Measuring Similarity between Curves on 2-manifolds via Minimum Deformation Area

Yusu Wang *

Abstract

Measuring curve similarity is a fundamental problem arising in many application fields. Recently, increasing interest has been focused on measuring similarity between curves embedded in more general metric spaces than the traditional Euclidean space, such as curves in the plane with obstacles. However, so far, only limited results are known, especially for the case when the underlying domain is a general surface. This paper aims at developing a natural curve similarity measure that can be easily extended and computed for curves on general orientable 2-manifolds. Specifically, we measure similarity between homotopic curves based on how hard it is to deform one curve into the other one continuously, and define this "hardness" based on the minimal total surface area swept by any such deformation. We consider cases where curves are embedded in the plane, or on the triangulation of an orientable surface with genus g, and present efficient algorithms for both cases. Let n be the total number of vertices from input curves, and I the number of intersections between input curves. The running time of our algorithm depends near-quadratically on I, and is near linear in n if I is a constant.

*Dept. of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210.

1 Introduction

Measuring curve similarity is a fundamental problem arising in many application fields, including graphics, computer vision, and geographic information systems. Traditionally, much research has been done on comparing curves embedded in the Euclidean space. However, many times, it is natural to study curves embedded in a more general space, such as the similarity between two paths on a terrain. In this paper, we study the problem of curve similarity on a 2-manifold. Specifically, given two simple homotopic curves P and Q embedded on an orientable 2-manifold (including the plane), we measure their similarity by the minimum total area swept when deforming one curve to the other, and present a near-quadratic time algorithm to compute this new measure.

Previously, one of the most popular similarity measures for curves is the Fréchet distance. Intuitively, imagine that a man and his dog are walking in a fixed direction on their respective paths (curves) with a leash between them. The Fréchet distance between these two paths is the minimum leash length necessarily for them to move from one end of the paths to the other end. The Fréchet distance takes the order between points along the curves into account, making it often a better similarity measure for curves than alternatives such as the Hausdorff distance [3, 4].

Given two polygonal curves P and Q with n total edges in $\mathbb{R}^d$, the Fréchet distance can be computed in $O(n^2 \log n)$ time [2]. An $\Omega(n \log n)$ lower bound for the decision problem in the algebraic computation tree model is known [8]. Sub-quadratic approximation algorithms for special families of curves are known [4, 5]. However, so far, no algorithm, exact or approximate, with running time $o(n^2)$ is known for general curves. Furthermore, while the Fréchet distance is a natural curve similarity measure, it is sensitive to outliers. Variants of it, such as the summed-Fréchet distance, and the partial Fréchet similarity, have been proposed [9, 10, 15], usually at the cost of further increasing the time complexity.

Recently, the problem of extending and computing the Fréchet distance to more general metric space has received much attention. Geodesic distance between points is usually considered when the underlying domain is not $\mathbb{R}^d$. For example, Maheshwari and Yi [17] computed the geodesic Fréchet distance between two polygonal paths on a convex polytope in roughly $O(n^3K^4 \log(Kn))$ time, where n and K are the complexity of the input paths and of the convex polytope, respectively. Geodesic Fréchet distance between polygonal curves in the plane, but within a simple polygon (as obstacles), have been studied in [6, 12, 16]. Chambers et al. [11] proposed and studied the so-called homotopic Fréchet distance under a more general setting, where curves reside in a planar domain with a set of points or polygons as obstacles. The extra requirement for this homotopic Fréchet distance is that the leash itself (not just the length of it, as was in the case of geodesic Fréchet distance) has to vary in a continuous manner.

New work. So far, only limited results are known for extending curve similarity measures to surfaces (for example, the Fréchet distance was only extended to convex polytope [17] previously). Computing (variants of) the Fréchet distance also seems to typically induce high computational complexity for more general domains. In this paper, we aim at developing a natural similarity measure for curves on general surfaces that can be computed efficiently. Intuitively, we measure distances between homotopic curves based on how hard it is to deform one curve into the other one, and define this "hardness" based on the minimal total surface area swept by any continuous deformation between them[1] (formal definition is in next Section). We consider both cases where curves are embedded in the plane, or on a closed, triangulated orientable surface with genus g. For the former case, our algorithm runs in $O(n + I^2 \log n)$ time, where n is the total complexity of input curves, and I is the number of intersections between them. The running time is near linear in n if there are only constant number of intersection points. For the latter case, assume that the input

[1] One can consider the geodesic Fréchet distance and its variants as measuring this "hardness" based on the geodesic-width between P and Q. See e.g [16].

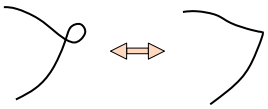
surface is a triangulation of complexity K, then our algorithm runs in time $O(gK \log K + I^2 \log n + In)$ if the genus of the input surface is $g > 0$; otherwise, if $g = 0$, it runs in time $O(K \log K + I^2 \log n)$.

Our similarity measure is natural, and robust against noise. It is efficient both for curves in the planar domain, and on surfaces. We remark that the idea of measuring deformation based on areas is rather common in practice [13, 18]. For example, similarity between two convex polygons can be measured by their symmetric difference [1, 23]. In [7], the area sandwiched between a *x-monotone curve* and another curve is used to measure their similarity. However, "area" between general curves and the theoretical aspects of its computation have not yet been investigated.

The main ideas behind our approach are developed based on several key observations of the structure of the optimal morphing, and its relation with the so-called winding number of a closed curve. Specifically, the use of the winding number enables us to compute the optimal deformation area for a certain family of closed curves (related to boundaries of immersed disk) efficiently in the plane. This forms the basis of our overall dynamic programming framework to compute similarity between curves in the plane. For the case where the underlying surface is a topological sphere, we extend the winding number for such compact surface in a natural way, and show how the algorithm for the planar case can be adapted to the sphere case. For the case when the underlying surface has non-zero genus, we show, that the ideas behind the planar algorithm can still be applied via the concept of universal cover. Finally, we remark that in the remainder of this paper, we focus only on the computation of the similarity measure. However, an accompanying optimal deformation can be constructed easily by our algorithm within the same time and space complexity.

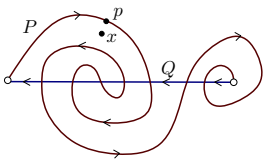
2 Optimal Morphings

2.1 Problem Definition and Notations



Our input are two *simple* curves P and Q sharing the same endpoints on an orientable 2-manifold M without boundary. Given P and Q, a *morphing* between P and Q is a continuous deformation from P to Q. More precisely, there is a continuous family of curves $\Phi_t(s)$, with $t, s \in [0, 1]$, such that $\Phi_0(\cdot) = P$ and $\Phi_1(\cdot) = Q$. We further require that at any time t, the intermediate curve Φ_t is *regular* [24], where a parameterized curve ϕ is regular if $\phi'(s) \neq 0$ for any $s \in [0, 1]$ [2]. Intuitively, this means that the deformation is "kink"-free [19], and cannot create or destroy a local loop as shown in the right figure (the singular point in the right curve is a kink).

The *cost* of a given morphing Φ, denoted by $\mathcal{E}(\Phi)$, is the total area swept by this deformation: note that if a point is swept multiple times by the deformed curve, then it will be counted multiple times when computing the total area. More precisely, given a point $x \in M$, consider the pre-image of x in$[0, 1] \times [0, 1]$ under the map Φ. The *degree* of this point x, denoted by $\rho(x)$, is the number of connected component in its pre-image (roughly speaking, it is how many times the deformed curve sweeps the point x). The cost of Φ is $\mathcal{E}(\Phi) = \int_M \rho(x)dx$. The *similarity* between P and Q is the minimum cost any morphing between P and Q may have, denoted by $\sigma(P, Q) = \min_\Phi \mathcal{E}(\Phi)$. A morphing that produces the minimum cost is an *optimal morphing*. In this paper, we study the problem of computing the similarity between two homotopic input polygonal curves P and Q in the plane or on an orientable compact surface.



Since P and Q share the same endpoints, their concatenation form a (not necessarily simple) closed curve denoted by $C = P \circ Q$. Let $\mathrm{Arr}(C)$ denote the arrangement formed by C, where vertices in $\mathrm{Arr}(C)$ are the intersection points between P and Q (note the vertices of the polygonal curves of P and Q are ignored), and an edge in $\mathrm{Arr}(C)$ is a subcurve of either P or Q. For sake of exposition, the illustrations in this paper usually draw Q as a horizontal segment (see the right figure). We give C (thus P

[2] We sometimes abuse the notations slightly by using the parameterization of a curve to refer to the underlying curve.

and Q) an arbitrary orientation. Hence we can talk about the *sidedness* with respect to C at a point $p \in P$ — a point $x \in \mathbb{R}^2$ is to the *right* of C at p, if it is a counter-clockwise turn from the orientation of the vector px the orientation of (tangent of) C at p (see the previous figure for an example). Given two oriented curves Γ_1 and Γ_2, an intersection point p of them is *positive* if it is a counter-clockwise turn from the orientation of Γ_1 to that of Γ_2 at p. For a curve Γ and a point $p \in \Gamma$, the *index of p* is the parameter of p under the arc-length parameterization of Γ. We sometimes use p to represent its index along Γ when its meaning is clear from the context. Given two points $p, q \in \Gamma$, $\Gamma[p, q]$ denotes the sub-curve of Γ between points p and q.

Finally, we say that a morphing Φ from P to Q is *sense-preserving* if for all times $t \in [0, 1]$, each point $\Phi_t(s)$ deforms to the same side of the curve Φ_t. In other words, for any $t, s \in [0, 1]$, either $\Phi_{t+dt}(s) = \Phi_t(s)$ or $\Phi_{t+dt}(s)$ is to the same side of Φ_t at $\Phi_t(s)$. If it is the former case, then we say that $p = \Phi_t(s)$ is a *fixed point* at time t. The sense-preserving property means that we can continuously deform the curve P in the same direction, without causing local folds in the regions swept. Intuitively, any optimal morphing should have this property to some extent, which we will make more precise and prove later.

2.2 Structure of Optimal Morphings

Given P and Q embedded on a 2-manifold M, let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_I\}$ denote the set of I intersection points between them, sorted by their order along Q. Given a morphing Φ from P to Q, a point $p \in M$ is called an *anchor point w.r.t* Φ if it remains on Φ_t at all times $t \in [0, 1]$. If p is an anchor point, then it is necessarily an intersection point between P and Q, as $p \in \Phi_0 = P$ and $p \in \Phi_1 = Q$. We exclude the beginning and ending end points of P and Q from the list of anchor points, as they remain fixed for all morphings. In what follows, we show that any optimal morphing can be decomposed by anchor points such that each of the resulting sub-morphings has a simple structure.

Specifically, consider an arbitrary optimal morphing Φ^*. Let $\mathbb{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_k\}$ be the set of anchor points w.r.t. Φ^*. We order $\mathbf{B}_i$s by their indices along Q. In fact, the order of their indices along P is the same, and the proof of this simple observation is in Appendix A.

Observation 2.1 *The order of* $\mathbf{B}_i$*s along* P *and along* Q *are the same.*

This observation implies that we can decompose Φ^* into a list of sub-morphings, where Φ_i^* morphs $P[\mathbf{B}_i, \mathbf{B}_{i+1}]$ to $Q[\mathbf{B}_i, \mathbf{B}_{i+1}]$. Obviously, each Φ_i^* is necessarily optimal, and it induces no anchor points.

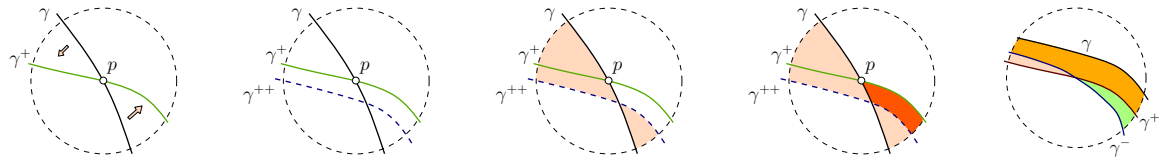


Figure 1: (a) and (b) p is fixed from γ to γ^+, but not so in γ^{++}. Sweeping γ to γ^{++} directly through the shaded region in (c) has a smaller area than first to γ^+ then to γ^{++} (see shaded region in (d)). The darker shaded region in (d) is swept twice. (e) If the deformation changes orientation at γ, then there is a local fold in the regions swept.

Lemma 2.2 *An optimal morphing* ϕ *from* P' *to* Q' *is sense-preserving if it induces no anchor points.*

Proof: Consider a specific time t_0. First we argue that every point on the curve $\gamma = \phi_{t_0}$ is either fixed, or deforms consistently to the same side of γ.

Suppose that this is not the case, and there exist portions of γ deforming to the left and portions deforming to the right of γ. Since ϕ is a continuous map, there must be a (set of) fixed point(s) connecting these left-deforming and right-deforming subcurves. Assume p is this fixed point; see Figure 1 (a). (The case

where there is a piece of fixed points connecting them can be handled similarly.) Since p is not an anchor point, at some point, it will stop being fixed and deform either to the left or right. Assume that this happens at time $t^+ = t_0 + dt$, and that p will deform to the right of $\gamma^+ = \phi_{t^+}$ at time $t^+ + dt$. In other words, p is fixed in curves γ and γ^+, but not in curve $\gamma^{++} = \phi_{t^+ + dt}$. See Figure 1 (b) for an illustration. However, in this case, ϕ cannot be optimal, as we can locally deform γ to γ^{++} directly and sweep a smaller area than first detouring to γ^+. See Figure 1 (c) and (d). Hence the assumption is wrong, and all points in Γ_t deform consistently at a fixed time t.

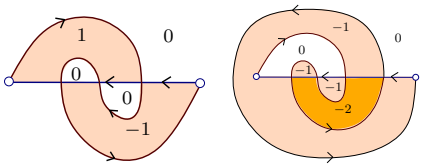
Similarly, we can argue that throughout the deformation, intermediate curves always deform in the same direction. Indeed, if the deformation changes direction at some point, then locally, we must have a "fold" in the region swept as shown in Figure 1 (e), where the curve γ^- will deform to γ then back to γ^+. This is not possible as we can deform γ^- to γ^+ directly without going through C, and sweep a smaller area. The claim then follows. ■

3 Morphing In The Plane

In this section, we consider the case where input curves are from the plane, and develop an algorithm to compute the similarity between P and Q in $O(I^2 \log I + n)$ time, where n is the total complexity of input polygonal curves, and I is the number of their intersections. Note that $I = \Theta(n^2)$ in worst case.

3.1 Winding Number

Previously, we have shown that if an optimal morphing does not induce anchor points, then it is sense-preserving. The implication of this result is manifested via the concept of *winding number*. Specifically, given an oriented closed curve Γ in the plane, let $\mathrm{wn}(x;\Gamma)$ denote the *winding number* of Γ at x (Γ may be omitted when its choice is clear) [3]. Imagine starting from a point y on Γ, and connecting x and y by a string. $\mathrm{wn}(x;\Gamma)$ is an integer measuring how many times this string winds, in a clockwise manner, around x as y traverses Γ. An alternative way to interpret $\mathrm{wn}(x;\Gamma)$ is as follows: Consider any oriented path π from x to the infinity. The winding number $\mathrm{wn}(x;\Gamma)$ is the sum of all signed crossings between Γ and π. Based on this interpretation, it is easy to verify that every point in the same cell of the arrangement $\mathrm{Arr}(\Gamma)$ of Γ has the same winding number, and the winding numbers of two neighboring cells differ by 1.



We say an oriented curve Γ has *consistent winding numbers* if $\mathrm{wn}(x,\Gamma)$ is either all non-negative, or all non-positive, for all $x \in \mathbb{R}^2$. Note that for a curve with consistent winding numbers, we can always orient the curve appropriately so that $\mathrm{wn}(x,\Gamma)$ is all non-negative. Two examples are shown in the figure on the right, where the second example has consistent winding numbers. The relation of consistent winding numbers and sense-preserving morphings is given below, and the proof can be found in Appendix B.

Lemma 3.1 *If there is a sense-preserving morphing Φ from P' to Q', then the closed curve $P' \circ Q'$ has consistent winding numbers.*

Next, we describe two results to connect the above lemma to the computation of optimal morphing. First, we define the *total winding number* $\mathrm{Tw}(\gamma)$ of a curve γ as $\mathrm{Tw}(\gamma) = \int_{\mathbb{R}^2} \mathrm{wn}(x;\gamma)dx$. The following observation is straightforward.

Observation 3.2 *For any P and Q in the plane, $\sigma(P,Q) \geq |\mathrm{Tw}(P \circ Q)|$.*

[3] The term "winding number" is also sometimes used to refer to the *rotation index* [19, 24], which is a number associated with the *entire* input curve, roughly measuring how many turns this curve has. We point it out to avoid confusion.

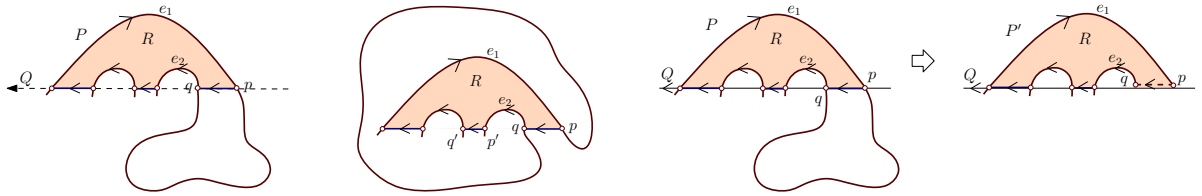


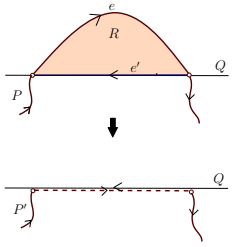
Figure 2: Two cases of relations between $P[p,q]$ and R are shown in (a) and (b). For case (a), we can deform P to P' as shown in (c), and reduce the number of intersections by 2.

Proof: Take any optimal morphing Φ^* from P to Q. Consider the function $F : [0,1] \to \mathbb{R}$ defined as $F(t) = \mathrm{Tw}(\Phi_t^* \circ Q)$. Obviously, $F(0) = \mathrm{Tw}(P \circ Q)$, $F(1) = 0$, and F is a continuous function. Furthermore, each time the winding number at a point x changes by 1 means that some intermediate curve $\Phi(t)$ sweeps through it. In other words, the winding number at x is the lower bound for the degree of x. The claim then follows. ■

Lemma 3.3 *Given P and Q, if $\Gamma = P \circ Q$ has consistent winding numbers, then $\sigma(P,Q) = |\mathrm{Tw}(\Gamma)|$.*

Proof: We prove the claim by induction on the number of intersections between P and Q. The base case is when there is no intersection between P and Q. In this case, Γ is a Jordan curve which decomposes $\mathbb{R}^2$ into two regions, one inside Γ and one unbound. By orienting Γ appropriately, every point in the bounded cell has winding number 1 and the claim follows.

Now assume that the claim holds for cases with at most $k-1$ intersections. We now prove it for the case with k intersections. Let an *X-arc* denote a subcurve of curve X. Consider the arrangement $\mathrm{Arr}(\Gamma)$ formed by $\Gamma = P \circ Q$. Since P and Q are simple, every cell in this arrangement has boundary edges alternating between P-arcs and Q-arcs. Assume without loss of generality that Γ has all non-negative winding numbers. Consider a cell $R \in \mathrm{Arr}(\Gamma)$ with largest winding number. Since its winding number is greater than all its neighbors, it is necessary that all boundary edges are oriented consistently as shown in Figure 2 (a), where the cell R (shaded region) lies to the right of its boundary arcs. (Horizontal edges are Q-arcs in this figure.)



If R has only two boundary arcs, e from P and e' from Q, then we modify P to another simple curve P' by deforming e through R to $-e'$ (where '$-$' means reversing the orientation). See the right figure for an illustration. The area swept by this deformation is exactly the area of cell R. Furthermore, after the deformation, every point $x \in R$ decreases their winding number by 1, and no other point changes its winding number. Since points in this cell initially has strictly positive winding number, the resulting curve $\Gamma' = P' \circ Q$ still has all non-negative winding number. The number of intersections between P' and Q is $k-2$. By induction hypothesis, $\sigma(P',Q) = \mathrm{Tw}(\Gamma')$. Since $\mathrm{Tw}(\Gamma) - \mathrm{Tw}(\Gamma') = \mathrm{Area(R)}$, we have that $\mathrm{Tw}(\Gamma) = \sigma(P',Q) + \mathrm{Area(R)}$. It then follows from Observation 3.2 and the fact $\sigma(P,Q) \le \sigma(P',Q) + \mathrm{Area(R)}$ that $\sigma(P,Q) = \mathrm{Tw}(\Gamma)$.

Otherwise, the cell R has more than one P-arc. Take the P-arc e_1 with smallest index along P (assume w.l.o.g that it is as shown in Figure 2), and let p be the ending endpoint of it. Let e_2 be the next P-arc along the boundary of R, and q its starting endpoint, and $Q[p,q]$ the Q-arc between e_1 and e_2. Obviously, the subcurve $P[p,q]$ cannot intersect R, and $P[p,q]$ and $-Q[p,q]$ bound a simple polygon, which we denote by Ω. Either Ω is on the opposite side of the Q-arc $\bar{e}$ from the interior of R (Figure 2 (a)), or they are on the same side (Figure 2 (b)). If it is the second case (b), then it turns out that the subcurve $P[p',q']$ is necessarily contained inside Ω and has to lie to the opposite side of the Q-arc $Q[p',q']$ as R. Thus we are back to Case (a). Hence we now only discuss how to handle the Case (a).

Note that P does not intersect the interior of Ω; as otherwise, P will either intersect itself or intersect $Q[p,q]$, neither of which is possible. Hence only Q can intersect Ω. Since Q is also a simple curve, there is no vertices of $\mathrm{Arr}(\Gamma)$ contained in the interior of Ω. As a result, every cell of $\mathrm{Arr}(\Gamma)$ contained in Ω must have at least one boundary edge coming from $P[p,q]$. Furthermore, each of such cell has strictly positive winding number; that is, $\mathrm{wn}(x;\Gamma) > 0$ for any $x \in \Omega$. This is because if a cell $\xi \subseteq \Omega$ has winding number 0, then its neighbor across its boundary on $P[p,q]$ will have winding number -1, as Ω is to the right of $P[p,q]$. This violates the condition that Γ has all non-negative winding numbers and thus cannot happen.

We now deform P to P' by sweeping $P[p,q]$ through Ω to $Q[p,q]$. See Figure 2 (d). The cost of this sweeping is $\mathrm{Area}(\Omega)$ and $\mathrm{Tw}(\Gamma) - \mathrm{Tw}(P' \circ Q) = \mathrm{Area}(\Omega)$. P' is still simple, and the number of intersection points between P' and Q is now $k-2$. Since $\mathrm{wn}(x;\Gamma) > 0$ for any $x \in \Omega$, we have $\mathrm{wn}(x; P' \circ Q) \geq 0$ for $x \in \Omega$. No other point will change their winding number after this deformation. Thus the curve $P' \circ Q$ has all non-negative winding numbers as well. Hence by induction hypothesis, we have that $\sigma(P', Q) = \mathrm{Tw}(P' \circ Q)$. Since $\sigma(P,Q) - \sigma(P',Q) \leq \mathrm{Area}(\Omega)$ and $\mathrm{Tw}(\Gamma) - \mathrm{Tw}(\Gamma') = \mathrm{Area}(\Omega)$, it then follows from Observation 3.2 that $\sigma(P,Q) = \mathrm{Tw}(\Gamma)$. This proves the lemma. ■

3.2 The Algorithm

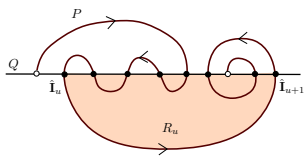
To summarize, Lemma 3.1 implies that if the closed curve $P \circ Q$ produces both positive and negative winding numbers, then any optimal morphing from P to Q must have at least one anchor point. On the other hand, if it has consistent winding numbers, then by Lemma 3.3 we can compute the optimal cost to deform them by simply computing the total winding number. This leads to a simple dynamic-programming (DP) approach to compute $\sigma(P,Q)$.

Specifically, let $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_I$ denote the intersection points between P and Q, ordered by their indices along Q, where $\mathbf{x}_0$ and $\mathbf{x}_I$ are the beginning and ending points of P and Q, respectively. Let $T(i)$ be the cost of the optimal morphing between $P[\mathbf{0}, \mathbf{x}_i]$ and $Q[\mathbf{0}, \mathbf{x}_i]$, and $C[i,j]$ the closed curve formed by $P[\mathbf{x}_i, \mathbf{x}_j] \circ Q[\mathbf{x}_i, \mathbf{x}_j]$. We say that a pair of indices (i,j) is *valid* if (1) $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same order along P and along Q; and (2) the closed curve $C[i,j]$ has consistent winding numbers. We have the following recursion:

$$T(i) = \begin{cases} \mathrm{Tw}(C[0,i]), & \text{if } C[0,i] \text{ has consistent winding numbers} \\ \min_{j<i \text{ and } (j,i) \text{ is valid}} \{\, \mathrm{Tw}(C[j,i]) + T(j) \,\}, & \text{otherwise} \end{cases}$$

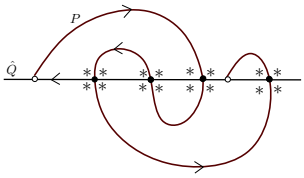
Time complexity. The main components of the above DP framework is to compute $\mathrm{Tw}(C[i,j])$ for all pairs of i,js, and to check whether each pair (i,j) is valid or not. These can be done in $O(I^2 n)$ total time in a straightforward manner. We now show how to compute them in $O(I^2 \log I)$ time after $O(I \log I + n)$ pre-processing time. Specifically, we describe how to compute such information for all $C[r,i]$s for a fixed $r \in [1, I]$ and all indices $i > r$ in $O(I \log I)$ time.

To simplify the description of the algorithm, we extend Q on both sides until infinity, and obtain $\hat{Q}$. We now collect all intersection points between P and $\hat{Q}$, $\{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_L\}$, which is a super-set of previous intersection points, and sort them by their order along the curve P (instead of along Q as in the DP procedure). Such an extension is not necessary for the algorithm, but makes its description cleaner.



Now for a fixed r, we traverse P starting from $\hat{\mathbf{x}}_r$. Each time we pass through an intersection point $\hat{\mathbf{x}}_i$ with $\hat{Q}$, we need to maintain information so that we can (1) check whether (r,i) is valid, and (2) obtain total winding number of $C[r,i]$. Assume we reach the intersection point $\hat{\mathbf{x}}_{u+1}$ from $\hat{\mathbf{x}}_u$. Since these are two consecutive intersection points along P, $P[\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_{u+1}]$ and $\hat{Q}[\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_{u+1}]$ form a simple closed polygon which we denote by R_u (shaded region in the right figure). It is easy to verify

that from $C[r, u]$ to $C[r, u + 1]$, only points within R_u will change their winding number, either all by $+1$ or all by -1, and the winding numbers for points outside R_u are not affected. Hence the change in the total winding number is simply $\alpha_u \mathrm{Area}(\mathrm{R_u})$, where α_u is $+1$ or -1 depending on the sidedness of R_u with respect to $P[\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_{u+1}]$. See the previous figure for an example, where all points in R_u will decrease their winding number by 1 after $\hat{\mathbf{x}}_{u+1}$. We can pre-compute the area of R_u's for all u in $O(I \log I + n)$ time, by observing that the set of R_us satisfy the parentheses property: Namely, either R_u and R_v are disjoint in their interior, or one contains the other. The details can be found in Appendix C.



We now describe how to maintain the winding number for all cells of $\mathrm{Arr}(P + \hat{Q})$ as we pass each $u > r$, so that we can check whether $C[r, u]$ has consistent winding numbers efficiently. To this end, observe that the arrangement of $\mathrm{Arr}(P + \hat{Q})$ is always a refinement of the arrangement of any $C[i, j]$. Thus all points within the same cell of $\mathrm{Arr}(P + \hat{Q})$ always have the same winding number, and we simply need one point from each cell and maintain its winding number. Furthermore, vertices in $\mathrm{Arr}(P + \hat{Q})$ are intersection points between P and $\hat{Q}$. Now take four points around each intersection point $\mathbf{x}_i$ (shown as stars in the right figure). The collection of such *representative points* hit all cells in $\mathrm{Arr}(P + \hat{Q})$. Let U be the set of representatives that are to the right of $\hat{Q}$, which are the stars above $\hat{Q}$ in the right figure. (Those to the left of $\hat{Q}$ will be handled in a symmetric manner). Each point has a key associated with it which is its index along $\hat{Q}$. We build a standard balanced 1-D range tree on U based on such keys, where each leaf f stores a point from U. Every internal node v is associated with an interval $[l_v, r_v]$, where l_v and r_v are the smallest and largest keys stored in the subtree rooted at v. In other words, all representatives with an index along $\hat{Q}$ within $[l_v, r_v]$ are stored in the subtree rooted at v. At every node v, interior or not, we also store a value $addW_v$. To compute the winding number for the representative point p_f stored at a leaf node f, we identify the path $\{v_0, v_1, \ldots, v_a = f\}$ from the root v_0 to f. The winding number for p_f is simply $\sum_{i=0}^{a} addW_{v_i}$. Finally, each internal node v also stores the maximum and minimum winding numbers associated with all leaves in its subtree. At the beginning, all winding numbers are zero. The size of this tree is $O(I)$ with height $O(\log I)$, and can be built in $O(I \log I)$ time once the arrangement $\mathrm{Arr}(P + \hat{Q})$ is known.

Let $\mathbf{q}_i$ denote the index of point $\mathbf{x}_i$ along $\hat{Q}$ (or can be considered as the x-coordinate of $\mathbf{x}_i$). At each intersection $\mathbf{x}_u$, cells of $\mathrm{Arr}(P + \hat{Q})$ contained in R_u should either all increase or all decrease their winding number by 1. In particular, representatives of such cells are simply those contained in the horizontal interval $[\mathbf{q}_u, \mathbf{q}_{u+1}]$. Hence updating the winding number is similar to an interval query of $[\mathbf{q}_u, \mathbf{q}_{u+1}]$, and nodes in the canonical decomposition of $[\mathbf{q}_u, \mathbf{q}_{u+1}]$ update their $addW_v$ values accordingly. The minimum and maximum winding numbers can also be updated $O(1)$ time per visited node. The whole process visits $O(\log I)$ nodes, and thus takes $O(\log I)$ time. To see whether $C[r, u + 1]$ has consistent winding numbers or not, we only need to check the minimum and maximum winding numbers stored at the root of the tree, denoted by w_{min} and w_{max}, respectively. If $w_{min} \times w_{max}$ equals to zero, then all winding numbers w.r.t. $C[r, u + 1]$ are either all non-negative or all non-positive. Otherwise, $(r, u + 1)$ is not valid.

Repeat the above process for every $r \in [1, I]$. Overall, after $O(I \log I + n)$ pre-processing, we can check whether (r, i) is valid or not and compute $\mathrm{Tw}(C[r, i])$ for all $r \in [1, I]$ and all $i > r$ in $O(I^2 \log I)$ time. The dynamic programming procedure then takes $O(I^2 \log I + n)$ time and $O(I^2)$ space to compute $\sigma(P, Q) = T[I]$. We thus conclude:

Theorem 3.4 *Given two polygonal chains P and Q in the plane, of $O(n)$ total complexity, and with I intersection points between them, we can compute the optimal morphing and the similarity between them in $O(I^2 \log I + n)$ time and space.*

4 Morphing on 2-Manifolds

In this section, we consider similarity between two curves P and Q on an orientable and triangulated 2-manifold M without boundary. Our input is a triangulation of M with complexity K, and edges in P and Q are necessarily edges from this triangulation. The total complexity of P and Q is n, with I the number of intersections between them. Note that in this setting, $I = O(n)$. We discuss the cases when M is a topological sphere, and when M has non-zero genus separately.

4.1 Surface with Non-zero Genus

Given an orientable 2-manifold M, let $\mathcal{U}(M)$ be a universal covering space of M, and $\phi : \mathcal{U}(M) \to M$ is the corresponding covering map. Note that ϕ is continuous, surjective, and a local homeomorphism. Given any path γ in M, once the lift (pre-image) of its starting point is fixed, it can be lifted to a unique path $\tilde{\gamma}$ in $\mathcal{U}(M)$, such that $\phi(\tilde{\gamma}) = \gamma$ [14, 21].

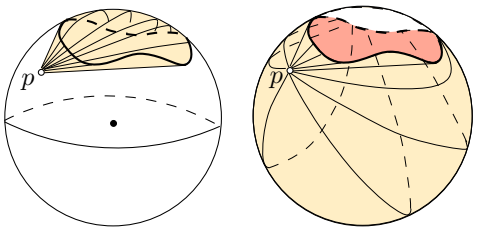
We consider two homotopic simple curves P and Q from M that share endpoints. Thus the closed curve formed by $C = P \circ Q$ is contractible on M, and the lift of C, denoted by $\tilde{C}$, is a closed curve in $\mathcal{U}(M)$. More generally, by the Homotopy Lifting Property of the universal covers [20], we have the following observation.

Observation 4.1 *Once we fix the lift of the starting point of P and Q in $\mathcal{U}(M)$, there is a one-to-one correspondence between morphings between P and Q in M and those between $\tilde{P}$ and $\tilde{Q}$ in $\mathcal{U}(M)$.*

We now impose a area measure in $\mathcal{U}(M)$ by lifting the area measure in M (this can be done as the map ϕ is a local homeomorphism). Hence the cost of a morphing in M is the same as its lifting in $\mathcal{U}(M)$. As such, we can convert the problem of finding an optimal morphing from M to $\mathcal{U}(M)$. Furthermore, for any orientable compact 2-manifold with genus $g > 0$, its universal cover is topologically equivalent to $\mathbb{R}^2$. Thus algorithms and results from previous section hold in the universal covering space. Specifically, given two homotopic paths with n edges from a triangulation of a surface with K simplices, the entire algorithms takes $O(gK \log K + I^2 \log I + In)$. Roughly speaking, we construct the portion of the universal covering space that the lift of C will traverse (and enclose in some sense), which consists of $O(n)$ copies of some polygonal schema of M [14]. The main observation is that only the combinatorial structure of $\tilde{C}$ is needed. Hence we do not fill in each copy of polygonal schema with all triangles. Details are in Appendix D.

4.2 The Case of Sphere

We now consider the remaining case, where the input M is a topological sphere (genus is zero). All paths on M are homotopic. The universal cover of a sphere is itself, hence compact. However, the previous algorithm in Section 3.2 works for a domain homeomorphic to $\mathbb{R}^2$ and cannot be directly applied.



For simplicity, assume the input M is the unit sphere $\mathbf{S}$. First, we observe that the results in Section 2.2 still hold. However, as the sphere is compact, the winding number is not well-defined any more. For example, see the right figure, where there are two ways that we can consider how the curve γ winds around the point p. In the first case, the winding number is 0, while in the second case, the winding number is -1. In order to use a similar DP framework as before to compute the optimal morphing between P and Q, we need to develop analogs of Lemma 3.1 and 3.3.

To this end, observe that if we remove one point, say $\mathbf{z} \in \mathbf{S}$ from the sphere $\mathbf{S}$, then the resulting space $\mathbf{S_z} = \mathbf{S} - \mathbf{z}$ is homeomorphic to $\mathbb{R}^2$, and the concept of the winding number is well defined for $\mathbf{S_z}$. Specifically, $\mathbf{z}$ can be considered as the point of infinity in $\mathbb{R}^2$. The *winding number of $x \in \mathbf{S}$ w.r.t.*

C *and* z, denoted by $\mathrm{wn}(x; \mathbf{z}, C)$ (C omitted when its choice is clear), is simply the summation of signed crossing number for any path connecting x to $\mathbf{z}$. As in the planar case, we say that C is *consistent w.r.t.* $\mathbf{z}$ if $\mathrm{wn}(x; \mathbf{z}, C)$ is either non-negative, or non-positive for all $x \in \mathbf{S}_\mathbf{z}$. Let $\sigma(P, Q; \Omega)$ denote the best cost to morph P to Q within domain Ω. The proofs for the following observations are straightforward and can be found in Appendix E.

Observation 4.2 *Given a closed curve* Γ *and any two points* $\mathbf{z}, \mathbf{w} \in \mathbf{S}$*, we have that:* $\mathrm{wn}(x; \mathbf{w}) = \mathrm{wn}(x; \mathbf{z}) + \mathrm{wn}(\mathbf{z}; \mathbf{w})$ *(all winding numbers are w.r.t the curve* Γ*). In particular, for any two points* $\mathbf{z}_1, \mathbf{z}_2$ *from the same cell of* $\mathrm{Arr}(\Gamma)$*, we have that* $\mathrm{wn}(x; \mathbf{z}_1) = \mathrm{wn}(x; \mathbf{z}_2)$ *for all* $x \neq \mathbf{z}_1, \mathbf{z}_2$*.*

Observation 4.3 *If there is an optimal morphing between* P *and* Q *that does not sweep through some point* $\mathbf{z}$*, then* $\sigma(P, Q; \mathbf{S}) = \sigma(P, Q; \mathbf{S}_\mathbf{z})$*.*

Observation 4.4 *Suppose* Φ^* *is an optimal morphing between* P *and* Q *with no anchor points. For any cell* R *in* $\mathrm{Arr}(P + Q)$*, if* Φ^* *sweeps through one point in its interior, then it sweeps through all points in* R*.*

Lemma 4.5 *If there is an optimal morphing* Φ^* *of* P *and* Q *with no anchor point, then this optimal deformation cannot sweep every point in* $\mathbf{S}$*.*

Proof: We show the lemma by induction on the number of intersections between P and Q. When there is no intersection between P and Q (other than the common endpoints), $P \circ Q$ cuts the sphere into two connected components, and the optimal morphing is the smaller area of the two. The lemma holds for this base case.

Now assume that the lemma holds for P and Q with at most k intersection points. We wish to show the result for the case where P and Q have $k+1$ intersection points. Since Φ^* has no anchor points, this optimal morphing is sense-preserving by Lemma 2.2. Assign an orientation to the closed curve $C = P \circ Q$ so that locally, every point on the curve P will continuously deform to its right during the optimal morphing. Now pick an *arbitrary* point $\mathbf{z}$ not on P and Q, and compute the winding number for each cell of $\mathrm{Arr}(P + Q)$ w.r.t. $\mathbf{z}$. Take the cell R with the largest winding number. Similar to the proof of Lemma 3.3, the boundary of this cell consists of alternating arcs from P and from Q, and they necessarily have the orientation as shown in Figure 2 (a) (otherwise, one of the neighboring cell if R will have a larger winding number). Similar to the proof of Lemma 3.3, choose the P-arc that appears earliest along P, with p being its ending endpoint. Let $P[q]$ be the next intersection between P and R. We have, as shown in Figure 2 (a) and (b), that $P[p, q]$ and $Q[p, q]$ do not intersect each other. The Jordan curve $P[p, q] \circ Q[p, q]$ bounds two regions on the sphere (instead of a bounded one and an unbound one in the case of plane). We consider the one that lies to the right of $P[p, q]$ (thus left of $Q[p, q]$), and denote it by Ω. Since R is to the left of $Q[p, q]$, $\Omega \cap R = \emptyset$.

Let P' be a new curve obtained by replacing $P[p, q]$ with (slightly above) $Q[p, q]$ (same as in Figure 2 (c)). Since the optimal morphing deforms P to its right continuously, and that Ω is simple, there is an optimal morphing between P and Q that consists of first sweeping $P[p, q]$ to $Q[p, q]$ through Ω, and then optimally morph P' to Q. On the other hand, by induction hypothesis, there is an optimal morphing Φ' from P' to Q that does not sweep some point, say $\mathbf{z}_1$ in $\mathbf{S}$. There are now two cases:

(i) If $\mathbf{z}_1 \in \mathbf{S} - \Omega$, then there is an optimal morphing from P to Q that does not sweep $\mathbf{z}_1$ as well. The induction step then holds and the claim follows.

(ii) Otherwise, $\mathbf{z}_1 \in \Omega$. Consider the cell $R' \in \mathrm{Arr}(P' + Q)$ that contains $\mathbf{z}_1$. Note that $R' \cap (\mathbf{S} - \Omega) \neq \emptyset$, as there is no vertices of $\mathrm{Arr}(P' + Q)$ contained neither on nor inside Ω. Hence R' must also contain some point, say $\mathbf{z}_2$, that is outside of Ω. It then follows from Observation 4.4 that $\mathbf{z}_2$ is not swept either. This leads us back to case (i), and the induction step again holds.

The claim then follows by induction. ■

The results above have the following implications. Given any two homotopic paths P' and Q' from $\mathbf{S}$, Lemma 4.5 and Observation 4.3 mean that if P' can be morphed to Q' optimally without anchor points, then there exists some point $\mathbf{z} \in \mathbf{S}$ such that $\sigma(P', Q'; \mathbf{S}) = \sigma(P', Q'; \mathbf{S}_\mathbf{z})$. Once this $\mathbf{z}$ is given, $\sigma(P', Q'; \mathbf{S}_\mathbf{z})$ is simply the total winding number of $P' \circ Q'$ w.r.t. $\mathbf{z}$, as suggested by Lemma 3.3, because $\mathbf{S}_\mathbf{z}$ is homeomorphic to the plane. Furthermore, by Observation 4.4, we only need to pick one point from each cell of $\mathrm{Arr}(P+Q)$ as potentially $\mathbf{z}$. Specifically, let $\{\mathbf{z}_1, \ldots, \mathbf{z}_l\}$ be a set of such *representatives*, with $l = O(I)$. $\sigma(P', Q')$ is simply the smallest of all $\mathrm{Tw}(P' \circ Q'; \mathbf{z}_i)$ for those $\mathbf{z}_i$s with respect to whom the curve $P' \circ Q'$ has consistent winding numbers. In summary, if we assume that if there is an *optimal* morphing between P' and Q' with *no* anchor points, then the results in this section provide an algorithm to compute $\sigma(P', Q')$.

The algorithm for sphere. To compute the optimal morphing between P and Q, we follow the same dynamic programming framework as before. The main difference lies in the component of computing $\sigma(i, j) := \sigma(P[\mathbf{x}_i, \mathbf{x}_j], Q[\mathbf{x}_i, \mathbf{x}_j])$, assuming that $P' = P[\mathbf{x}_i, \mathbf{x}_j]$ *can* indeed optimally deform to $Q' = Q[\mathbf{x}_i, \mathbf{x}_j]$ with no anchor points. Previously, this is done by checking whether $P' \circ Q'$ has consistent winding numbers. Now, we need to check it against $l = O(I)$ number of potential representatives $\{\mathbf{z}_1, \ldots, \mathbf{z}_l\}$. A straightforward implementation of this checking, just for *one* pair (i, j), takes $O(I^2 n)$ time, by computing each $\sigma(P', Q'; \mathbf{S}_{\mathbf{z}_k})$, for $k \in [1, l]$, in $O(n)$ time independently. This can be improved to $O(I \log I)$ time to compute *all* $\sigma(r, j)$s for a fixed r and all $j > r$, after $O(K \log K)$ pre-processing time. See Appendix F for details. Overall, the total time complexity is the same as before, which is $O(I^2 \log I + n + K \log K)$, where K is total complexity of the underlying simplicial complex $\mathbf{S}$.

Theorem 4.6 *Given a piecewise linear 2-manifold M with complexity K and genus g, and given two homotopic paths P and Q of total complexity n, we can compute their similarity $\sigma(P, Q; M)$ in $O(I^2 \log I + K \log K)$ space and time if $g = 0$, and in $O(gK \log K + I^2 \log I + In)$ time for $g > 0$.*

5 Conclusion

In this paper, we proposed a new curve similarity measure, which can be easily extended and computed for homotopic curves on compact orientable 2-manifolds. The measure has a natural interpretation of capturing how hard it is to deform from one curve to the other, based on the amount of total area swept. It is robust to noise (as it is area-based), and can be computed reasonably efficiently, especially for the case of 2-manifolds. An immediate next question is whether we can improve the time complexity of our algorithm for the case when curves are from the plane. Currently, our algorithm is quadratic in I, which can be $\Theta(n^2)$ in the worst case. It will be interesting to see whether we can improve this dependency to linear.

Currently, we assume that two input paths share starting and ending points, which makes it easier to define homotopy equivalence. Another natural question is how to handle them if they do not share endpoints. Can we find the best way to connect the endpoints in all possible homotopy classes?

How to measure similarity for curves on surfaces is an interesting problem. Current methods can usually be viewed as computing certain minimum deformation cost. Geodesic Fréchet-based measures ignore the topological constraints of underlying surface, while the homotopy Fréchet distance and our methods require to identify a homotopy-equivalent deformation. One interesting question is how to develop an area-based curve similarity measure that allows topological changes. For example, we may allow a region to be swept as long as it has trivial homology. This is one of the future directions. Other directions include developing efficient curve simplification algorithms based on this measure, and studying similarity between curves from more general simplicial complexes than considered in this paper (such as manifold with boundary / holes, or non-manifolds).

Acknowledgment. The author would like to thank Joseph O'Rourke and Rephael Wenger for useful discussions at the early stage of this work, and Michael Davis and Tadeusz Januszkiewicz for helpful discussions at the later stage of this paper. We thank anonymous reviewers for helpful comments.

References

[1] H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. *Algorithmica*, 21:89–103, 1998.

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

[3] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier Science Publishers B. V. North-Holland, Amsterdam, 2000.

[4] H. Alt, C. Knauer, and C. Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2004.

[5] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, Revisited. In *Proc. 14th Annu. European Sympos. Algorithms*, pages 52–63, 2006.

[6] S. Bespamyatnikh. An optimal morphing between polylines. *Internat. J. Comput. Geom. Appl.*, 12(3):217–228, 2002.

[7] P. Bose, S. Cabello, O. Cheong, J. Gudmundsson, M. var Kreveld, and B. Speckmann. Area-preserving approximations of polygonal paths. *Journal of Discrete Algorithms*, 4:554–566, 2006.

[8] K. Buchin, M. Buchin, C. Knauer, G. Rote, and C. Wenk. How difficult is it to walk the dog? In *Proc. 23rd Europ. Workshop Comput. Geom.*, pages 170–173, 2007.

[9] K. Buchin, M. Buchin, and Y. Wang. Partial curve matching under the fréchet distance. In *Proc. 20th ACM-SIAM Sympos. Discrete Algorithms*, 2009, To appear.

[10] M. Buchin. *On the Computability of the Fréchet Distance Between Triangulated Surfaces*. PhD thesis, Dept. of Comput. Sci., Freie Universität Berlin, 2007.

[11] E. W. Chambers, E. C. de Verdière, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Walking your dog in the woods in polynomial time. In *Proc. 24th ACM Sympos. Comput. Geom.*, pages 101–109, 2008.

[12] A. F. Cook and C. Wenk. Geodesic Fréchet distance inside a simple polygon. In *Proc. 25th Internat. Sympos. Theoret. Asp. Comp. Sci.*, pages 193–204, 2008.

[13] R. G. Cromley. *Digital Cartography*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[14] T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete and Computational Geometry*, 14(1):93–110, 1995.

[15] A. Efrat, Q. Fan, and S. Venkatasubramanian. Curve matching, time warping, and light fields, new algorithms for computing similarity between curves. *J. Math. Imaging Vis.*, 27(3):203–216, 2007.

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

[17] A. Maheshwari and J. Yi. On computing Fréchet distance of two paths on a convex polyhedron. In *Proc. 21th European Workshop on Computational Geometry*, pages 41–44, 2005.

[18] R. B. McMaster and K. S. Shea. *Generalization in Digital Cartography*. Association of American Cartographers, Washington DC, 1992.

[19] K. Mehlhorn and C.-K. Yap. Constructive Whitney-Graustein Theorem: Or how to untangle closed planar curves. *SIAM J. Comput.*, 20(4):603–621, 1991.

[20] J. J. Rotman. *An Introduction to Algebraic Topology*. Graduate Texts in Mathematics; 119. Springer-Verlag New York Inc., 1988.

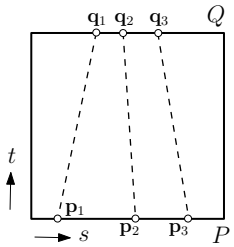
[21] H. Schipper. Determining contractibility of curves. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 358–367, 1992.

[22] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 102–111, 1990.

[23] R. C. Veltkamp. Shape matching: similarity measures and algorithms. In *Proc. Shape Modeling International*, pages 188–199, 2001.

[24] H. Whitney. On regular closed curves in the plane. *Compositio Mathematica*, 4:276–284, 1937.

A Proof for Observation 2.1

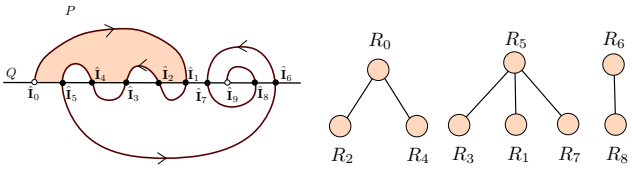


Note that Φ^* is a map from $\square \to M$, where $\square = [0,1] \times [0,1]$ is the unit square and a point $(s,t) \in \square$ will be mapped to $\Phi^*_t(s)$. See the right figure for an illustration. (Since P and Q share starting and ending endpoint, the left and right sides of $\square$ should be contracted to a point. We use the square view for simpler illustration.) The top and bottom boundary edges of this square are mapped to Q and P, respectively. Given an anchor point $\mathbf{B}_i$, let $\mathbf{p}_i$ and $\mathbf{q}_i$ be the parameters of $\mathbf{B}_i$ in Φ^*_0 and Φ^*_1, respectively; that is, $\Phi^*_0(\mathbf{p}_i) = \Phi^*_1(\mathbf{q}_i) = \mathbf{B}_i$. By definition of anchor points, the pre-image of $\mathbf{B}_i$ under the map Φ^* necessarily includes a curve in $\square$ connecting $\mathbf{p}_i$ on the bottom edge to $\mathbf{q}_i$ on the top boundary edge of $\square$. Since $\mathbf{B}_i \neq \mathbf{B}_j$, the pre-images of $\mathbf{B}_i$ cannot intersect with that of $\mathbf{B}_j$. Hence no two such curves can intersect each other, which means that $\mathbf{p}_i$s must be ordered in the same way as $\mathbf{q}_i$s.

B Proof for Lemma 3.1

Without loss of generality, assume that the sense-preserving map Φ always deforms an intermediate curve to its right. Consider the time-varying function $F : [0,1] \times \mathbb{R}^2 \to \mathbb{Z}$, where $F(t,x) = \mathrm{wn}(x; \Phi_t)$ is the winding number at $x \in \mathbb{R}^2$ with respect to the curve parameterized by Φ_t. Obviously, $F(0,x) = \mathrm{wn}(x; P' \circ Q')$, and $F(1,x) = 0$. During the deformation, $F(t,x)$ changes by either 1 or -1 whenever the intermediate curve sweep over it. Since the morphing is sense-preserving, intermediate curves always sweep through x from its left side to the right side. Hence the winding number x decreases monotonically. Since in the end, the winding number at each point is zero, $\mathrm{wn}(x; P' \circ Q') = F(0,x) \geq 0$.

C Computing the Area of R_u for All $u \in [1, I]$



We can pre-compute the area of R_u's for all u in $O(In)$ time in a straightforward manner. This can be improved to $O(I \log I + n)$ time as follows. First, we compute the area of all cells in $\mathrm{Arr}(P+Q)$. Since there are $O(I)$ number of cells in $\mathrm{Arr}(P+Q)$ and the total complexity of all faces is $O(n)$, the arrangement and the area can be computed in $O(I \log I + n)$ time. Next recall that each R_u is the region bounded between a P-arc and a $\hat{Q}$-segment. Consider only P-arcs of the form $P[\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i+1}]$. Since no two P-arcs intersect, the containment relationship between such P-arcs satisfies parentheses property. In particular, we can use a collection of trees to represent the containment relation among all regions R_us. See the right figure for an illustration. The difference between the region represented at a parent node and the union of regions represented by all its children is a cell in $\mathrm{Arr}(P+Q)$. For example, the shaded cell in the right figure is the difference between R_0 and its children R_2 and R_4. We can thus compute the area of all R_us by a bottom-up traversal of this tree. The entire process takes $O(I \log I)$ time once the arrangement is known. With this pre-processing, updating the total winding number at each intersection point takes only $O(1)$ time.

D Algorithm for the Cases with Non-zero Genus

For a triangulation of M with genus $g > 0$ and complexity K, we can construct a universal covering space $\mathcal{U}(M)$ by tiling the so-called polygonal schema of M [22, 14]. Specifically, we are given two homotopic paths P and Q of M, where every edge of them is an edge of M. Let n be the total number of edges in

P and Q, and I the number of intersection points between them. First, we use the algorithm from [14] to construct a polygonal schema T in $O(K)$ time, together with a set of generators G for the first fundamental group of M serving as the boundary of the polygonal schema.

Next, we will construct a portion of the universal covering space that the lift of the curve $C = P \circ Q$ will traverse (in fact, the portion that each cell in $\mathrm{Arr}(\tilde{C})$ will enclose). To this end, we need to separate the cases when $g = 1$ and $g > 1$, because for the case $g = 1$, $\tilde{C}$ can potentially enclose $\Theta(n^2)$ number of copies of polygonal schema inside a bounded cell in its arrangement. For $g > 1$, we again use the algorithm by Dey and Schipper [14] to compute a portion of the covering space that $\tilde{C}$ lies in; this portion U is a topological disk and each copy of polygonal schema traversed is represented by a reduced $4g$-gon. The algorithm takes $O(n \log g + K)$ time. Next, we need to compute the combinatorial structure of the arrangement of $\tilde{C}$ in U. This can be done in $O(gn + I \log K)$ time without explicitly filling all triangles in each copies of the polygonal schema.

For the case $g = 1$, the input manifold is a torus, and the polygon schema is a rectangle. Imagine that we give the base polygonal schema R_0 a *coordinate* $(0, 0)$, and the coordinate for every other copy of the polygon is shown in Figure D. Specifically, a copy of polygonal schema R has coordinate (i, j) if closed cycles whose lift starting from R_0 and ending in R has the same homotopy type as $a^i b^j$. We can easily obtain the sequence of the rectangles (i.e, the coordinates of them) that the curve $\tilde{C}$ will pass through by tracing which generator it intersects as we traverse $\tilde{C}$. This can be done in $O(n)$ time. The combinatorial structure of $\mathrm{Arr}(\tilde{C})$ can then be computed in $O(gn + I \log K)$ time as well.

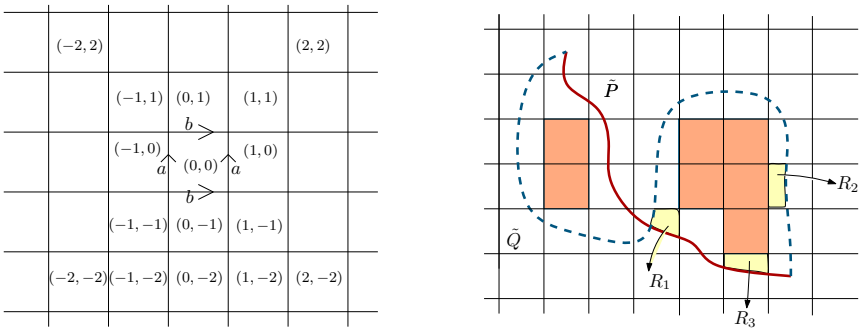


Figure 3: (a) A combinatorial view of the universal covering space $\mathcal{U}(M)$. a and b are the generators and we can give each cell a coordinate. (b) The lift of P (solid curve) and the lift of Q (dashed curve). The heavily shaded region are copies of polygonal schema contained inside cells of $\mathrm{Arr}(\tilde{C})$, and their total number can be easily computed by a scanning algorithm. R_1 is an essential cell; R_2 and R_3 are two non-essential cells.

In order to perform our algorithm introduced in Section 3, we also need the area of each cell in $\mathrm{Arr}(\tilde{C})$. We first describe how to compute it for the case $g = 1$. Specifically, consider any cell X in $\mathrm{Arr}(\tilde{C})$, and assume the boundary of X intersects m copies of polygonal schema. Even though that X may contain $\Theta(m^2)$ copies of polygonal schema in its interior, it turns out that their total area can be computed more efficiently after $O(gK \log K)$ preprocessing.

Indeed, first, by a scanning algorithm from left to right, we can compute in $O(m)$ time how many copies of canonical polygons are completely contained inside X (heavily–shaded region in Figure D (b)) (note that the coordinates of each rectangle traversed by the boundary of X are known). Since the area of every polygonal schema is simply the total area of the input triangulation, we can compute the total area of rectangles contained inside X in $O(m)$ time. Now let $\mathbf{R}$ be the collection of rectangles that intersect the boundary of X. It remains to compute the total area of $\mathbf{R} \cap X$. Since X is a cell in $\mathrm{Arr}(\tilde{C})$, its boundary

consists of alternating arcs from $\tilde{P}$ and from $\tilde{Q}$. Now consider the arrangement formed by the boundary of X and the boundary edges of those copies of rectangles in $\mathbf{R}$. There are two types of cells: the *essential* ones which contain at least one intersection point between $\tilde{P}$ and $\tilde{Q}$ as their vertices, and the *non-essential* ones which are bounded either by arcs from the generators G and arcs from P, or bounded by arcs from the generators G and arcs from Q. See Figure D (b) for examples.

Consider all non-essential cells formed by boundaries of generators and arcs from P — they satisfy the parentheses property as P is a simple curve. Hence similar to the data structure used in Section C to compute the area of R_us, we can compute the area of all such non-essential cells for all X's from $\mathrm{Arr}(\tilde{C})$ in $O(gK \log K)$ time. On the other hand, for every essential cell τ, note that it is the union of a set of cells from $\mathrm{Arr}(P + Q + G)$ on M. Hence to compute the area such a cell τ, we simply traverse cells in $\mathrm{Arr}(P + Q + G)$ to see which ones are contained inside τ, and sum up their areas. This process takes $O(In)$ total time for all essential cells. Putting everything together, the total time to build the combinatorial structure of $\mathrm{Arr}(\tilde{C})$, and to compute the area of each cell, is $O(gK \log K + In)$ for $g = 1$.

The case when $g > 1$ is similar and simpler. The only difference is that for each cell $X \in \mathrm{Arr}(\tilde{C})$, first, we scan through all copies of polygonal schema output by Dey and Schipper's algorithm [14] to find those contained inside and compute their total area. We can afford to do so now as it is shown in [14, 21] that the total number is bounded by $O(m)$ where m is the number of edges of the boundary of X. For the areas of the intersections between X and those canonical polygons traversed by X's boundary, we compute them using the same algorithm as above. The total time complexity is $O(gK \log K + In)$.

With such information computed, we now apply the algorithm from Section 3.2 to compute the best morphing in $O(I^2+K+n)$ time in $\mathcal{U}(M)$, which, by 4.1, gives the optimal morphing between P and Q in M in the same time bound. The total time complexity for the entire algorithm is $O(gK \log K + I^2 \log I + In)$.

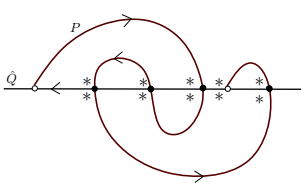
E Proofs for Observations in the Sphere Case

Proof for Observation 4.2. Let $\gamma(x, y)$ be a path connecting point x to y. Note that the concatenation between $\gamma(x, \mathbf{z})$ and $\gamma(\mathbf{z}, \mathbf{w})$ is a path from x to $\mathbf{w}$. Since $\mathrm{wn}(x; \mathbf{w})$ is simply the summed signed crossing number of any path from x to $\mathbf{w}$ with respect to Γ, the claim follows immediately.

Proof for Observation 4.4. Suppose x and y are two points from the interior of R such that Φ^* sweeps through x, but not y. Connect x with y by any path γ in the interior of R. This path has to intersect the boundary of the region swept by Φ^*, and let z be one such intersection point on γ. Obviously, there is a local fold in the optimal morphing as it sweeps through z; namely, some intermediate curve will touch z and immediately trace back. Thus the input morphing Φ^* cannot be sense-preserving. Contradiction. Hence Φ^* sweeps y as well.

F Details of Algorithm for Sphere Case

Here we describe how to compute $\sigma(i, j) := \sigma(P[\mathbf{x}_i, \mathbf{x}_j], Q[\mathbf{x}_i, \mathbf{x}_j])$ efficiently. Specifically, we now show how to compute all $\sigma(r, j)$s for all $j > r$ in $O(I)$ time, for any fixed r, after $O(I \log I + K)$ preprocessing.



First, let us choose the representatives as in Section 3.2 by taking two points around each intersection points $\mathbf{x}_i$ between P and Q. Consider only those representatives to the right of Q (which are those above Q in the right figure), and denote them by $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_I\}$. Z is sorted by their indices along Q. Those to the left of it can be handled similarly. The first observation is that for any two consecutive representatives, $\mathrm{wn}(x; \mathbf{z}_i) - \mathrm{wn}(x; \mathbf{z}_{i+1})$ is 1 or -1, depending on the orientation of P at

$\mathbf{x}_i$. Now to compute which $\mathbf{z}_i$ will give consistent winding numbers, we first compute the winding number of each cell in $\mathrm{Arr}(P+Q)$ for $\mathbf{z}_1$. Next, take the cells R_1 and R_2 with minimum and maximum winding numbers, and assume that $\mathbf{z}_{i_1}$ and $\mathbf{z}_{i_2}$ are their representatives. Based on the previous observation, the closed curve $C[r,j] := P[\mathbf{x}_r, \mathbf{x}_j] \circ Q[\mathbf{x}_r, \mathbf{x}_j]$ has all non-negative winding number w.r.t $\mathbf{z}_{i_1}$, and all non-positive winding number w.r.t. $\mathbf{z}_{i_2}$. Hence we simply compute the total winding number $\mathrm{Tw}(C[r,j]; \mathbf{z}_{i_1}])$ and $\mathrm{Tw}(C[r,j]; \mathbf{z}_{i_2}])$, and return the smaller one as $\sigma(r,j)$. We refer to the indices i_1 and i_2 as the *wn-min and wn-max indices*, respectively, and these two total winding numbers *valid total winding numbers*, which is simply the best cost to deform $P[\mathbf{x}_r, \mathbf{x}_j]$ to $Q[\mathbf{x}_r, \mathbf{x}_j]$ without using anchor points. This improves the time complexity of computing each $\sigma(r,j)$ to $O(I)$ time, instead of the previous $O(In)$ time.

To further improve the time complexity, we will start with $C[r, r+1]$, and update the winding number information in each cell as well as the valid total winding number, as we traverse P and pass through each intersection point $\mathbf{x}_i$. To this end, we use the same range tree data structure as in Section 3.2. Specifically, we use this data structure to maintain the winding number information w.r.t $\mathbf{z}_1$. The wn-min and wn-max indices can be easily maintained by storing at each internal node the minimum and maximum winding number within its subtree. The time complexity for updates remains the same as before (i.e, $O(\log I)$ time per update).

The remaining question is to maintain the valid total winding numbers. Consider the valid total winding number corresponding to wn-min index (that for wn-max index can be maintained similarly). Let A denote the area of $\mathbf{S}$. First, observe that for any C, $\mathrm{Tw}(C; \mathbf{z}_{i+1})$ is simply $\mathrm{Tw}(C; \mathbf{z}_i) + \alpha A$, where $\alpha = 1$ or -1, depending whether P and Q forms a positive or negative crossing at $\mathbf{x}_i$ (a positive crossing means that P rotates in a clockwise manner to Q at the intersection point). Now let $V[j]$ denote the this number for $C[r,j]$. Note that as we pass $\mathbf{x}_{u+1}$, wn-min index κ_{u+1} either stays the same, or change from κ_u by 1 or -1. Let V' be the total winding number w.r.t. κ_u after we pass through intersection $\mathbf{x}_{u+1}$. If $\kappa_u = \kappa_{u+1}$, then $V[u+1] = V'$. Otherwise, $V[u+1] = V' + (\kappa_{u+1} - \kappa_u)A$. Hence the update of valid total winding numbers takes only $O(1)$ time per intersection point.

Hence with $O(K \log K + I \log I)$ pre-processing time, we can compute all $\sigma(r,j)$s for all $j > r$ in $O(I)$ time, for any fixed r, in $O(I \log I)$ total time.