d \log n)$, where $d$ is the maximum distance of any point in $D$ from either $L$ or $R$.

We use the following observations (see Section 2 for details):

(A) The shortest path from a vertex to the whole surface can be computed in $O(n^2 \log n)$ time.

(B) The shortest path from a set of $O(n)$ edges to the whole surface can be computed in $O(n^3 \log n)$ time.

(C) A shortest path (i.e., a geodesic) intersects a face along a segment and it locally looks like a segment if the adjacent faces are rotated to be coplanar.
4.1 Homotopy height if edges are short

Here, we assume that the longest edge in $D$ has length at most $2d_L$, where $d_L$ is the maximum distance for any point of $D$ from $L$.

As in the discrete case, let $E$ be the union of all the shortest paths from the vertices of $D$ to $L$ (as before, we treat the edges and vertices of $R$ as having infinitesimal thickness). For a vertex $v$ of $D$, its shortest path $\pi_v$ is a polygonal path that crosses between faces (usually) in the middle of edges (it might also go to a vertex, merge with some other shortest paths and then follow a common shortest path back to $L$). In particular, each such shortest path might intersect a face of $D$ along a single segment. Thus, the polygon resulting from cutting $D$ along $E$, call it $P$, is a polygon that has complexity $O(n^2)$. A face of $P$ is a hexagon, a pentagon, a quadrilateral, or a triangle. However, each such facet has at most three edges that are portions of the edges of $D$. We say the degree of a face is $i$ if it has $i$ edges that are portions of the edges of $D$. Observe that, each triangle of $D$ is now decomposed into a set of faces. Obviously, each triangle of $D$ contains at most one face of degree 3 in $P$. Overall, there are $O(n)$ faces of degree 3 in $P$.

Now consider $C^*$, the dual of the graph that is inside the polygon (ignore the edges on the boundary). More precisely, $C^*$ has a vertex corresponding to each face inside the polygon $P$. Two vertices of $C^*$ are adjacent if and only if their corresponding faces share a portion of an edge of $D$ (this shared edge is a diagonal of $P$). Note that because $P$ is simply connected $C^*$ is a tree. Since the maximum degree of the tree $C^*$ is 3, there is an edge that is a good separator (i.e., a separator that has at most 2/3 of the faces on one side). Since the length of the edge is at most $2d_L$ it can be used in a similar fashion as the proof of Theorem 3.3. However, in the recursion of the continuous case we avoid recomputing the shortest paths (i.e., we use the old shortest paths and distances computed in the original disk), see Remark 3.4. So, we compute the shortest paths once in the beginning in $O(n^3 \log n)$ time. Then, in each step we can find the separator in $O(n^2)$ time. Namely, the total time spent on computing the separators is $T(n) = T(n_1) + T(n_2) + O(n^2)$, where $n_1 + n_2 = O(n^2)$ and $n_1, n_2 \leq (2/3)(n_1 + n_2)$; that is, $T(n) = O(n^3 \log n)$. As such, the total running time is dominated by the computation of the shortest paths. The output is a list of $O(n^2)$ paths each of complexity $O(n)$, and so it can be explicitly presented in $O(n^3)$ time and space. Note that, we need a continuous deformation between any two consecutive paths in the list, which can be implicitly presented by a collection of functions in linear time and space (this is similar to what we describe below in the beginning of Section 4.3).

The proof of Theorem 3.3 then goes through literally in this case. Since all the edges have length at most $2d_L$, by assumption, we obtain the following.

**Lemma 4.1.** Let $D$ be a topological disk with $n$ faces where every face is a triangle (here, the distance between any two points on the triangle is their Euclidean distance). Furthermore, the boundary of $D$ is formed by two walks $L$ and $R$ (that share two endpoints $s,t$). Let $d_L$ be the maximum distance of any point of $D$ from $L$. Furthermore, assume that all edges of $D$ have length at most $2d_L$. Then, one can compute a continuous homotopy from $L$ to $R$ of height at most $\|L\| + \|R\| + O(d_L \log n)$ in $O(n^3 \log n)$ time.

4.2 Breaking the disk into strips, pockets and chunks

For any two points in $D$ consider a shortest path $\pi$ connecting them. The signature of $\pi$ is the ordered sequence of edges (crossed or used) and vertices used by $\pi$, see Section 2. For a point $p \in R$, let $s_L(p)$ denote the signature of the shortest path from $p$ to $L$. The signature $s_L(p)$ is well defined in $R$ except for a finite set of medial points, where there are two (or more) distinct shortest paths from $L$ to $p$. In particular, let $\Pi_R$ be the set of all shortest paths from any medial point on $R$ to $L$. Observe that, the
medial points are the only points that the signature of the shortest path from R to L changes in any non-degenerate triangulation.

Cutting D along the paths of ΠR breaks D into corridors. If the intersection of a corridor with R is a point (resp. segment) then it is a delta (resp. strip). In a strip C, all the shortest paths to L from the points in the interior of the segment C ∩ R have the same signature. Intuitively, strips have a natural way to morph from one side to the other. We further break each delta into chunks and pockets, as follows.

So, consider a delta C with an apex c (i.e., the point of R on the boundary of C). For a point x ∈ L ∩ C, we define its signature (in relation to C), to be the signature of the shortest path from x to c (restricted to lie inside C). Again, we partition L ∩ C into maximum intervals that have the same signature, and let P be the set of endpoints of these intervals. For each point p ∈ P, consider all the different shortest paths from c to p inside the delta C, and cut C along these paths. This breaks C into regions. If a newly created region has a single intersection point with both L and R, then it is a pocket, otherwise, it is a chunk. Clearly, this process decomposes C into pockets and chunks.

Applying the above partition scheme to all the deltas results in a decomposition of D into strips, chunks and pockets.

4.2.1 Analysis

Let d be the maximum distance of any point of D to either L or R, and consider a chunk C. Its intersection with L is a segment, and its intersection with R is a point (i.e., the apex c of the delta). Observe that the distance of any point x ∈ L ∩ C to c is at most 2d. To see this, consider the shortest path πx from x to R in D, and observe that if it intersects the boundary of C then it can be modified to connect to c, and its new length is at most 2d. Hence, for a chunk C there is a natural way to morph L ∩ C to c.

A pocket, on the other hand, is a topological disk such that its intersections with L and R are both single points, and the two boundary paths between these intersections are of length at most 2d. Pockets are handled by using the recursive scheme developed for the discrete case.

4.3 Homotopy height if there are long edges

We use the above algorithm to break the given disk D into strips, chunks and pockets (notice, that we assume nothing on the length of the edges). Next, order the resulting regions according to their order along L, and transform each one of them at time, such that starting with L we end up with R.

(A) Morphing a chunk/strip S.
Let σL = L ∩ S and σR = R ∩ S, and let πt and πb be the top and bottom paths forming the two sides of S. There is a natural homotopy from πt ∪ σL to σR ∪ πb. The strip/chunk S has no vertex of D in its interior, and therefore it is formed by taking planar quadrilaterals and gluing them together along common edges.

Observe that by the triangle inequality, all such edges of any of these quadrilaterals are of length at most max(∥σL∥, ∥σR∥) + 4d.
It is now easy to check that we can collapse each such quadrilateral in turn to obtain the required homotopy. Since each of \( \pi_t \) and \( \pi_b \) is composed of two shortest paths, there is a linear number of such quadrilaterals, and each collapse can be done in constant time. See the figure for an example.

(B) Morphing a pocket: A pocket has perimeter at most \( 4d \), and there is a point on its boundary, such that the distance of any point in it to this base point is at most \( 2d \). By the triangle inequality, we have that if in a topological disk \( \mathcal{D} \) all the points of \( \mathcal{D} \) are in distance at most \( 2d \) from some point \( c \), then the longest edge in \( \mathcal{D} \) has length at most \( 4d \). Therefore, all the edges inside a pocket cannot be longer than \( 4d \). We can now apply Lemma 4.1 to such a pocket. This results in the desired homotopy.

4.3.1 Analysis

The shortest paths from \( \mathcal{R} \) to \( \mathcal{L} \) can be computed in \( O(n^3 \log n) \) time. The shortest paths inside a delta to its apex can be computed in \( O(n^2 \log n) \) time. Since there is a linear number of deltas, the total running time for building the strips is \( O(n^3 \log n) \).

**Lemma 4.2.** The number of paths in \( \Pi_\mathcal{R} \) is \( \Theta(|V(\mathcal{D})|) \), where \( V(\mathcal{D}) \) is the set of vertices of \( \mathcal{D} \).

**Proof:** Let \( \{\sigma_1, \sigma_2, \ldots, \sigma_k\} \) be the paths in \( \Pi_\mathcal{R} \) sorted by the order of their endpoints along \( \mathcal{R} \). Observe that these paths are geodesics and so one can assume that they are interior disjoint, or share a suffix. Now, if \( l_i \in \mathcal{L} \) and \( r_i \in \mathcal{R} \) are the endpoints of \( \sigma_i \), for \( i = 1, \ldots, k \), then these endpoints are sorted along their respective curves. In particular, let \( \mathcal{D}_i \) be the disk with boundary \( L[s, l_i] \cup \sigma_{i+1} \cup R[s, r_i] \). We have that \( \mathcal{D}_1 \subseteq \mathcal{D}_2 \subseteq \cdots \subseteq \mathcal{D}_k \). The signatures of \( \sigma_i \) and \( \sigma_{i+2} \) must be different as otherwise they would be consecutive. Furthermore, because of the inclusion property, if an edge or a vertex of \( \mathcal{D} \) intersects \( \sigma_i \) but does not intersect \( \sigma_{i+1} \) then, it cannot intersect any later path. Therefore, every other path in \( \Pi_\mathcal{R} \) can be charged to vertices or edges that are added or removed from the signature of the respective path. Since an edge or a vertex can be added at most once, and deleted at most once, this implies the desired bound on the number of paths. \( \blacksquare \)

Arguing as in Lemma 4.2, we have that the total number of parts (i.e., strips, chunks, and pockets) generated by the above decomposition is \( O(|V(\mathcal{D})|) \).

**Lemma 4.3.** Consider a strip or a chunk \( S \) generated by the above partition of \( \mathcal{D} \). Let \( \sigma_L = \mathcal{L} \cap S \) and \( \sigma_R = \mathcal{R} \cap S \). Let \( \pi_t \) and \( \pi_b \) be the top and bottom paths forming the two sides of \( S \) that do not lie on \( \mathcal{R} \) or \( \mathcal{L} \).

(A) We have \( \|\pi_b\| \leq 2d \) and \( \|\pi_t\| \leq 2d \).

(B) If \( \|\sigma_L\| > 0 \) or \( \|\sigma_R\| > 0 \) then there is no vertex of \( \mathcal{D} \) in the interior of \( S \).

(C) If \( \|\sigma_L\| > 0 \) or \( \|\sigma_R\| > 0 \) then there is a homotopy from \( \pi_t \cup \sigma_L \) to \( \sigma_R \cup \pi_b \) of height \( \max(\|\sigma_L\|, \|\sigma_R\|) + 4d \). We can compute such a homotopy in linear time.

**Proof:** (A) If the strip was generated by the first stage of partitioning then the claim is immediate.

Otherwise, consider a delta \( \mathcal{C} \) with an apex \( c \). For any point \( x \in \mathcal{L} \cap \mathcal{C} \) we claim that there is a path of length at most \( 2d \) to \( c \). Indeed, consider the shortest path \( \pi_x \) from \( x \) to \( \mathcal{R} \) in \( \mathcal{D} \). If this path goes to \( c \) the claim holds immediately. Otherwise, the shortest path (that has length at most \( d \)) must cross either the top or bottom shortest path forming the boundary of \( \mathcal{C} \) that are emanating from \( c \). We can now modify \( \pi_x \), so that after its intersection point with this shortest path, it follows it back to \( c \). Clearly, the resulting path has length at most \( 2d \) and lies inside the resulting chunk.

(B) Indeed, the boundary paths \( \pi_t \) and \( \pi_b \) have the same signature (formally, they are the limit of paths with the same signature). Since \( \mathcal{D} \) is non-degenerate, if there was any vertex in the middle, then
the path on one side of the vertex, and the path on the other side of the vertex cannot possibly have
the same signature.

(C) Immediate from the algorithm description.

4.4 The result

Theorem 4.4. Suppose that we are given a triangulated piecewise linear surface with the topology of a
disk, such that its boundary is formed by two walks \(L\) and \(R\). Then, there is a continuous homotopy from
\(L\) to \(R\) of height at most \(\|L\| + \|R\| + O(d \log n)\), where \(d\) is the maximum geodesic distance of any point
of \(D\) from either \(L\) or \(R\). This homotopy can be computed in \(O(n^3 \log n)\) time.

5 Approximating the Homotopic Frechét Distance

In this section, fix \(D\) to be a triangulated topological disk with \(n\) faces. Let the
boundary of \(D\) be composed of \(T, R, B,\) and \(L\), four internally disjoint walks appearing
in clockwise order along the boundary. Also, let \(t_L = L \cap T, b_L = L \cap B, t_r = R \cap T,\) and \(b_r = R \cap B\). See the figure on the right.

5.1 Approximating the Regular Frechét Distance

5.1.1 The continuous case

Let \(d_F(T, B)\) (resp. \(d_H(T, B)\)) be the regular (resp. homotopic) Frechét distance between \(T\) and \(B\). Clearly, \(d_F(T, B) \leq d_H(T, B)\). The following lemma implies that the Frechét distance can be approxi-
mated within a constant factor.

Lemma 5.1. Let \(D, n, L, T, R,\) and \(B\) be as above. Then, for the continuous case, one can compute,
in \(O(n^3 \log n)\) time, reparametrizations of \(T\) and \(B\) of width at most \(2\delta\), where \(\delta = d_F(T, B)\).

Proof: In the following, consider \(D\) to be the region bounded by these four curves \(L, T, R,\) and \(B\). We
decompose \(D\) into strips, chunks and pockets using the algorithm of Section 4.2. Let \(\Pi\) be the set of
shortest paths from all points of \(T\) to the curve \(B\). As in the algorithm of Section 4.2, let \(\Pi_T\) be the set of
all shortest paths from medial points on \(T\) to \(B\). Arguing as in Section 4.2, we have that the set \(\Pi_T\) is composed of a linear number of paths. The paths in \(\Pi_T\) do not cross and so partition \(D\) into a set of
regions. Each region is bounded by a portion of \(T\), a portion of \(B\) and two paths in \(\Pi_T\). A region is a
delta if the two paths of \(\Pi_T\) in its boundary share a single endpoint (on \(T\)), it is a pocket if they share
two endpoints (one on \(T\) and one on \(B\)), and it is strip if they share no endpoints. We refer to deltas,
pockets and strips as atomic regions.

Obviously, the (endpoints of the) paths in \(\Pi\) cover all of \(T\). The paths in \(\Pi\) also cover all of \(B\) except
for the bases of deltas. Now, for each delta we compute the set of all shortest paths from the vertices of
its base to its apex inside the delta. Let \(\Pi_B\) be the set of all such paths in all deltas. Clearly, the union
of \(\Pi_B\) and \(\Pi_T\) is a set of non-crossing paths whose endpoints cover all the vertices of \(T\) and \(B\).

The shortest path from any point of \(T\) to \(B\) is at most \(\delta\). So, all paths in \(\Pi\) have length at most \(\delta\).
Similarly, the shortest path from a point of \(B\) to \(T\) is at most \(\delta\). Now, consider a delta \(C\) with apex \(c\).
Let \(b\) be a point on the base of \(C\) (and so on \(B\)). The shortest path \(\pi_b\) from \(b\) to \(T\) has length at most \(\delta\).
Let \( x \) be the first point that \( \pi_b \) intersects a boundary path of \( C, \pi_C \). Now, \( \pi_b[b, x] \cdot \pi_C[x, c] \) has length at most \( 2\delta \) and it is inside \( C \). We conclude that all paths in \( \Pi_B \) have length at most \( 2\delta \).

The paths in \( \Pi_B \cup \Pi_T \) decompose \( D \) into strips and corridors. The left and right portions of a strip is of length at most \( 2\delta \), and its top and bottom sides have as such Frechét distance at most \( 2\delta \) from each other. Similarly, the leash can jump over a pocket from the left leash to the right leash. Doing this to all corridors and pockets, results in reparametrizations of \( L \) and \( R \) such that their maximum length of a leash for these reparametrizations are at most \( 2\delta \). This implies that the Frechét distance is at most \( 2\delta \), and we have an explicit reparametrization that realizes this distance.

As for the running time, in \( O(n^3 \log n) \) time, one can compute all shortest paths from \( T \) to the whole surface. Then, one can, in \( O(n^2 \log n) \) time, compute the shortest paths inside each of the linear number of deltas. It follows that the total running time is \( O(n^3 \log n) \).

\subsection{The discrete case}

We can use a similar idea to the decomposition into atomic regions as done in the proof of Lemma 5.1.

\begin{lemma}
Let \( D \) be a triangulated topological disk with \( n \) faces, and \( T \) and \( B \) be two internally disjoint walks on the boundary of \( D \). Then, for the discrete case one can compute, in \( O(n) \) time, reparametrizations of \( T \) and \( B \) that approximate the discrete Frechét distance between \( T \) and \( B \). The computed reparametrizations have width at most \( 3\delta \), where \( \delta \) is the Frechét distance between \( T \) and \( B \).
\end{lemma}

\begin{proof}
First, compute the set of shortest paths, \( \Pi_T = \{ \pi_1, \pi_2, \ldots, \pi_k \} \), from vertices of \( T \) to the path \( B \). To this end, we (conceptually) collapse all the vertices of \( B \) into a single vertex, and compute the shortest path from this meta vertex to all the vertices in \( T \).

Now, let \( \pi_i \) and \( \pi_{i+1} \) be two consecutive paths; that is, the endpoints of \( \pi_i \) and \( \pi_{i+1} \), \( a_i \) and \( a_{i+1} \), are adjacent vertices on \( T \). For all \( 1 \leq i < k \), we add the paths \( \pi_i^+ = (a_i, a_{i+1}) \cdot \pi_{i+1} \) to the set \( \Pi_T \) to obtain \( \Pi_T^+ \). Observe that each path in \( \Pi_T^+ \) has length at most \( 2\delta \); it is composed of zero or one edge of \( T \) and a shortest path from a vertex of \( T \) to \( B \). Further, \( \Pi_T^+ \) partitions the graph into regions, similar to the continuous case. Now for each vertex of \( B \) that is not an endpoint of a path in \( \Pi_T^+ \), we compute the shortest path inside its region to \( T \). Because the region is bounded by paths of length at most \( 2\delta \), the length of such a shortest path is at most \( 3\delta \). If \( \Pi_B \) is the set of all such shortest paths, then \( \Pi_T^+ \cup \Pi_B \) is a leash sequence of height at most \( 3\delta \).

We use the algorithm of Henzinger et al. [HKRS97] to compute the shortest paths from \( B \) in linear time. Since all regions are disjoint, and every edge appears on the boundary of at most two regions, we can compute all the shortest paths inside all these regions to \( T \) in \( O(n) \) time overall (this step requires careful implementation to achieve this running time).

\end{proof}

\begin{remark}
The paths realizing the Frechét distance computed by Lemma 5.2 are stored using an implicit data-structure (essentially two shortest path trees that are intertwined). This is why the space used is linear and why it can be constructed in linear time. Of course, an explicit representation of the sequence of walks realizing the Frechét distance might require quadratic space in the worst case.
\end{remark}

\subsection{Homotopic Frechét distance if there are no mountains}

The following lemma implies a \( O(\log n) \)-approximation algorithm for the case that all vertices in \( D \) are sufficiently close to both of the two curves.
Lemma 5.4. Let \( D \) be a triangulated topological disk with \( n \) faces, and \( T \) and \( B \) be two internally disjoint walks on the boundary of \( D \). Further, assume for all \( p \in D \), \( p \)'s distance to \( T \) is at most \( x \), and \( p \)'s distance to \( B \) is at most \( x \). Then, one can compute reparametrization of \( T \) and \( B \) of width \( O(x \log n) \). The running time is \( O(n^4 \log n) \) (resp. \( O(n^2) \)) in the continuous (resp. discrete) case.

In particular, if \( x = O(d_H(T, B)) \) then this is an \( O(\log n) \)-approximation to the optimal homotopic Frechét distance.

Proof: Consider the continuous case. Using the algorithm of Lemma 5.1 we compute a reparametrization of \( T \) and \( B \) of width \( \delta \), realizing approximately the regular Frechét distance, where \( \delta = O(x) \). Let \( \ell(t) \) denote the leash at time \( t \) that we obtain from the reparametrization mentioned above. Note that the leash \( \ell(\cdot) \) is not required to deform continuously in \( t \). In particular, for a given time \( t \in [0, 1] \), let \( \ell^-(t) = \lim_{\tau \to t-} \ell(\tau) \) and \( \ell^+(t) = \lim_{\tau \to t+} \ell(\tau) \), where \( \lim_{\tau \to t-} \) and \( \lim_{\tau \to t+} \) are the left-sided and right-sided limits, respectively. By definition, the leash is discontinuous at \( t \) if and only if \( \ell^-(t) \neq \ell^+(t) \).

Naturally, the above reparameterization can be used as long as it is continuous. Whenever the leash jumps over a gap (i.e., the leash is discontinuous at this point in time), say at time \( t \), we are going to replace this jump by a \((\ell^-(t), \ell^+(t))\)-homotopy between the two leashes. Clearly, this would result in the desired continuous homotopy.

To this end, observe that all the vertices inside the disk with boundary \( \ell^-(t) \cup \ell^+(t) \) have distance \( O(x) \) to \( T \) and \( B \), and thus also to \( \ell^-(t) \) and \( \ell^+(t) \). Hence, using the algorithm of Theorem 4.4, compute an \((\ell^-(t), \ell^+(t))\)-homotopy with height \( O(x \log n) \). Since a gap must contain a vertex there are \( O(n) \) gaps, so this filling in is done at most \( O(n) \) times. Computing the initial reparameterization takes \( O(n^3 \log n) \) time. Each gap can be filled in \( O(n^3 \log n) \) time.

The discrete case is similar. The Frechét distance here can be computed in linear time using the algorithm of Lemma 5.2 (see also Remark 5.3). However, we can only obtain the value of the Frechét distance as well as an implicit representation of the actual deformation in linear time. Indeed we can compute an explicit listing of the paths in \( O(n^2) \) time. Each path in the list can be charged to a single face or edge of \( D \). It immediately follows that the number of paths is linear. For any two consecutive paths, \( \pi_i \) and \( \pi_{i+1} \) in the list, we can fill in the possible gap and compute the explicit solution in \( O(n^2) \) time, where \( n_i \) is the number of faces between \( \pi_i \) and \( \pi_{i+1} \), see Theorem 3.3 and Remark 3.4 (D). Since \( \sum n_i = O(n) \) the total running time of the algorithm is \( O(n^2) \).

The above lemma demonstrates that if the starting and ending leashes are known (i.e., the region of the disk \( D \) swept over by the morph) then an approximation algorithm can be obtained. The challenge is that a priori, we do not know these two leashes, as the input is a topological disk \( D \) with the two curves \( T \) and \( B \) on its boundary, and the start/end leashes might be curves that lie somewhere in the interior of \( D \).

5.3 A Decision Procedure for the Homotopic Frechét distance in the presence of mountains

For a parameter \( \tau \geq 0 \), a vertex \( v \in V(D) \) is \( \tau \)-tall if and only if its distance to \( T \) or \( B \) is larger than \( \tau \) (intuitively \( \tau \) is a guess for the value of \( d_H(T, B) \)). Here, we consider the case where there are \( \tau \)-tall vertices. Intuitively, one can think about tall vertices as insurmountable mountains. Thus, to find a good homotopy between \( T \) and \( B \), we have to choose which “valleys” to use (i.e., what homotopy class the solution we compute belongs to if we think about tall
vertices as punctures in the disk). As a concrete example, consider the figure on the right, where there are three tall vertices, and two possible solutions are being shown.

In the discrete case, we subdivide each edge in the beginning so that if an edge has length \( > 2\tau \), then the vertex inserted in the middle of it is \( \tau \)-tall. Observe that, if \( \tau > d_H(T, B) \) then no leash of the optimum homotopic motion can afford to contain a \( \tau \)-tall vertex. We use \( M^\tau \) to denote the set of all \( \tau \)-tall vertices in \( V(D) \).

Now, let \( \omega \) and \( \omega' \) be two walks connecting points on \( T \) and \( B \). We say that \( \omega \) and \( \omega' \) are \textit{homotopic} in \( D \setminus M^\tau \) if and only if they are homotopic in \( D \setminus M^\tau \) after contracting \( T \) and \( B \) (each to a single point).

Two \textit{non-crossing} walks \( \omega \) and \( \omega' \) are homotopic if and only if \( T \cup B \cup \omega \cup \omega' \) contains no tall vertices. It is straightforward to check that homotopy is an equivalence relation. So it partitions \( (T, B) \)-paths into \textit{homotopy classes}; we call each class a \( \tau \)-homotopy class or simply a homotopy class (given that \( \tau \) is fixed).

For a homotopy class \( h \), let \( \pi_{L,h} \) (resp. \( \pi_{R,h} \)) be the \textit{left geodesic} (resp. \textit{right geodesic}); that is, \( \pi_{L,h} \) denotes the shortest path in \( h \) from \( t_l \) to \( b_l \) (resp. from \( t_r \) to \( b_r \)).

Let \( \omega \) be any walk in \( h \) from \( b \in B \) to \( t \in T \). We define the \textit{left tall set} of \( \omega \), denoted \( M_l(\omega) = M_l(\omega) \) to be the set of all \( \tau \)-tall vertices to the left of \( \omega \). Namely, \( M_l(\omega) \) is the set of tall vertices inside the disk with boundary \( \left[L \cup T[t, t] \cup \omega \cup B[b_l, b_r]\right] \), where \( L \) is the “left” portion of the boundary of \( D \), having endpoints \( t_l \) and \( b_l \). We similarly define the \textit{right tall set} of \( h \), \( M_r(h) = M_r(\omega) \), to be the set of all \( \tau \)-tall vertices to the right of \( \omega \). See the figure on the right.

Note that the sets \( M_l(\omega) \) and \( M_r(h) \) do not depend on the particular choice of \( \omega \), since all paths in \( h \) are homotopic and so have the same set of \( \tau \)-tall vertices to their left and right side. However, we emphasize that the left and right tall sets do not identify homotopy classes. The figure on the right demonstrates two non-homotopic paths with identical left and right tall sets.

We say that \( h \) is \( \tau \)-\textit{extendable} from the left if and only if \( \| \pi_{L,h} \| \leq \tau \) and there is a homotopy class \( h' \), such that \( \| \pi_{L,h'} \| \leq \tau \) and \( M_l(h) \subset M_l(h') \). In particular, \( h \) is \( \tau \)-\textit{saturated} if it is not \( \tau \)-extendable and \( \| \pi_{L,h} \| \leq \tau \).

### 5.3.1 On the left and right geodesics

**Lemma 5.5.** Let \( h \) be a \( \tau \)-saturated homotopy class, where \( \tau \geq d_H(T, B) \). Then, \( \| \pi_{R,h} \| \leq 4\tau \).

**Proof:** Let \( h_{\text{opt}} \) be the homotopy class of the leashes in the optimum solution. Of course, no leash in the optimum solution contains a \( \tau \)-tall vertex. Further, all leashes in the optimal solution are homotopic because there is a homotopy that contains all of them by definition.

Since \( h \) is saturated, the set \( M_l(h) \) is not a proper subset of \( M_l(h_{\text{opt}}) \). It follows that either \( M_l(h) = M_l(h_{\text{opt}}) \) or \( M_l(h) \) intersects \( M^\tau \setminus M_l(h_{\text{opt}}) = M_r(h_{\text{opt}}) \).

If \( M_l(h) = M_l(h_{\text{opt}}) \) then either \( h = h_{\text{opt}} \), and in particular \( \| \pi_{R,h} \| = \| \pi_{R,h_{\text{opt}}} \| \leq \tau \), or \( \pi_{L,h} \) crosses \( \pi_{R,h_{\text{opt}}} \).

Otherwise, the set \( M_l(h) \cap M_r(h_{\text{opt}}) \) is not empty. Again, it follows that \( \pi_{L,h} \) crosses \( \pi_{R,h_{\text{opt}}} \).

Therefore, we only need to address the case that \( \pi_{L,h} \) crosses \( \pi_{R,h_{\text{opt}}} \).

Let \( x \) be the first intersection point between \( \pi_{L,h} \) and \( \pi_{R,h_{\text{opt}}} \), as one traverses \( \pi_{L,h} \) from \( t_l \) to \( b_l \). Let \( x' \) be the last intersection point of \( \pi_{L,h}[x, x] \) with \( \pi_{L,h_{\text{opt}}} \). Similarly, \( y \) is the last intersection point between \( \pi_{L,h} \) and \( \pi_{R,h_{\text{opt}}} \), and \( y' \) is the
first intersection of \( \pi_{L,h}[y, b_i] \) and \( \pi_{L,h,\text{opt}} \). Observe that the interiors of \( \pi_{L,h}[x', x] \) and \( \pi_{L,h}[y, y'] \) do not intersect the curves \( \pi_{L,h,\text{opt}} \) and \( \pi_{R,h,\text{opt}} \).

As the curves \( \pi_{L,h} \) and \( \pi_{R,h} \) are homotopic (by definition), the disk with the boundary \( T \cdot \pi_{L,h} \cdot B \cdot \pi_{R,h} \) does not contain any tall vertex, and \( T \cdot \pi_{L,h} \cdot B \) is homotopic to \( \pi_{R,h} \).

Consider the walk \( T' = \pi_{R,h,\text{opt}}[t_r, x] \cdot \pi_{L,h}[x, x'] \cdot \pi_{L,h,\text{opt}}[x', t_l] \). The walk \( T' \) is homotopic to \( T \). Similarly, \( B' = \pi_{L,h,\text{opt}}[b_t, y'] \cdot \pi_{L,h}[y', y] \cdot \pi_{R,h,\text{opt}}[y, b_r] \) is homotopic to \( B \). It follows that \( \pi_{R,h} \) is homotopic to \( T', \pi_{L,h} \cdot B' \). As \( \pi_{R,h} \) is the shortest path in its homotopy class with these endpoints, it follows that

\[
\|\pi_{R,h}\| \leq \|T' \cdot \pi_{L,h} \cdot B'\| \leq \|\pi_{L,h}\| + (\|\pi_{L,h,\text{opt}}\| + \|\pi_{L,h}\| + \|\pi_{R,h,\text{opt}}\|) \leq 4\tau,
\]

as \( T' \) and \( B' \) are disjoint, and \( T' \cup B' \subseteq \pi_{R,h,\text{opt}} \cup \pi_{L,h,\text{opt}} \cup \pi_{L,h} \).

A region that contains no \( \tau \)-tall vertices can still, potentially, contain \( \tau \)-tall points (that are not vertices) on its edges or faces. We next prove that this does not happen in our setting.

**Lemma 5.6.** For any \( \tau \geq 0 \), let \( h \) be a \( \tau \)-homotopy class, such that \( \max(\|\pi_{L,h}\|, \|\pi_{R,h}\|) \leq x \), where \( x \geq \tau \geq d_{h}(T, B) \). Let \( D' \) be the disk with boundary \( T \cdot \pi_{R,h} \cdot B \cdot \pi_{L,h} \). Then, all the points inside \( D' \) are within distance \( O(x) \) to both \( T \) and \( B \) in \( D' \).

**Proof:** We first consider the continuous case. By the definition of \( \tau \)-homotopy, the disk \( D' \) has no \( \tau \)-tall vertices. Furthermore, by the definition of \( \tau \), we have that the distance of any point on \( T \) to \( B \), restricted to paths in \( D' \) is at most \( \delta_1 \), where \( \delta_1 = x + d_{x}(T, B) \leq 2x \). Indeed, the shortest path from any point on \( T \) to \( B \) in \( D \), either stays inside \( D' \), or alternatively intersects either \( \pi_{L,h} \) or \( \pi_{R,h} \).

We can now deploy the decomposition of \( D' \) into strips, pockets and chunks as done in Section 4.2. Every strip (or a chunk) is being swept by a leash of length at most \( \delta_2 = 2\delta_1 \leq 4x \) (the factor two is because a strip might rise out of a delta), and therefore the claim trivially holds for points inside such regions.

Every pocket \( P \) has perimeter of length at most \( \|\partial P\| \leq \delta_3 = 2\delta_2 = 8x \) (the perimeter also contains two points of \( T \) and \( B \) and they are in distance at most \( \delta_2 \) from each other in either direction along the perimeter).

So, consider such a pocket \( P \). Since \( D' \) contains no \( \tau \)-tall vertices, \( P \) does not contain any tall vertex. Let \( e \) be an edge in \( P \) (or a subedge if it intersects the boundary of \( P \)). The two endpoints of \( e \) are in \( P \), and such an endpoint is either a (not tall) vertex or it is contained in \( \partial P \). In either case, these endpoints are in distance at most \( x \) from \( \partial P \), and so they are in distance at most \( \delta_4 = 2x + \|\partial P\|/2 = 2x + \delta_2 \leq 6x \) from each other even if the geodesic distance is restricted to \( P \). We conclude that \( \|e\| \leq \delta_4 \), and consequently, any point in \( e \) is in distance at most \( \delta_5 = \|e\|/2 + x + \delta_2 \leq 3x + x + 8x \leq 12x \) from \( T \) and \( B \).

Now, consider any point \( p \) in \( P \), and consider the face \( F \) that contains it. Since the surface is triangulated, \( F \) is a triangle. Clipping \( F \) to \( P \) results in a planar region \( F' \) that has perimeter at most \( \delta_6 = 3\delta_4 + \|\partial P\| \leq 3 \cdot 6x + \delta_3 \leq (18 + 8)x \leq 26x \) (note, that an edge might be fragmented into several subedges, but the distance between the furthest two points along a single edge is at most \( \delta_4 \) using the same argument as above). Thus, the furthest a point of \( P \) can be from an edge of \( P \) is at most \( \delta_7 = \delta_6/2\pi \leq 5x \). Hence, the maximum distance of a point of \( P \) from either \( T \) or \( B \) (inside \( D' \)) is at most \( \delta_5 + \delta_7 \leq 12x + 5x = 17x \).

The discrete case is easy. Any edge of length \( \geq 2\tau \) was split, by introducing a middle vertex, which must be \( \tau \)-tall. So the claim immediately holds.
5.3.2 The decision algorithm

**Lemma 5.7.** Let $D, n, T, L, B, R, t_i, b_i$ as in the first paragraph of Section 5 and $\tau$ as in the previous subsection, and let $X \subseteq V(D)$ be a set of $\tau$-tall vertices. Consider the shortest path $\sigma$ (between $t_i$ and $b_i$) that belongs to any homotopy class $h$ such that $X \subseteq M_i(h)$. Then, the path $\sigma$ can be computed in $O(n^4 \log n)$ (resp. $O(n \log n)$) time in the continuous (resp. discrete) case.

**Proof:** For each vertex of $v \in X$, compute its shortest path $\psi_v$ to $L$ in $D$. Cut the disk $D$ along these paths. The result is a topological disk $D'$. Compute the shortest path $\zeta$ in $D'$ between $t_i$ and $b_i$.

We claim that $\zeta = \sigma$. To this end, consider $\sigma$ and any path $\psi_v$ computed by the algorithm. We claim that $\sigma$ and $\psi_v$ do not cross in their interior. Indeed, if $\sigma$ cross $\psi_v$ an odd number of times, then $v$ is inside the disk $\sigma \cdot T \cdot R \cdot B$, which contradicts the condition that $v \in X \subseteq M_i(h)$. Clearly, $\sigma$ and $\psi_v$ cannot cross in their interiors more than once, because otherwise, one can shorten one of them, which is a contradiction as they are both shortest paths. Thus, $\sigma$ is a path in $D'$ connecting $t_i$ to $b_i$, thus implying that $\zeta$ is $\sigma$.

As for the running time, each shortest path computation takes time $O(n^2 \log n)$, in the continuous (resp. discrete) case. The resulting disk has complexity $O(n^2)$, and computing a shortest path in it takes $O(n^2 \log n)$ time in the continuous case. In the discrete case, computing the paths can be done by collapsing $L$ to a vertex, forbid the shortest path tree edges, and run a shortest path algorithm in the remaining graph. Clearly, this takes $O(n \log n)$ time.

**Lemma 5.8.** Let $D$ be a triangulated topological disk with $n$ faces, and $T$ and $B$ be two internally disjoint walks on $D$’s boundary. Given $\tau > 0$, one can compute a $\tau$-saturated homotopy class, in $O(n^5 \log n)$ (resp. $O(n^2 \log n)$) time, in the continuous (resp. discrete) case.

**Proof:** Start with an empty initial set $X = \emptyset$. At each iteration, try adding one of the $\tau$-tall vertices $v \in M^r$ of $D$ to $X$, by using Lemma 5.7. The algorithm of Lemma 5.7 outputs a path $\sigma$ between $t_i$ and $b_i$ and a set $X' \supseteq X \cup \{v\}$.

If $\sigma$ is of length at most $\tau$ update $X$ to be the new set $X'$, otherwise reject $v$. If $v$ is rejected then the left geodesic of any superset of $X \cup \{v\}$ has length larger than $\tau$. It follows that $v$ cannot be accepted in any later iteration, so we do not need to reinspect it. Clearly, after trying all the vertices of $M^r$, the set $X$ defines the desired saturated class, which can be computed by using the algorithm of Lemma 5.7.

**Lemma 5.9.** Let $D$ be a triangulated topological disk with $n$ faces, and $T$ and $B$ be two internally disjoint walks on the boundary of $D$. Given a real number $x > 0$, one can either:

(A) Compute a homotopy from $T$ to $B$ of width $O(x \log n)$, or

(B) Return that $x < d_H(T, B)$.

The running time of this procedure is $O(n^5 \log n)$ (resp. $O(n^2 \log n)$) in the continuous (resp. discrete) case.

**Proof:** Assume $x \geq \delta_H = d_H(T, B)$, and we use $x$ as a guess for this value $\delta_H$. Using Lemma 5.8, one can compute a $x$-saturated homotopy class, $h$. Lemma 5.5 implies that both $\pi_{L,h}$ and $\pi_{R,h}$ are at most $4x$. Let $D' \subseteq D$ be the disk with boundary $T \cup \pi_{L,h} \cup B \cup \pi_{R,h}$. By Lemma 5.6, all the vertices in $D'$ are in distance $O(x)$ from $T$ and $B$ (this holds for all points in $D'$ in the continuous case). That is, there are no $O(x)$-tall vertices in $D'$. Finally, Lemma 5.4 implies that a continuous leash sequence of height $\leq Z = O(x \log n)$ between $T$ and $B$, inside $D'$, can be computed.

Thus, if $x$ is larger than $d_H(T, B)$ then this algorithm returns the desired approximation; that is, a homotopy of width $\leq Z$. If the width of the generated homotopy is however larger than $Z$ (a value that can be computed directly from $x$), then the value of $x$ was too small. That is, the algorithm fails in this case only if $x < d_H(T, B)$. In the case of such failure, return that $x$ is too small.
5.4 A strongly polynomial approximation algorithm

For a vertex \( v \in V(D) \), define \( \text{cost}(v) \) to be the length of the shortest path between \( t \) and \( b \) that has \( v \) on its left side. Similarly, for a set of vertices \( X \subseteq V(D) \), let \( \text{Cost}(X) \) be the length of the shortest path between \( t \) and \( b \) that has \( X \) on its left side. For a specific \( v \) or \( X \), one can compute \( \text{cost}(v) \) and \( \text{Cost}(X) \) by invoking the algorithm of Lemma 5.7 once.

5.4.1 The algorithm

(I) **Identifying the tall vertices.** Observe that using the algorithm of Lemma 5.9, we can decide given a candidate value \( \delta_H \) for \( d_H(T, B) \) if it is too large, too small, or leads to the desired approximation. Indeed, if the algorithm returns an approximation of values \( O(\delta_H \log n) \) but fails for \( \delta_H / 2 \), we know it is the desired approximation.

So, compute for each vertex \( v \in V(D) \) its tallness; that is \( \alpha_v \) would be the maximum distance of \( v \) to either \( T \) or \( B \). Sort these values, and using binary search, compute the vertex \( w \), with the minimum value \( \alpha_w \), such that Lemma 5.9 returns a parameterization with homotopic Frechet distance \( O(\alpha_w \log n) \). In this case, we can use binary search to find an interval \([\gamma/2, \gamma]\) that contains \( \delta_H \) and use Lemma 5.9 to obtain the desired approximation. Similarly, if \( v \) is the tallest vertex shorter than \( w \), then we can assume that \( \alpha_v \) is too small of a guess, otherwise we are again done as \([\alpha_v, \alpha_w n]\) contains \( \delta_H \).

Therefore, in the following, we know that the desired distance \( \delta_H \) lies in the interval \([x, y]\) where \( x = \alpha_v n \) and \( y = \alpha_w / n \), and for every vertex \( u \) of \( D \) it holds that (i) \( \alpha_u \leq x / n \), or (ii) \( \alpha_u \geq y n \). Naturally, we consider all the vertices that satisfy (ii) as tall vertices, by setting \( \tau = 2x / n \). In the following, let \( M \) denote the set of these \( \tau \)-tall vertices.

(II) **Computing candidate homotopy classes.** Start with \( X_0 = \emptyset \). In the \( i \)th iteration, the algorithm computes the vertex \( v_i \in M \setminus X_{i-1} \), such that \( \text{Cost}(X_{i-1} \cup \{v_i\}) \) is minimized, and set \( X_i = X_{i-1} \cup \{v_i\} \). Let \( h_i \) be the homotopy class having \( X_i \) on its left side, and \( M \setminus X_i \) on its right side.

(III) **Binary search over candidates.** We approximate the homotopic Frechet width of each one of the classes \( h_1, \ldots, h_n \). Let \( x \) be the minimum homotopic Frechet width computed among these \( n \) candidates.

Next, do a binary search in the interval \([x/n^2, x]\) for the homotopic Frechet distance. We return the smallest width reparametrization computed as the desired approximation.

5.4.2 Analysis

**Lemma 5.10.** (i) For any \( X' \subseteq X \subseteq V(D) \), we have \( \text{Cost}(X') \leq \text{Cost}(X) \).

(ii) For any \( x \in X \subseteq V(D) \), we have \( \text{cost}(x) \leq \text{Cost}(X) \).

(iii) For \( X, Y \subseteq V(D) \), we have that \( \text{Cost}(X \cup Y) \leq \text{Cost}(X) + \text{Cost}(Y) \).

**Proof:** (i) Observe that the path realizing \( \text{Cost}(X') \) is less constrained than the path realizing \( \text{Cost}(X) \), therefore it might only be shorter.
(ii) Follows immediately from (i).

(iii) Consider the disk $D$ and the two paths $\sigma_X$ and $\sigma_Y$ realizing $\text{Cost}(X)$ and $\text{Cost}(Y)$, respectively. The close curves $\sigma_X \cup L$ and $\sigma_Y \cup L$ encloses two topological disks. Consider the union of these two disks, and its connected outer boundary $\sigma_{X \cup Y} \cup L$. Clearly, $\sigma_{X \cup Y}$ connects $t$ and $b$, and it has all the points of $X$ and $Y$ on one side of it, and finally $\|\sigma_{X \cup Y}\| \leq \|\sigma_X\| + \|\sigma_Y\|$ as $\sigma_{X \cup Y} \subseteq \sigma_X \cup \sigma_Y$. See the figure on the right.

\[\text{Lemma 5.11.} \] The cheapest homotopic Frechet parameterization computed among $h_1, \ldots, h_n$ has width $O(d_H(T, B) n \log n)$.

\[\text{Proof:}\] Consider the set $Y$ that is the subset of tall vertices on the left side of the optimal solution. Let $i$ be the first index such that $Y \subseteq X_i$ and $Y \not\subseteq X_{i-1}$. Let $v$ be any vertex in $Y \setminus X_{i-1}$. By construction, we have that $\text{Cost}(X_i) \leq \text{Cost}(X_{i-1} \cup \{v\})$, and furthermore, for all $j \leq i$, we have that $\text{Cost}(X_j) \leq \text{Cost}(X_{j-1} \cup \{v\})$, by the greediness in the construction of $X_1, \ldots, X_i$. Now, we have

\[
\begin{align*}
\text{Cost}(X_i) & \leq \text{Cost}(X_{i-1} \cup \{v\}) & (\text{by construction of } X_i) \\
& \leq \text{Cost}(X_{i-1}) + \text{cost}(v) & (\text{by Lemma 5.10 (iii)}) \\
& \leq \text{Cost}(X_{i-1}) + \text{Cost}(Y) & (\text{by Lemma 5.10 (ii)}) \\
& \leq (\text{Cost}(X_{i-2}) + \text{Cost}(Y)) + \text{Cost}(Y) & (\text{applying same argument to } X_{i-1}) \\
& = \text{Cost}(X_{i-2}) + 2\text{Cost}(Y) \\
& \leq \cdots \leq i\text{Cost}(Y) \leq n\text{Cost}(Y).
\end{align*}
\]

Now, setting $\tau = \text{Cost}(X_i)$, it follows that $X_i$ is $\tau$-saturated. Applying Lemma 5.5, implies that $\|\pi_{R, h_i}\| \leq 4\tau$. Observe, that the disk defined by $T$, $\pi_{L, h_i}$, $B$, $\pi_{R, h_i}$ cannot contain any tall vertex (by construction).

Now, plugging this into Lemma 5.4 implies the homotopic Frechet width of $h_i$ (starting with $\pi_{L, h_i}$ and ending up with $\pi_{R, h_i}$), so $D$ in Lemma 5.4 is bounded by $T, B, \pi_{L, h_i}, \pi_{R, h_i}$ is $O(\tau \log n)$, which implies the claim since $\text{Cost}(X_i) \leq n\text{Cost}(Y) \leq nd_H(T, B)$.

\[\text{5.4.3 The algorithm}\]

\[\text{Theorem 5.12.} \] Let $D$ be a triangulated topological disk with $n$ faces, and $T$ and $B$ be two internally disjoint walks on the boundary of $D$. One can compute a homotopic Frechet parameterization of $T$ and $B$ of width $O(d_H(T, B) \log n)$, where $d_H(T, B)$ is the homotopic Frechet distance between $T$ and $B$ in $D$.

The running time of this procedure is $O(n^6 \log n)$ (resp. $O(n^3 \log n)$) in the continuous (resp. discrete) case.

\[\text{Proof:}\] Consider the algorithm described in the previous subsection. For correctness, observe that the algorithm either found the desired value, or identified correctly the tall vertices. Next, by Lemma 5.11, the range the algorithm searches over contains the desired value.

The algorithm requires $O(n^2)$ calls to Lemma 5.7, which takes $O(n^6 \log n)$ (resp. $O(n^3 \log n)$) time in the continuous (resp. discrete) case. Then, the algorithm requires the method of Lemma 5.4 to compute the homotopic Frechet distance of the classes $h_1, \ldots, h_n$. The algorithm also performs $O(\log n)$ calls to the algorithm of Lemma 5.9.
6 Conclusions

We presented a $O(\log n)$ approximation algorithm for approximating the homotopy height and the homotopic Frechet distance between curves on piecewise linear surfaces. It seems quite believable that the approximation quality can be further improved, and we leave this as the main open problem. Since our algorithm works both for the continuous and discrete cases, it seems natural to conjecture that this algorithm should also work for more general surfaces and metrics.

Connection to planar separator. Our basic algorithm (Theorem 3.3) is inspired to some extent by the proof of the planar separator theorem [LT79]. In particular, our result implies sufficient conditions to having a separator that can continuously deform from enclosing nothing in a planar graph, till it encloses the whole graph, without being too long at any point in time. As a result, our work can be viewed as extending the planar separator theorem. A natural open problem is to extend our work to graphs with higher genus.

Acknowledgments The authors thank Jeff Erickson and Gary Miller for their comments and suggestions. The authors also thank the anonymous referees for the detailed reviews.

References

[AB05] H. Alt and M. Buchin. Semi-computability of the Frechet distance between surfaces. In Proc. 21st Euro. Workshop on Comput. Geom., pages 45–48, 2005.

[AG95] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. IJCGA, 5:75–91, 1995.

[BBG08a] K. Buchin, M. Buchin, and J. Gudmundsson. Detecting single file movement. In Proc. 16th ACM SIGSPATIAL Int. Conf. Adv. GIS, pages 288–297, 2008.

[BBG+08b] K. Buchin, M. Buchin, J. Gudmundsson, Maarten L., and J. Luo. Detecting commuting patterns by clustering subtrajectories. In Proc. 19th Annu. Internat. Sympos. Algorithms Comput., pages 644–655, 2008.

[BDS13] M. Buchin, A. Driemel, and B. Speckmann. Computing the Fréchet distance with shortcuts is np-hard. CoRR, abs/1307.2097, 2013.

[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In Proc. 31st VLDB Conference, pages 853–864. VLDB Endowment, 2005.

[BVIG91] C. Bennis, J.-M. Vézien, G. Iglésias, and A. Gagalowicz. Piecewise surface flattening for non-distorted texture mapping. In Thomas W. Sederberg, editor, Proc. SIGGRAPH ’91, volume 25, pages 237–246, 1991.

[BW09] G. A. Brightwell and P. Winkler. Submodular percolation. SIAM J. Discret. Math., 23(3):1149–1178, 2009.

[CCE+10] E. W. Chambers, E. Colin de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time. Comput. Geom. Theory Appl., 43(3):295–311, 2010.
[CDH+11] A. F. Cook, A. Driemel, S. Har-Peled, J. Sherette, and C. Wenk. Computing the Fréchet distance between folded polygons. In Proc. 12th Workshop Algorithms Data Struct., pages 267–278, 2011.

[CL09] E. W. Chambers and D. Letscher. On the height of a homotopy. In Proc. 21st Canad. Conf. Comput. Geom., 2009.

[CL10] E. W. Chambers and D. Letscher. Erratum for on the height of a homotopy. http://mathcs.slu.edu/~chambers/papers/hherratum.pdf, 2010.

[CLJL11] E. W. Chambers, D. Letscher, T. Ju, and L. Liu. Isotopic Fréchet distance. In Proc. 23rd Canad. Conf. Comput. Geom., 2011.

[CW10] A. F. Cook and C. Wenk. Geodesic Fréchet distance inside a simple polygon. ACM Trans. Algo., 7:9:1–9:19, 2010.

[CW12] A. F. Cook and C. Wenk. Shortest path problems on a polyhedral surface. Algorithmica, 2012. to appear.

[DHW12] A. Driemel, S. Har-Peled, and C. Wenk. Approximating the Fréchet distance for realistic curves in near linear time. Discrete Comput. Geom., 48:94–127, 2012.

[EGH+02] A. Efrat, L. J. Guibas, S. Har-Peled, J. S.B. Mitchell, and T.M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. Discrete Comput. Geom., 28:535–569, 2002.

[EM94] T. Eiter and H. Mannila. Computing discrete Frechét distance. Tech. Report CD-TR 94/64, Christian Doppler Lab. Expert Sys., TU Vienna, Austria, 1994.

[Flo97] M. S. Floater. Parameterization and smooth approximation of surface triangulations. Comput. Aided Geom. Design, 14(4):231–250, 1997.

[Fre24] M. Fréchet. Sur la distance de deux surfaces. Ann. Soc. Polonaise Math., 3:4–19, 1924.

[God99] M. Godau. On the complexity of measuring the similarity between geometric objects in higher dimensions. PhD thesis, Free University of Berlin, 1999.

[HKRS97] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. J. Comput. Sys. Sci., 55:3–23, August 1997.

[HNSS12] S. Har-Peled, A. Nayyeri, M. Salavatipour, and A. Sidiropoulos. How to walk your dog in the mountains with no magic leash. In Proc. 28th Annu. ACM Sympos. Comput. Geom., 2012. 121–130.

[HR11] S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. In Proc. 27th Annu. ACM Sympos. Comput. Geom., pages 448–457, 2011. http://www.cs.uiuc.edu/~sariel/papers/10/frechet3d/.

[KKS05] M.S. Kim, S.W. Kim, and M. Shin. Optimization of subsequence matching under time warping in time-series databases. In Proc. ACM symp. Applied comput., pages 581–586, 2005.
[KP99] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping to massive dataset. In Proc. of the Third Euro. Conf. Princip. Data Mining and Know. Disc., pages 1–11, 1999.

[LT79] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. SIAM J. Appl. Math., 36:177–189, 1979.

[MDBH06] A. Mascret, T. Devogele, I. Le Berre, and A. Hénaфф. Coastline matching process based on the discrete Fréchet distance. In Proc. 12th Int. Sym. Spatial Data Handling, pages 383–400, 2006.

[MMP87] J. S.B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. SIAM J. Comput., 16:647–668, 1987.

[PB00] D. Piponi and G. Borshukov. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In Proc. SIGGRAPH 2000, pages 471–478, August 2000.

[SdS00] A. Sheffer and E. de Sturler. Surface parameterization for meshing by triangulation flattening. In Proc. 9th International Meshing Roundtable, pages 161–172, 2000.

[SGHS08] J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. IEEE Transactions on Audio, Speech & Language Processing, 16(6):1138–1151, 2008.

[SW13] J. Sherette and C. Wenk. Simple curve embedding. CoRR, abs/1303.0821, 2013.

[WSP06] C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing global curve-matching algorithms. In Proc. 18th Int. Conf. Sci. Statis. Database Manag., pages 879–888, 2006.