Lifted Bijections for Low Distortion Surface Mappings

Noam Aigerman Roi Poranne Yaron Lipman
Weizmann Institute of Science

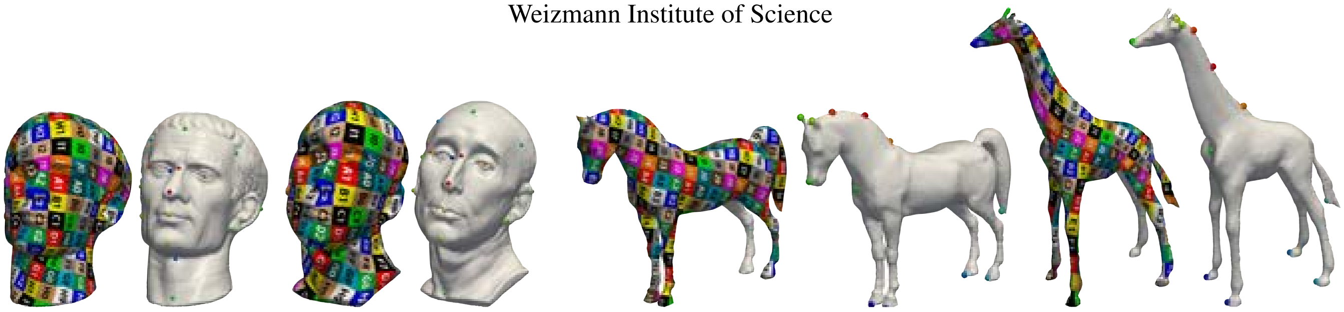


Figure 1: *The algorithm presented in this paper generates low-distortion bijective mappings between surfaces from a sparse set of landmarks (visualized as colored spheres here). The maps are visualized by transferring the texture of the visible part in the left mesh of each pair to the right mesh, using the computed mappings. For example, the right pair shows a mapping of a horse to a giraffe; note how the map stretches gracefully at the neck area.*

Abstract

This paper introduces an algorithm for computing low-distortion, bijective mappings between surface meshes. The algorithm receives as input a coarse set of corresponding pairs of points on the two surfaces, and follows three steps: (i) cutting the two meshes to disks in a consistent manner; (ii) jointly flattening the two disks via a novel formulation for minimizing isometric distortion while guaranteeing local injectivity (the flattenings can overlap, however); and (iii) computing a unique continuous bijection that is consistent with the flattenings.

The construction of the algorithm stems from two novel observations: first, bijections between disk-type surfaces can be uniquely and efficiently represented via consistent *locally* injective flattenings that are allowed to be globally overlapping. This observation reduces the problem of computing bijective surface mappings to the task of computing locally injective flattenings, which is shown to be easier. Second, locally injective flattenings that minimize isometric distortion can be efficiently characterized and optimized in a convex framework.

Experiments that map a wide baseline of pairs of surface meshes using the algorithm are provided. They demonstrate the ability of the algorithm to produce high-quality continuous bijective mappings between pairs of surfaces of varying isometric distortion levels.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

Keywords: surface mesh, isometric distortion, bijective simplicial mappings

Links: DL PDF

1 Introduction

Computing high quality mappings of surfaces is a fundamental task in computer graphics, with a large number of applications, including (but not limited to) texture mapping, morphing, animation transfer, attribute transfer, shape analysis, and shape matching. For many of these applications it is vital to use high-quality maps that are bijective (i.e., without fold-overs) and with low isometric distortion. Achieving these goals is still, to a large extent, an open problem.

Given two topologically equivalent surface meshes, the goal of this paper is to devise an efficient algorithm, requiring minimal user input, for calculating a low-distortion bijective mapping between the surfaces. The task of constructing such mappings between surfaces poses two challenges: (i) how to represent bijections between surfaces, and (ii) how to efficiently minimize the isometric distortion, while preserving the bijectivity of the mapping?

A common approach to tackling these challenges is to first define a common and simple *base domain* that is topologically equivalent to the two surfaces, and to then construct the final map using bijective parameterizations to and from the base domain [Kraevoy and Sheffer 2004; Schreiner et al. 2004]. Constructing these base domains is usually a non-trivial task. Additionally, *a priori* fixing the domain and requiring it to be homeomorphic to the two surfaces makes it hard to optimize over the bijections and/or reduce the isometric distortion. Thus, computation of the map is usually addressed by using an expensive non-linear numerical optimization of the isometric distortion energy.

The first key observation in this paper is that it is *not* necessary to have injective parameterizations of the two surfaces to a common homeomorphic base domain for the construction of a bijective map between them. We introduce a weaker, yet sufficient condition on the parameterizations of the two surfaces, assuring that they uniquely define a bijection between the surfaces. The crucial point is that this condition only requires enforcing *local* injectivity of the parameterizations during the optimization (they are allowed to be globally overlapping). This observation reduces the problem of constructing continuous bijective maps between two surface meshes to the considerably easier problem of optimizing *locally* injective parameterizations of the two surfaces into the plane.

Equipped with this paradigm for representing bijections, the second challenge is addressed by simultaneously optimizing for two locally injective flattenings of the two surfaces with low isomet-

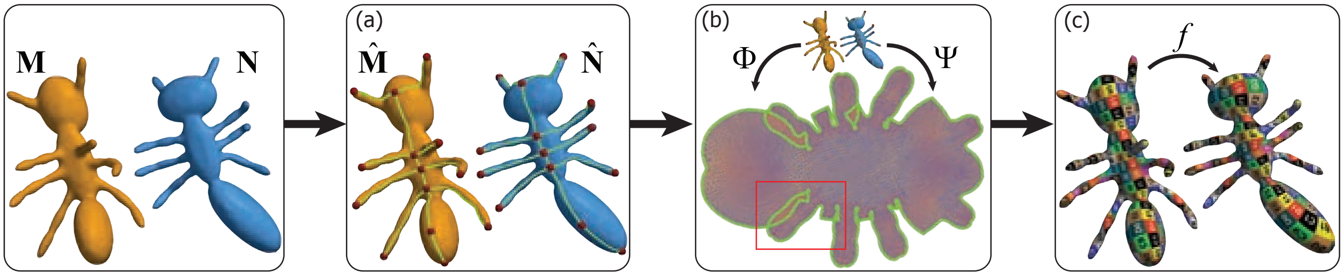


Figure 2: *The three steps of our algorithm for computing a bijective map between two surface meshes: (a) the input meshes* $\mathbf{M}, \mathbf{N}$ *are cut along the cut-graph (in green) spanning input landmarks (red spheres) to generate the cut meshes* $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ *with boundary correspondence* ν*. (b) the locally injective, low-distortion flattenings* Φ, Ψ *of* $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ *(resp.) are computed (note the self-overlapping in the plane, e.g., the area highlighted with a red rectangle). (c) the lifting algorithm recovers the unique bijection* f *between* $\mathbf{M}$ *and* $\mathbf{N}$*, induced by* Φ, Ψ*.*

ric distortion. For minimizing the non-convex isometric distortion energy of the parameterizations while maintaining local injectivity, we suggest a novel convexification of a couple of isometric distortion functionals that is in some sense optimal, and allows more efficient and global optimization than was possible before. Thus, the problem of computing low-distortion bijections of surfaces is reduced to solving a sequence of convex problems, without taking any elaborate geometric considerations into account during the optimization process.

We have tested our algorithm on a wide baseline, spanning nearly isometric pairs of surfaces (e.g., Figure 1, left) to significantly different pairs (e.g., Figure 1, right), and were able to compute global, bijective, and low isometric-distortion maps between the pairs, using $2-20$ points specified by the user. Our method is shown to compare favorably to several state-of-art surface map generation algorithms, in terms of distortion and injectivity.

2 Previous work

Surface mapping via base domains. Mesh parameterization [Sheffer et al. 2006; Hormann et al. 2007] is often used to compute mappings between surfaces. A common approach is to find a parameterization of the two surfaces to a common base domain [Lee et al. 1999; Praun et al. 2001; Lin et al. 2003; Michikawa et al. 2001] and then define the final map by composing one parameterization with the inverse of the other. The base domain is often chosen to be a coarse triangulation, or a simplicial complex. Schreiner et al., [2004] used progressive meshes to define the base domain and optimize the map between the surfaces. Kraevoy and Sheffer [2004] and Bradley et al., [2008] built the base domain by consistently connecting feature points with equivalent paths over the two meshes; in both cases, the common domain is non-trivial to compute. To optimize the map, these techniques use only local (one-ring) updates that are sometimes slow to converge and could easily get stuck in a local minimum. In contrast, our method does not require that a common homeomorphic base domain be given - only a consistent cut of the two surfaces to topological disks is required. Furthermore, our method applies global optimization steps, and the common domain is allowed to self-overlap. Finally, our method is guaranteed to produce a bijection between the surfaces.

Global surface registration. Several methods apply surface deformation priors along with fitting criteria for surface registration [Chui and Rangarajan 2003; Allen et al. 2003; Anguelov et al. 2004; Anguelov et al. 2005; Huang et al. 2008]. In the context of iterative closest point algorithms [Besl and McKay 1992], the correspondences are updated in each iteration, often based on the closest point operator [Pauly et al. 2005; Li et al. 2008; Brown and Rusinkiewicz 2007]. Some authors use specific classes of mappings, such as conformal [Gu et al. 2004; Lipman and Funkhouser 2009] or quasi-conformal [Zeng et al. 2009] mappings, in order to find correspondences between surfaces. Blending of different maps was employed in [Kim et al. 2011].

Generalized surface mappings and correspondences. Several methods concentrate on the problem of finding some generalized notion of mapping between surfaces (see [van Kaick et al. 2011] for a detailed survey). Several papers have used ideas from metric-geometry, such as the Gromov-Hausdorff distance, to compare surfaces and establish correspondences [Bronstein et al. 2006; Mémoli and Sapiro 2005]. Others use spectral embeddings to cancel isometric deformations [Jain et al. 2007; Ovsjanikov et al. 2010], or relaxation to measures instead of maps [Solomon et al. 2012]. Functional maps [Ovsjanikov et al. 2012] relax the non-linear nature of surface mapping by embedding in the functional linear space. Geodesic consistency was used to establish correspondences in [Tevs et al. 2009]. Although able to produce good correspondences between pairs of surfaces, all of these methods do not produce a continuous bijective map between the surfaces. Recently, Panozzo et al., [2013] used generalized weighted averages on surfaces to define mappings between surfaces. However, this is done without guaranteeing bijectivity.

3 Method

3.1 Overview

Let $\mathbf{M} = (\mathbf{V_M}, \mathbf{E_M}, \mathbf{T_M})$, $\mathbf{N} = (\mathbf{V_N}, \mathbf{E_N}, \mathbf{T_N})$ be two topologically equivalent, oriented and boundaryless surface meshes. Let a sparse collection of pairs of point correspondences (landmarks) $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{y}_i)\} \subset \mathbf{M} \times \mathbf{N}$, $i = 1, .., k$ be given as well. Our goal is to construct a continuous bijective surface mapping $f : \mathbf{M} \to \mathbf{N}$ with low isometric distortion that interpolates the correspondences. Our approach consists of three steps (illustrated in Figure 2):

1. **Cutting to disk topology.** A *cut-graph* $\mathcal{G}$ is computed. $\mathcal{G}$ is embedded on the two surfaces and spans the input corresponding pairs $\mathcal{P}$. For example, Figure 2 (a) shows the cut-graph in green, and the input correspondences in red. Then, $\mathbf{M}, \mathbf{N}$ are cut according to $\mathcal{G}$ to achieve the two disk-type surfaces $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ (resp.). The cut-graph represents the boundaries of $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{N}}$ and provides a bijective correspondence between the boundaries, denoted by $\nu : \partial\widehat{\mathbf{M}} \to \partial\widehat{\mathbf{N}}$.

2. **Computing the joint flattenings** Φ, Ψ**.** Two *locally injective*[1] flattenings $\Phi : \widehat{\mathbf{M}} \to \mathbb{R}^2$, $\Psi : \widehat{\mathbf{N}} \to \mathbb{R}^2$ are computed. These

---

[1] Φ is locally injective if it is injective over the one-rings of $\widehat{\mathbf{M}}$.

two flattenings minimize the isometric distortion energy and are consistent on the boundary, i.e., $\Phi(\mathbf{x}) = \Psi(\nu(\mathbf{x}))$ for all $\mathbf{x} \in \partial\widehat{\mathbf{M}}$. Figure 2 (b) shows the result of a joint flattening of two meshes, where the joint, self-overlapping boundary is highlighted in green.

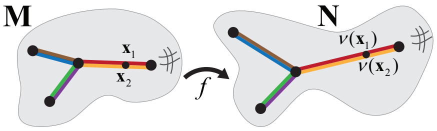
3. **Bijection Lifting.** The flattenings Φ, Ψ are "lifted" using a procedure we call *bijection lifting* to recover the (unique) bijective map $f : \mathbf{M} \to \mathbf{N}$ that satisfies (i) $\Phi = \Psi \circ f$ for all points in $\mathbf{M}$; and (ii) $f = \nu$ for all boundary points $\partial\widehat{\mathbf{M}}$. In particular, f interpolates the input correspondences, i.e. $f(\mathbf{x}_i) = \mathbf{y}_i$. This final map is shown in Figure 2 (c). Although the flattenings in Figure 2(b) have fold-overs, the final map is guaranteed to be a bijection.

In the following we elaborate on each step of this algorithm. We start by describing the bijection lifting, move to the computation of the joint flattenings, and end with the cutting to disk topology.

3.2 Bijection lifting

In this part of the algorithm we have at our disposal the cut meshes $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$, the boundary correspondence $\nu : \partial\widehat{\mathbf{M}} \to \partial\widehat{\mathbf{N}}$ and the mappings $\Phi : \widehat{\mathbf{M}} \to \mathbb{R}^2$, $\Psi : \widehat{\mathbf{N}} \to \mathbb{R}^2$. Our goal in this section is to discover when is it possible, and how, to recover a bijection $f : \mathbf{M} \to \mathbf{N}$ consistent with the flattenings Φ, Ψ.

Reduction to cut meshes. It is enough to compute a bijection $f : \widehat{\mathbf{M}} \to \widehat{\mathbf{N}}$ between the *cut* meshes, satisfying $f(\mathbf{x}) = \nu(\mathbf{x})$, for all $\mathbf{x} \in \partial\widehat{\mathbf{M}}$. Indeed, f is also a continuous bijection between the *uncut* surfaces, i.e. $f : \mathbf{M} \to \mathbf{N}$, since the boundary map ν takes duplicate points (e.g., $\mathbf{x}_1, \mathbf{x}_2$ in the inset) on different sides of the cut of $\mathbf{M}$ (shown here and after as lines colored according to ν) to duplicate points $(\nu(\mathbf{x}_1), \nu(\mathbf{x}_2))$ on the cut of $\mathbf{N}$.



Henceforth, we will visualize the cut meshes $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ as disks (see e.g. Figure 3, boundary colors matching to those of the above inset), and focus on computing a bijection $f : \widehat{\mathbf{M}} \to \widehat{\mathbf{N}}$ as follows,

Definition 1. *Given Φ, Ψ, ν, if there exists a continuous bijection $f : \widehat{\boldsymbol{M}} \to \widehat{\mathbf{N}}$ such that: i) f is* consistent *with Φ, Ψ, that is $\Phi(\boldsymbol{x}) = \Psi(f(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \widehat{\boldsymbol{M}}$; and ii) f interpolates ν at the boundary, that is $f(\boldsymbol{x}) = \nu(\boldsymbol{x})$, for all $\boldsymbol{x} \in \partial\widehat{\boldsymbol{M}}$, then Φ, Ψ are **liftable** and f is their **lifted bijection**.*

Condition (i) can be seen as a "weak" form of defining f via the inversion $f = \Psi^{-1} \circ \Phi$ which is valid also when Φ, Ψ are not injective. In the following, we first establish sufficient conditions on Φ, Ψ that guarantee they are liftable. Then, we derive an efficient algorithm to compute f from a given liftable Φ, Ψ. As the sufficient conditions of Φ, Ψ are met by our joint flattening algorithm (Section 3.3), the correctness of the algorithm is implied.

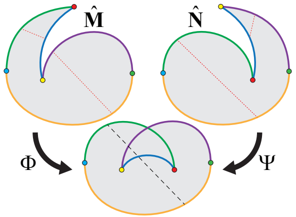
Sufficient conditions for liftable flattenings. We introduce the following conditions:

Definition 2. *Φ, Ψ are said to satisfy the **lifting conditions** if there exists a homotopy (i.e., a continuous deformation) of flattenings Φ_t, Ψ_t, $t \in [0, 1]$ where $\Phi_1 = \Phi, \Psi_1 = \Psi$, such that: a) Φ_0, Ψ_0 are liftable; b) Φ_t, Ψ_t are locally injective for all t; and c) $\Phi_t(\boldsymbol{x}) = \Psi_t(\nu(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \partial\widehat{\boldsymbol{M}}$ and t .*

The lifting conditions guarantee liftability of Φ, Ψ as the next theorem asserts.

Theorem 1. *Let $\widehat{\boldsymbol{M}}, \widehat{\boldsymbol{N}}$ be two disk-type surface meshes and $\nu : \partial\widehat{\boldsymbol{M}} \to \partial\widehat{\mathbf{N}}$ a bijection of their boundaries. Let $\Phi : \widehat{\boldsymbol{M}} \to \mathbb{R}^2$, $\Psi : \widehat{\boldsymbol{N}} \to \mathbb{R}^2$ be flattenings that satisfy the lifting conditions. Then, Φ, Ψ are liftable. In other words, there exists a unique continuous bijection $f : \widehat{\boldsymbol{M}} \to \widehat{\mathbf{N}}$ satisfying $\Phi(\boldsymbol{x}) = \Psi(f(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \widehat{\boldsymbol{M}}$, and $f(\boldsymbol{x}) = \nu(\boldsymbol{x})$, for all $\boldsymbol{x} \in \partial\widehat{\boldsymbol{M}}$.*

We give a sketch of the proof of the Theorem in the appendix. It is important to note that the lifting conditions (a)-(c) are necessary to *ensure* existence of the bijection. For example, the inset shows two non-liftable flattenings that violate the local injectivity at a single point (each at a different point). These flattenings are not liftable since a preimage of the dashed black line (bottom) cannot be mapped homeomorphically from $\widehat{\mathbf{M}}$ to $\widehat{\mathbf{N}}$ while respecting the boundary map ν (visualized with colored curves).



Computation of the bijection. Given two liftable flattenings Φ, Ψ the task at hand is to reconstruct f. The fact that f exists simplifies this task as follows (we refer to Figure 3 to exemplify the bijection lifting process): Let $\mathbf{x} \in \widehat{\mathbf{M}}$ be an arbitrary point (e.g., Figure 3, left) for which we want to compute an image under the (unknown) bijection $f(\mathbf{x})$ (right). According to Theorem 1, f satisfies $\Phi(\mathbf{x}) = \Psi(f(\mathbf{x}))$, but if we simply map $\mathbf{x}$ using Φ to $\mathbf{z} = \Phi(\mathbf{x})$ (middle) it would be impossible to determine which of its preimages $\Psi^{-1}(\mathbf{z})$ is $f(\mathbf{x})$ (it has three different preimages in $\widehat{\mathbf{N}}$ in Figure 3). Nevertheless, this ambiguity can be resolved: by choosing an arbitrary point on the boundary $\mathbf{x}_0 \in \partial\widehat{\mathbf{M}}$ (yellow point, left) and connecting $\mathbf{x}_0$ and $\mathbf{x}$ with a simple (not self-intersecting) polygonal curve $p = [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_n, \mathbf{x}]$ in $\widehat{\mathbf{M}}$ (black line, left). $f(\mathbf{x})$ can be recovered by traversing the image of p, that is $w = \Phi(p) = [\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{z}]$, and computing f as one progresses.

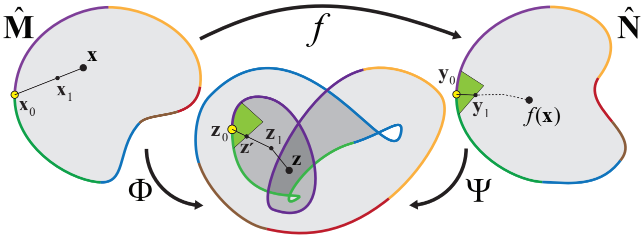
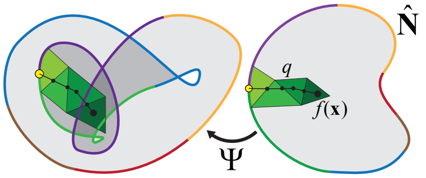


Figure 3: *The bijection lifting scheme.*

By the conditions of Theorem 1, f is known on the boundary $\partial\widehat{\mathbf{M}}$ - it is equal to ν, and so we let $\mathbf{y}_0 = \nu(\mathbf{x}_0)$. To progress along w, we first denote by $\mathcal{R}_{\widehat{\mathbf{N}}}(\mathbf{y})$ the *one-ring* of a point $\mathbf{y}$ in $\widehat{\mathbf{N}}$, that is the union of all faces of $\widehat{\mathbf{N}}$ that contain $\mathbf{y}$. $\mathcal{R}_{\widehat{\mathbf{N}}}(\mathbf{y}_0)$ is shown in green in Figure 3, right. Let $\mathcal{R} = \Psi(\mathcal{R}_{\widehat{\mathbf{N}}}(\mathbf{y}_0))$ be the image of the one-ring of $\mathbf{y}_0$ under Ψ (shown in green, middle). Since Ψ is injective when restricted to any one-ring of $\widehat{\mathbf{N}}$, we can "lift" $w \cap \mathcal{R} = [\mathbf{z}_0, \mathbf{z}']$ to $\widehat{\mathbf{N}}$, resulting in $[\mathbf{y}_0, \mathbf{y}_1]$. This process can be repeated for the polygonal curve $[\mathbf{z}', \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{z}]$ as shown in the inset figure. This produces a lifting $q = [\mathbf{y}_0, ..., \mathbf{y}_m, \mathbf{y}] \subset \widehat{\mathbf{N}}$ where $\mathbf{y} = f(\mathbf{x})$. A pseudo-code for this procedure is given in Algorithm 1.



Algorithm 1: Path Lifting

```
Input:
Path $w = [\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{z}]$
A point $\mathbf{y}_0 \in \widehat{\mathbf{N}}$ known to be corresponding to $\mathbf{z}_0$
Locally injective mapping $\Psi : \widehat{\mathbf{N}} \rightarrow \mathbb{R}^2$
Output:
A lift $q \subset \widehat{\mathbf{N}}$ such that $\Psi(q) = w$, where $q = [\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}]$

if $p$ contains only one vertex $z_0$ then
    return $\mathbf{y}_0$
Let $e = (\mathbf{z}_0, \mathbf{z}_1]$ be the first edge in $w$
Let $\mathcal{R} := \Psi(\mathcal{R}_{\widehat{\mathbf{N}}}(\mathbf{y}_0))$ the image of the one-ring of $\mathbf{y}_0$
if $e \subset \mathcal{R}$ then
    set $\mathbf{y}_1 \in \widehat{\mathbf{N}}$ by solving $\Psi|_{\mathcal{R}_{\widehat{\mathbf{N}}}(\mathbf{y}_0)}(\mathbf{y}_1) = \mathbf{z}_1$
    Log the face $t \in \mathbf{T}_{\widehat{\mathbf{N}}}$ containing $\mathbf{y}_1$
    Log the barycentric coordinates of $\mathbf{y}_1 \in t$
    run Path Lifting on $w = [\mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{z}]$, and $\mathbf{y}_1$
else
    define a new point $\mathbf{z}' = e \cap \partial\mathcal{R}$
    Set $p = [\mathbf{z}_0, \mathbf{z}', \mathbf{z}_1, ..., \mathbf{z}_n, \mathbf{z}]$
    run Path Lifting on $\mathcal{T}$
```

This algorithm can be used to compute the image under the bijection f of *any* point $\mathbf{x} \in \widehat{\mathbf{M}}$: one simply chooses a simple polygonal curve p in $\widehat{\mathbf{M}}$ connecting a boundary point (i.e., a point where f is known) to $\mathbf{x}$ and then applies Algorithm 1 with $w = \Phi(p)$. It is often the case that one is interested in computing f for many points in $\widehat{\mathbf{M}}$, e.g. for finding the image of all the vertices of $\widehat{\mathbf{M}}$ under f, namely, for computing $f(\mathbf{v})$ for all $\mathbf{v} \in \mathbf{V}_{\widehat{\mathbf{M}}}$ (we represent every vertex image $f(\mathbf{v})$ using barycentric coordinates in the target triangle). To avoid constructing a separate curve for each vertex, we first build a spanning tree $\mathcal{T}$ of all the vertices in $\widehat{\mathbf{M}}$ (or all points we want to map), rooted in some boundary point $\mathbf{x}_0 \in \partial\widehat{\mathbf{M}}$. Then, we map this tree to the plane via $\Phi(\mathcal{T})$, and run the same process as that of the Path Lifting algorithm, to lift the tree to $\widehat{\mathbf{N}}$.

3.3 Computing joint flattenings Φ, Ψ

In this section, we address the next logical step: computing the mappings Φ, Ψ that satisfy the lifting conditions of Theorem 1, while minimizing their isometric distortion. This is achieved by iteratively computing a sequence of flattenings $\Phi_0, \Psi_0 \rightarrow \Phi_1, \Psi_1 \rightarrow ... \rightarrow \Phi, \Psi$ with decreasing isometric distortion while ensuring each iteration preserves liftability.

We begin this subsection by exploring the optimization of the isometric distortion of a single affine map, then move to optimization of the flattenings Φ, Ψ, and lastly explain why this optimization procedure results in a liftable pair Φ, Ψ.

Isometric distortion of an affine map. The isometric distortion exerted by a planar (non-degenerate) affine map $\mathcal{A}(\mathbf{x}) = A\mathbf{x} + \boldsymbol{\delta}$, where $A \in \mathbb{R}^{2\times 2}$, $\boldsymbol{\delta} \in \mathbb{R}^{2\times 1}$, can be expressed using the two norms $\|A\|_2$, $\left\|A^{-1}\right\|_2$. Intuitively, $\|A\|_2$ measures the maximal stretch of the linear transformation A, and $\left\|A^{-1}\right\|_2$ measures the maximal stretch of the inverse transformation. Both these scalars can be written using the singular values $0 < \sigma \leq \Sigma$ of A: $\|A\|_2 = \Sigma$, and $\left\|A^{-1}\right\|_2 = 1/\sigma$. There is a rather large variety of possible isometric distortion measures involving these two numbers [Hormann and Greiner 2000; Sander et al. 2001; Floater and Hormann 2005]. We discuss isometric distortion measures that can be written as

$$E(A) = h\left(\|A\|_2, \left\|A^{-1}\right\|_2\right), \tag{1}$$

where $h(r, s) : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ is a convex function, nondecreasing in each of its arguments. This framework is instantiated here for two choices of the function h:

$$h(r, s) = \max\{r, s\} \tag{2}$$

$$h(r, s) = \sqrt{r^2 + s^2}. \tag{3}$$

Note that these functionals naturally prevent the degeneration of A, since they go to infinity as the determinant approaches 0. Using eq. (2) would provide the maximal isometric distortion energy (i.e., $\max\{\Sigma, 1/\sigma\}$), which is also known as the *dilation* in metric geometry. It was used for mesh parameterization in [Sorkine et al. 2002] to measure the isometric distortion, but was not optimized directly. In [Schreiner et al. 2004], a similar energy to (3) (i.e., $\Sigma^2 + 1/\sigma^2$) was used, but their optimization procedure required moving the vertices one by one in a random direction (line search), and was confined to small movements to avoid introducing flips.

We provide a useful convexification of the energy (1), and claim that this convexification is in some sense optimal. This leads to an efficient optimization of the usually difficult-to-optimize isometric distortion energies. First, equation (1) can be reformulated using a variant of the so-called epigraph form (see e.g., Section 3.1.7 in [Boyd and Vandenberghe 2004]):

$$\min \quad h(r, s) \tag{4a}$$

$$\text{s.t.} \quad \|A\|_2 \leq r \tag{4b}$$

$$\left\|A^{-1}\right\|_2 \leq s \tag{4c}$$

Second, the observations presented in [Lipman 2012] are used to formulate the constraint (4b) as two (convex) second-order cone constraints, and convexify eq. (4c) by optimizing over convex subsets of matrices A that satisfy this constraint.

Similarly to [Lipman 2012] we decompose an arbitrary 2×2 matrix A as $A = B + C$, where $B = \frac{1}{2}\left(A - A^T + \mathrm{tr}(A)I\right)$ is a similarity matrix, and $C = \frac{1}{2}\left(A + A^T - \mathrm{tr}(A)I\right)$ is an anti-similarity matrix (i.e., a similarity matrix composed with a reflection). In this notation, the singular values of A can be written as

$$\Sigma = \frac{1}{\sqrt{2}}\left(\|B\|_F + \|C\|_F\right), \ \sigma = \frac{1}{\sqrt{2}}\left|\,\|B\|_F - \|C\|_F\,\right|.$$

Hence, constraint (4b) is equivalent to the convex constraints:

$$s_B + s_C \leq r \tag{5a}$$

$$\|B\|_F \leq \sqrt{2}s_B \tag{5b}$$

$$\|C\|_F \leq \sqrt{2}s_C, \tag{5c}$$

where s_B, s_C are auxiliary variables. Constraints (5b),(5c) are standard second-order cone constraints.

The non-convex constraint in eq. (4c) can be equivalently written as the two equations $1 \leq us$, and $0 \leq u \leq \sigma$ (u auxiliary). For the first equation we use the identity $4us = (u+s)^2 - (u-s)^2$ to obtain

$$\sqrt{(u-s)^2 + 4} \leq u + s, \tag{6}$$

where we are allowed to use the square-root since $u, s \geq 0$.

To convexify the second inequality, $u \leq \sigma$, which is the only non-convex constraint in the optimization problem, we employ the technique of [Lipman 2012] and replace it with a maximal convex subset:

$$u \leq \frac{1}{2}\mathrm{tr}(R^T B) - \frac{1}{\sqrt{2}}\left\|C\right\|_F ,$$

where R is an (arbitrary for the moment) 2×2 rotation. Thus, the following convexification of eq. (4c) is obtained,

$$\sqrt{(u-s)^2 + 4} \leq u + s \tag{7a}$$

$$\frac{1}{\sqrt{2}}\left\|C\right\|_F \leq \frac{1}{2}\mathrm{tr}(R^T B) - u \tag{7b}$$

$$u \geq 0, \tag{7c}$$

where u is an auxiliary variable. The constraints (7a),(7b) are also second-order cones. Each choice of R sets a different convexification to the inequality $1/\sigma \leq s$. We will explain how to choose R shortly.

To conclude, a convexification to the minimization of eq. (1) is obtained by minimizing the convex energy (4a), under the convex constraints (5),(7), where eqs. (5c),(7b) can also be merged into a single cone.

Optimizing Φ, Ψ. We represent the simplicial map Φ by a matrix $\Phi = [\mathbf{u}_1, ..., \mathbf{u}_n] \in \mathbb{R}^{2\times n}$, where n is the number of vertices in $\widehat{\mathbf{M}}$, and the i^{th} column represents $\Phi(\mathbf{v}_i)$. For each face t_j we choose an arbitrary orthonormal frame, and denote the affine map of Φ, restricted to t_j in this frame, by $\Phi|_{t_j}(\mathbf{x}) = A_j\mathbf{x} + \boldsymbol{\delta}$. We denote by $\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3} \in \mathbb{R}^{2\times 1}$ the column vectors representing the vertices of face t_j in the local frame, and $\mathbf{u}_{j_1}, \mathbf{u}_{j_2}, \mathbf{u}_{j_3}$ the respective columns of Φ. Then A_j is defined uniquely by

$$A_j\left[\mathbf{v}_{j_1}, \mathbf{v}_{j_2}, \mathbf{v}_{j_3}\right] D = \left[\mathbf{u}_{j_1}, \mathbf{u}_{j_2}, \mathbf{u}_{j_3}\right] D,$$

where $D = \left(I - \frac{1}{3}\mathbf{1}\mathbf{1}^T\right)$ is the centering matrix subtracting the centroid from each column. Solving this equation for each face t_j in preprocess yields a constant linear representation of each A_j in terms of the variables Φ of the simplicial map. We similarly set A'_j to be the matrix representing the linear part of Ψ restricted to the face t'_j. If one of the triangles is close to being degenerate, the above system might be ill-conditioned and might cause numerical problems. In practice, we have worked with triangular meshes possessing a condition number of up to 10^5 without experiencing any deterioration in performance of our algorithm.

Eq. (1) measures the isometric distortion of a single affine map. Integrating eq. (1) over all faces of $\widehat{\mathbf{M}}$ leads to the following energy measuring the isometric distortion of the simplicial mapping Φ:

$$E_p^{\widehat{\mathbf{M}}\to\mathbb{R}^2}(\Phi) = \left[\sum_{j\in J} E(A_j)^p \left|t_j\right|\right]^{1/p}, \tag{8}$$

where J is the index set of the faces $\mathbf{T}_{\widehat{\mathbf{M}}}$ and $|t_j|$ denotes the area of face t_j. Similarly, we define $E_p^{\widehat{\mathbf{N}}\to\mathbb{R}^2}(\Psi)$, with J' as the index set of the faces $\mathbf{T}_{\widehat{\mathbf{N}}} = \left\{t'_j\right\}_{j\in J'}$. We then compute Φ, Ψ by solving

$$\min_{\Phi,\Psi} \quad E_p^{\widehat{\mathbf{M}}\to\mathbb{R}^2}(\Phi) + E_p^{\widehat{\mathbf{N}}\to\mathbb{R}^2}(\Psi) \tag{9a}$$

$$\text{s.t.} \quad \Phi(\mathbf{x}) = \Psi(\nu(\mathbf{x})), \quad \forall \mathbf{x} \in \partial\widehat{\mathbf{M}}, \tag{9b}$$

Algorithm 2: Joint Flattening

```
Input:
$\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$, disk-type meshes
$\nu : \partial\widehat{\mathbf{M}} \to \partial\widehat{\mathbf{N}}$ bijective boundary correspondence
Output:
$\Phi, \Psi$, liftable flattenings

Initialize $\Phi_0, \Psi_0$ with Tutte's embedding
while $\Phi_n, \Psi_n$ not converged do
    Set $R_j, \forall j \in J$ using polar decomposition of $A_j$ of $\Phi_{n-1}$.
    Set $R'_j, \forall j \in J'$ using polar decomposition of $A'_j$ of $\Psi_{n-1}$.
    Set $\Phi_n, \Psi_n$ by optimizing (10),
        using $\{R_j\}_{j\in J}, \{R'_k\}_{k\in J'}$ as the rotations in eq. (7b)
Output $\Phi_n, \Psi_n$
```

using the convex formulation (for $p \geq 1$):

$$\min_{\Phi,\Psi} \left[\sum_{j\in J} h\left(r_j, s_j\right)^p \left|t_j\right|\right]^{1/p} + \left[\sum_{j\in J'} h\left(r'_j, s'_j\right)^p \left|t'_j\right|\right]^{1/p} \tag{10a}$$

$$\text{s.t.} \quad \Phi(\mathbf{x}) = \Psi(\nu(\mathbf{x})), \quad \forall \mathbf{x} \in \partial\widehat{\mathbf{M}} \tag{10b}$$

$$\begin{cases} \left\{A_j, r_j, s_j\right\}_{j\in J} \\ \left\{A'_j, r'_j, s'_j\right\}_{j\in J'} \end{cases} \text{satisfy eqs. } (5), (7). \tag{10c}$$

The only missing detail is how to set the rotations R_j, R'_j needed for setting eq. (7b) for each face t_j, t'_j (resp.). As mentioned before, the rotations R_j, R'_j select convex subsets of the non-convex part of the isometric distortion optimization. We initialize these rotations from feasible initial mappings Φ_0, Ψ_0 (detailed below) by defining R_j (R'_j) to be the rotation part of the polar decomposition of the initial A_j (A'_j), and solve (the now feasible, see [Lipman 2012]) problem (10). We then extract the rotations in the same manner from the solution and resolve until convergence, which usually happens in 3-6 iterations. The entire process is summarized in Algorithm 2.

The algorithm is initialized by computing bijective convex combination mappings Φ_0, Ψ_0 (using Mean Value Coordinates [Floater 2003b]), also known as *Tutte's* embeddings, of the two meshes $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$, to a convex polygon inscribed in the unit disc.

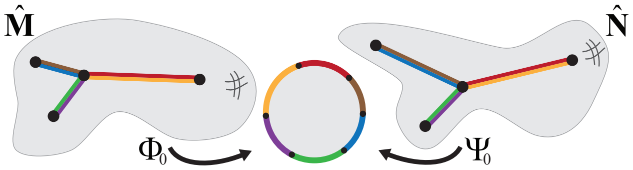


Figure 4: *Mapping the cut meshes to Tutte's embedding.*

The boundaries $\partial\widehat{\mathbf{M}}, \partial\widehat{\mathbf{N}}$ are constrained uniformly to the boundary of the unit disk such that all $\mathbf{v}_i \in \partial\widehat{\mathbf{M}}$ and $\nu(\mathbf{v}_i) \in \partial\widehat{\mathbf{N}}$ are assigned to the same boundary point on the unit circle, that is $\Phi_0(\mathbf{v}_i) = \psi_0(\nu(\mathbf{v}_i))$ (refer to subsection 3.4 for details). See Figure 4 for an illustration. This procedure is guaranteed to yield Φ_0, Ψ_0 that are feasible for problem (10) (see [Floater 2003a]), but the distortion can be arbitrarily high.

In our experiments, we used different choices of $p \in \{2, \infty\}$ and h (see eqs. (2),(3)). We decided to use to $p = 2$, and h as defined in eq. (3), due to a slightly better regularity behaviour.

Liftability of Φ, Ψ**.** The flattenings Φ, Ψ resulting from the optimization procedure detailed above are guaranteed to be liftable, as they satisfy the three conditions of Theorem 1. In the first iteration of the algorithm, a convex functional (10a) is optimized with the convex constraints (10b-10c), for which the initial flattenings Φ_0, Ψ_0 are feasible. Φ_0, Ψ_0 are liftable as they are both bijective and one can set $f = \Psi_0^{-1} \circ \Phi_0$, hence Condition (a) (Section 3.2) is satisfied. Let Φ_1, Ψ_1 be the (global) minimizer of the convex problem (10). Due to the convexity of the constraints, the flattenings Φ_t, Ψ_t defined by $\Phi_t = (1-t)\Phi_0 + t\Phi_1$, $\Psi_t = (1-t)\Psi_0 + t\Psi_1$ satisfy (10b-10c) as well. From (10b) it follows that $\Phi_t(\mathbf{x}) = \Psi_t(\nu(\mathbf{x}))$ for all $\mathbf{x} \in \partial\widehat{\mathbf{M}}$, and hence Condition (c) is satisfied. Lastly, to see that Condition (b) is satisfied, note the convex constraints (7), which prevent the smallest singular values from becoming zero, guarantee that Φ_t, Ψ_t, $t \in [0,1]$ do not degenerate any face of $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ (resp.), that is $\det(A_j) \neq 0$, $\det(A'_j) \neq 0$. Since $\det(A_j)$, $\det(A'_j)$ are continuous functions of t and are positive at $t = 0$, they are positive for all $t \in [0,1]$. Therefore, Φ_t, Ψ_t, $t \in [0,1]$ do not flip any of the faces of $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ (resp.). A simplicial map of a triangular mesh that does not flip any face is locally injective over faces and edges, but not necessarily at the vertices. Nevertheless, in our case, local injectivity at interior vertices is guaranteed and can be also guaranteed at boundary vertices via a slight change to the algorithm as explained below. With that settled, all lifting conditions hold for Φ_1, Ψ_1. By induction, the final flattenings Φ, Ψ, achieved by the algorithm after a finite number of iterations, are also liftable.

We end this section with the explanation of the local injectivity at vertices. Consider the homotopy (continuous deformation) Φ_t, Ψ_t as defined above. As Φ_t, Ψ_t do not flip faces, the sum of angles at a vertex is a continuous function of t. Since for interior vertices the sum of angles also has to be an integer multiple of 2π, we conclude that for interior vertices the sum of angles is a constant function. Since for Φ_0, Ψ_0 the sum of angles is 2π for interior vertices, the maps Φ_1, Ψ_1 maintain the angle sums of 2π at interior vertices. By the orientation preservation of the faces, local injectivity of Φ_1, Ψ_1 at interior vertices is implied. Therefore, the only breach of local-injectivity can occur at boundary vertices of $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$, when the angle sum at boundary vertices equals or exceeds 2π. In practice, this rarely happens, and in any case can be treated by simply completing each one-ring of a boundary vertex to a closed one-ring by adding two "ghost" faces and constraining them to avoid degeneracy (e.g., bounding the condition of their transformation, similarly to [Lipman 2012]).

3.4 Cutting to disk topology

The last component of our algorithm (which is applied first in our pipeline) is cutting the surface meshes $\mathbf{M}, \mathbf{N}$ to the disk topology meshes $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ and supplying the bijective correspondence between their boundaries $\nu : \partial\widehat{\mathbf{M}} \to \partial\widehat{\mathbf{N}}$. We assume $\mathbf{M}, \mathbf{N}$ are genus zero, boundaryless, and orientable surface meshes. We discuss higher genus at the end of this section.

We receive as input a coarse set of k corresponding pairs of points $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{y}_i)\} \subset \mathbf{M} \times \mathbf{N}$, $i = 1, .., k$. We then construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, 2, ..., k\}$, that describes the cuts that should be performed on the meshes, that is, if $(i,j) \in \mathcal{E}$ is an edge in $\mathcal{G}$, then we cut $\mathbf{M}$ along the minimal length geodesic $\gamma^{\mathbf{M}}$ from $\mathbf{x}_i$ to $\mathbf{x}_j$, and cut $\mathbf{N}$ along the minimal length geodesic $\gamma^{\mathbf{N}}$ from $\mathbf{y}_i$ to $\mathbf{y}_j$. See Figure 5 for an illustration of the cut-graph computed from a set of corresponding pairs of points. Having the same cut-graph embedded on the two meshes is necessary for constructing consistent cuts.

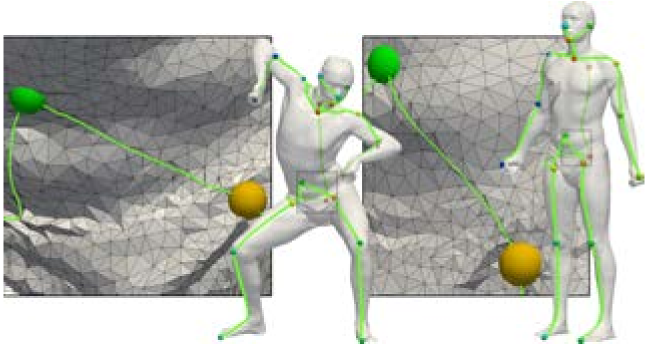


Figure 5: *The cut-graph, in the genus-zero case - a tree, shown in green. Input landmarks are shown as colored spheres.*

For genus zero surfaces, we build cut-graphs which are trees (to achieve disk-type surfaces after the cut). Specifically, we choose the cut tree so that its embedding in the two meshes is "minimal", as measured in terms of shortest-path geodesics on $\mathbf{M}$ and $\mathbf{N}$. That is, we assign for every pair $(i,j) \in \{1, 2, ..., k\} \times \{1, 2, ..., k\}$ the corresponding weight $\mathrm{dist}(\mathbf{x}_i, \mathbf{x}_j) + \mathrm{dist}(\mathbf{y}_i, \mathbf{y}_j)$, where $\mathrm{dist}(\cdot,\cdot)$ denotes the geodesic distance on the relevant surface, and construct a minimum spanning tree. Although this is a heuristic, we found it to provide a good choice of joint cuts for $\mathbf{M}, \mathbf{N}$, even for non-isometric surfaces (e.g., Figure 9).

The mesh cutting along the geodesics is performed sequentially, where for each edge $(i,j) \in \mathcal{E}$ we compute $\gamma^{\mathbf{M}}, \gamma^{\mathbf{N}}$ using the exact geodesics code of [Surazhsky et al. 2005]. Then, the intersections of the geodesic curves with the edges of the meshes are added as new vertices, and remeshed accordingly so that the geodesic passes through a connected set of edges. Next, the meshes are cut along the geodesic polygonal curves to get the disk-type meshes $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$: Adjacent triangles sharing an edge on the geodesic are separated, and their two shared vertices are duplicated.

Lastly, to be able to prescribe $\nu : \partial\widehat{\mathbf{M}} \to \partial\widehat{\mathbf{N}}$ we remesh the two boundaries $\partial\widehat{\mathbf{M}}, \partial\widehat{\mathbf{N}}$ to be isomorphic, so that ν becomes a simplicial map (taking edges bijectively to edges). To this end, we must prescribe the correspondence between the cuts $\gamma^{\mathbf{M}}, \gamma^{\mathbf{N}}$. The curves are parameterized so that they have unit speed close to the end points, and the speed changes smoothly to accommodate for any length differences of $\gamma^{\mathbf{M}}, \gamma^{\mathbf{N}}$ away from the end points. This is done by using the monotone smooth cubic polynomial $r(s) = s^2\frac{L-\ell}{\ell^2} - 2s^2(s-\ell)\frac{L-\ell}{\ell^3}$, where s is the arc-length parameter on the shorter curve among $\gamma^{\mathbf{M}}, \gamma^{\mathbf{N}}$, $\ell = \min\left\{\mathrm{length}(\gamma^{\mathbf{M}}), \mathrm{length}(\gamma^{\mathbf{N}})\right\}$, and $L = \max\left\{\mathrm{length}(\gamma^{\mathbf{M}}), \mathrm{length}(\gamma^{\mathbf{N}})\right\}$. Using this parameterization, we map the vertices of $\gamma^{\mathbf{M}}$ to $\gamma^{\mathbf{N}}$ and $\gamma^{\mathbf{N}}$ to $\gamma^{\mathbf{M}}$. We then remesh $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ to include these new boundary vertices. The remeshing of the boundaries is shown in the closeups in Figure 5. In the optimization of the flattenings Φ, Ψ, the boundary constraint in eq. (10b) is then realized by constraining corresponding vertices of the two remeshed boundaries to the same (unknown) location in the plane (a linear equality constraint).

Higher genus. For higher genus, only the cutting-to-disk step must be adapted, which can be done by using a more general cut-graph than a tree [Gu et al. 2002; Schreiner et al. 2004], as cutting along a tree will not yield a mesh homeomorphic to a disk. Since the output of this step is always disk-type meshes, the rest of the algorithm is agnostic to the genus.

4 Results

We present evaluations and experiments conducted using our algorithm. We begin by detailing the method we used for visualization of the surface mappings, continue with experiments of mappings generated for collections of pairs of surface models, and end with discussing different properties of the maps and comparisons to other methods.

4.1 Map visualization

To truly comprehend the different traits of a mapping between surfaces, one needs means to faithfully visualize and highlight its key properties: bijectiveness, smoothness and distortion. We use three different visualization techniques, as illustrated in Figure 6, each having different advantages. In all three methods, we transfer some function (scalar or vector field) from the left mesh to the right using our mapping, and then visualize it. First, we use the mapping to transfer a scalar function from the first mesh to the second mesh and display the corresponding isocontours (Figure 6, left). This is an effective means to visually assess the smoothness of the map, but using it makes it difficult to infer the actual correspondences. Second, we show the parameterization of the two surfaces by the joint flattenings from step 2 of our algorithm (Subsection 3.3), that is, we use the coordinates defined by Φ and Ψ to map a texture onto $\mathbf{M}$ and $\mathbf{N}$ (Figure 6, middle). Since $\Phi = \Psi \circ f$, this displays the parameterization of $\mathbf{M}$ as it is mapped to $\mathbf{N}$, and hence is quite effective in illustrating the map between the two meshes. However, the images on the surfaces will be discontinuous across the seams (i.e., cut-graphs), which will obscure the behaviour of the map across it. This leads to our third visualization method, specifically targeted to visualize the mapping in the vicinity of the cut-graph: We generate low-distortion texture coordinates on $\mathbf{M}$ by choosing a view point and selecting the largest connected component of visible triangles on $\mathbf{M}$. We then flatten it (using the low isometric distortion flattening from Subsection 3.3 for a single mesh) to get a good parameterization of the visible area, and then transfer the texture coordinates using the mapping to get corresponding texture coordinates on $\mathbf{N}$, see Figure 6, right. This method visualizes the mapping's properties well in the visible area of $\mathbf{M}$ but with the obvious drawback that some of the occluded parts of $\mathbf{M}$ are sometimes mapped to visible parts of $\mathbf{N}$ that are therefore untextured, see again Figure 6 (right).

Figure 6: *The three techniques for map-visualization used in this paper. Left: isocontours of a scalar function, middle: texture coordinates of the joint flattening, right: texturing the visible side on the left mesh and mapping to the right mesh (note textureless hands).*

4.2 Experiments

We tested our algorithm by computing mappings between pairs of surfaces from a wide baseline: starting from near-isometric, continuing with moderate isometric deviations and ending with extreme examples. We also compare to state-of-art mapping methods.

Near isometric - SCAPE dataset. We ran our algorithm on the SCAPE [Anguelov et al. 2005] dataset which contains 71 models of different poses of a human, with a given ground-truth 1-to-1 correspondence between their vertices. This dataset was originally generated by using a few landmarks and the correlated correspondence algorithm [Anguelov et al. 2004] to establish a dense correspondence set, followed by a standard non-rigid registration. In our experiment, we first marked 20 points on one model and then used the ground-truth correspondence of the models to automatically select the same 20 points on each of the other models and to cut each model automatically (see Figure 5 for an illustration of the landmarks and cut-graphs for two SCAPE models). We then drew random pairs of SCAPE models, excluding the ones whose automatic geodesic cutting failed, ending with 65 pairs of models. We ran our algorithm on all the pairs using the marked points as input. As can be seen in Figure 7, the resulting maps are smooth, have low distortion and produce a good correspondence overall.

Figure 7: *Mappings of 4 pairs of surfaces from the SCAPE dataset, generated automatically with 20 landmarks. The 2 top rows are visualized with the texture coordinates induced from the flattenings (seams highlighted in green). The two bottom rows are visualized with textures of the visible area.*

We have also compared our mappings to the original ground-truth mapping of the SCAPE models, computed as described in the beginning of this subsection. In Figure 8 we show a side-by-side comparison of our mapping and SCAPE's original map for a pair of models. While our map is similar in terms of accuracy, it is superior in terms of smoothness, as can be seen in the blowups. In the inset we display a histogram of the average isometric distortion over the faces (the isometric distortion of the j^{th} face is $\max\{\Sigma_j, 1/\sigma_j\}$) for all the 65 mappings. In blue, SCAPE's original mappings' distortions, and in green our mappings' distortions. We achieve considerably lower isometric-distortion levels than those of the original SCAPE maps.

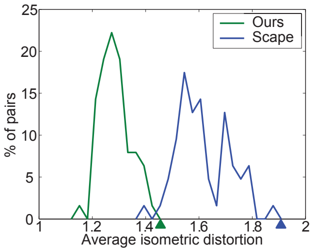


Figure 8: *Comparison of our map (bold) to the ground-truth map of SCAPE [Anguelov et al. 2005] (red frame). Left: textured source mesh. Note that the maps are similar but SCAPE's mapping lacks smoothness and exhibits high distortion.*

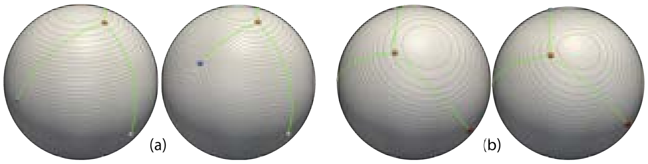
SHREC07 Dataset. We computed mappings between pairs of models from SHREC07 [Giorgi et al. 2007]. We marked landmarks between pairs of objects from the same class (ranging from 2 to 16 pairs of points). The resulting maps and their isometric distortion histograms are shown in Figure 9. Our method can handle a rather minimal set of landmarks, as in the vase models, in which only 4 points were marked. It is also well-suited for mapping nearly-isometric models, such as the ants and the octopuses. Note the limbs are mapped in a natural manner, without introducing excessive distortion. Mappings between differently-proportioned models are also handled gracefully by our method: for example, the boy is mapped to the woman, who has different limb lengths. All body parts, including the face and hair, are mapped correctly. In the coil pair, a short coil is mapped to a much longer one, necessitating a relatively high amount of distortion. Our method performs well also under these conditions and produces a bijection, which still has relatively low distortion levels (as much as possible, given the required stretch) and a natural, regular mapping. Our method can also easily handle thin and elongated parts, such as the glasses. Finally, as can be seen in the camel and cow example, the method is able to produce plausible mappings even between very different models.

Figure 10: *Mapping very different surfaces. Top row: mapping the boy model to the donkey. Anatomically corresponding areas are mapped to one another (see limbs and head). Bottom row: fandisk mapped to the stanford bunny. In both cases the maps are plausible with an expected distortion profile.*

Mappings between objects of different classes. Figure 10 shows examples of bijective mappings between models from different classes. The boy model is mapped to a donkey (top); notice the limbs, head, and body are mapped to one another, and that the mapping handles the tail, which requires introducing vast amounts of distortion, as it needs to be mapped to a relatively flat region. A map of the fandisk to the Stanford Bunny is shown at the bottom. Note the input correspondences are rather arbitrary, and the two models differ greatly. Nonetheless, the mapping is still plausible with a reasonable amount of distortion.

4.3 Map properties

Effect of the cut-graph on the mapping. The regularity across the cut is not part of the optimization problem (9) and therefore the resulting map, although continuous, may introduce stronger irregularity near the cut in presence of high isometric distortion.



The inset above shows an example where the cut-graphs contain one path that is considerably shortened (compare the paths that connect the blue and the orange dots in (a)), causing the map to warp around it. Other paths in this example approximately maintain their geodesic length, resulting in a more seamless mapping (see (b) for a different viewpoint of the same mapping).

Another example is shown in the inset, a blow-up from the side of the donkey in Figure 10. Note that near the cut the map is less regular, compared to more distant areas. Nevertheless, if the distortion of the flattenings Φ, Ψ is low near the cuts, the mapping will be regular across the cut, and in practice this case prevails. In this paper we made a special effort to display mappings with all (or most) cuts visible, see for example Figure 9, and in general, the mappings do not exhibit lack of regularity and are well-behaved in the cut areas.

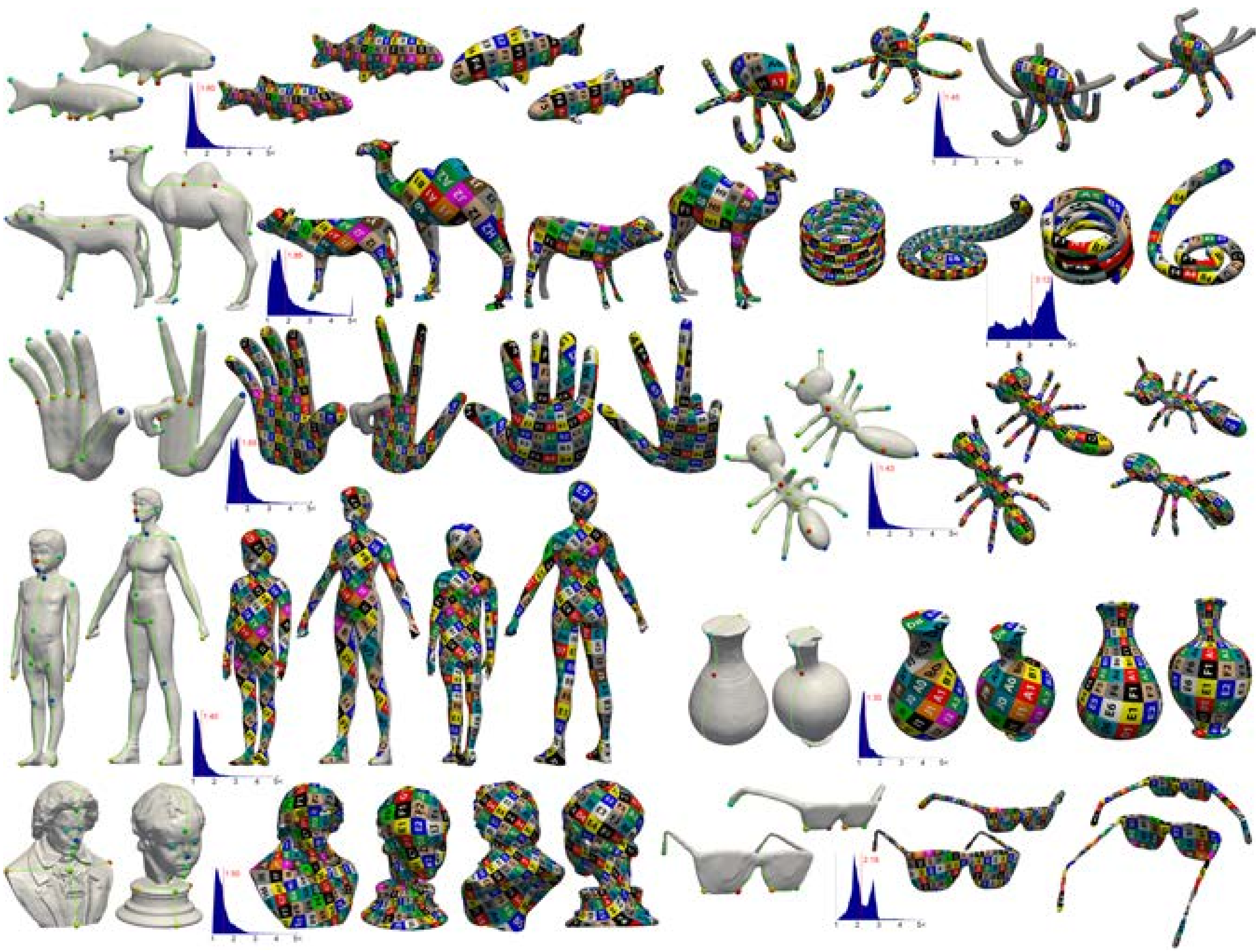


Figure 9: *Bijective mappings between pairs of models from different classes in SHREC07 [Giorgi et al. 2007]. The cut-graphs are shown in green, landmarks shown as colored spheres. The distortion histogram (mean marked in red) of each mapping is shown next to it.*

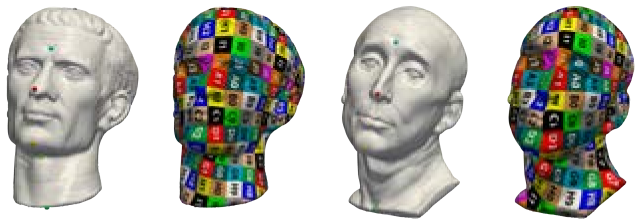


Figure 11: *The same pair from Figure 1 is mapped again with fewer landmarks (7 instead of 19) picked on the symmetry axis of the model. The mapping is less accurate than in Figure 1 but has good extrapolation properties, e.g., the mouth, eyes and cheeks.*

Number of landmarks. We have tested the quality of the map with respect to the number of landmarks prescribed by the user. As expected, adding more point constraints indeed improves the map's accuracy. However, even for a small number of points, the algorithm seems to produce reasonably well-behaved maps, with good extrapolation properties. Figure 11 shows the same head models from Figure 1 (left) but with 7 input points (instead of the 19 that were used in Figure 1) that are placed only along the line of symmetry of the model. Note that the map, although not accurate in areas far from the points (e.g. the ear), behaves well for most parts of the face, for example the mouth, eyes and cheeks areas.

Timings. The algorithm was implemented in Matlab using the YALMIP environment [Löfberg 2004] and Mosek's Second-Order Cone Program solver [Andersen and Andersen 1999]. Typical running times on a single 3.50GHz Intel i7 core for a pair of meshes with a total of 15k verts/30k faces are: 1 minute for setup, less than 7 minutes for optimization, and half a minute for the bijection lifting. For a pair with a total of 25k verts/50k faces: 2 minutes for setup, 13 minutes for optimization, and 1 minute for bijection lifting. We note that our code is not optimized and we believe significant speedups are possible.

4.4 Comparisons with other methods

Weighted Averages on Surfaces. We compared our method to the Weighted Averages on Surfaces framework of [Panozzo et al. 2013]. We ran their code on the pair of SCAPE models shown in Figure 7 (bottom row) and Figure 8, and provided their method with the same input landmarks we used as input (total of 20 pairs of points). The results are shown in Figure 12. In most areas the mapping is reasonable, however the method suffers from two drawbacks compared to ours: the map is not surjective (e.g., the white patches of missing texture on the scalp and on the torso in the middle figure), and not injective (e.g. the leg).

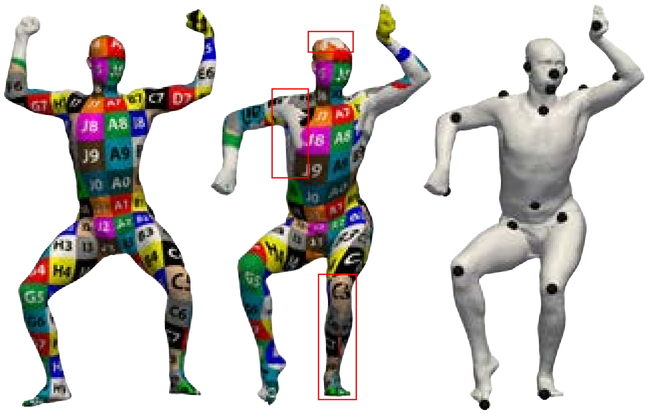


Figure 12: *Comparison of our algorithm to Weighted Averages (WA) on Surfaces [Panozzo et al. 2013] using the same input landmarks (right, in black). Left column: textured source model. Middle: the mapping produced by the WA method. Note the lack of injectivity in the leg region, and lack of surjectivity at the scalp. Compare to our result shown in Figure 7, bottom row and Figure 8.*

Blended Intrinsic Maps. We compared our method to Blended-Intrinsic-Maps (BIM) [Kim et al. 2011] by running our code on one of their examples, using the same input landmarks as the ones that generated the final blended map in their algorithm. The comparison is shown in Figure 13. Although it behaves well in large parts of the surface, BIM generates a map which is not bijective, and possesses high isometric distortion in some areas. For example, in the middle row closeup, BIM's map shrinks considerably at the tip of the wing (marked with an arrow). Also, the correspondence along the wing is inferior to our result (e.g., square H0 in the texture). The bottom-right inset shows the tail area where BIM is neither continuous nor bijective, while our algorithm (middle) generates an intuitive bijective correspondence to the tail of the original plane (left).

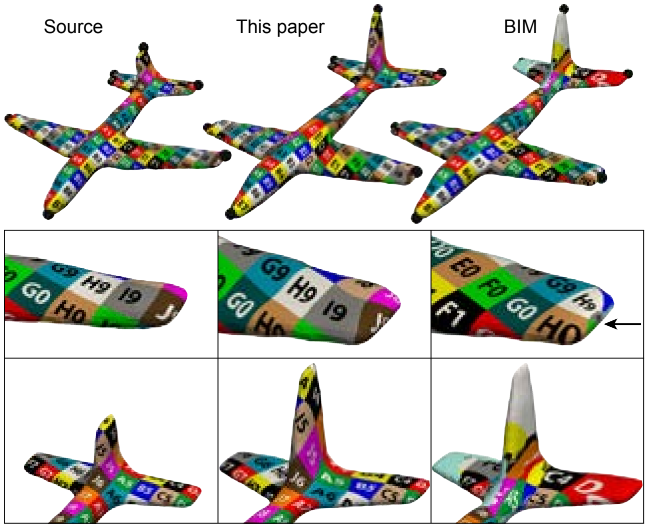


Figure 13: *Comparison of our algorithm (middle) to Blended Intrinsic Maps [Kim et al. 2011] (right) using the same input landmarks (in black). Left: source model. BIM can create high isometric distortion (e.g., middle row, marked with an arrow) which also causes bad alignment (see square H0), and can be non-bijective (e.g., bottom row).*

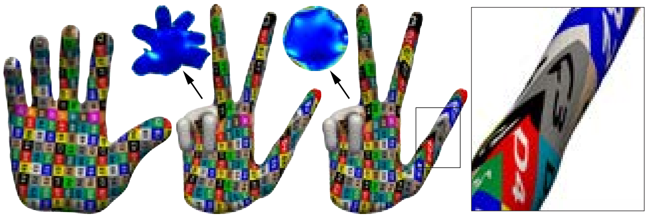


Figure 14: *The result of mapping the cut meshes to a fixed common domain (right), note the large amounts of distortion (a maximum of 1000) e.g. in the blowup. Left: source model. Middle: the result of our algorithm for the same input (same as in Figure 9).*

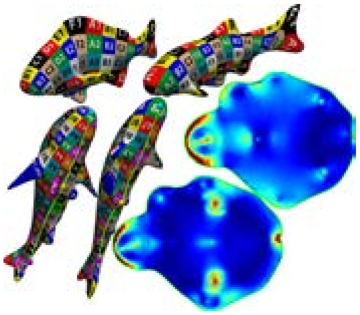
Fixed common base domain. We also compared our mapping to the bijective mapping between $\mathbf{M}$ and $\mathbf{N}$ computed by flattening the cut meshes via Φ_0, Ψ_0 to a convex polygonal domain in the plane using Tutte's embedding (Floater's convex combination maps [Floater 2003b]) and setting $f = \Psi_0^{-1} \circ \Phi_0$. Recall that this is the initialization stage of our flattening algorithm. Figure 14 shows this surface mapping f compared to our mapping in this case. Evidently, f exhibits high distortion, showing our optimization is indeed effective in reducing the distortion, while maintaining bijectivity.

5 Conclusion, limitations and future work

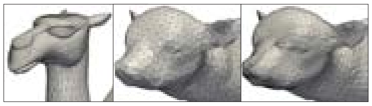
A conceptually simple algorithm for computing low-distortion continuous bijections between surface meshes from a sparse set of correspondences is presented. The experiments affirm it is robust and suitable for mapping nearly isometrical models, as well as ones which are considerably far from being isometric.

The main limitation of the method lies in the choice of cut-graphs. In case the correspondences are chosen poorly, the geodesic curves may pass through different regions on the two models, which are not naturally corresponding. In this case, one can simply move the points or add more correspondences in order to refine the choice of the cut-graph. However, this is a suboptimal solution and we mark an automatic algorithm for finding good corresponding cut-graphs as an interesting future work venue.

A second limitation is that the distortion of the bijection $f : \widehat{\mathbf{M}} \to \widehat{\mathbf{N}}$ is not directly optimized but rather is implicitly controlled via the optimization of the flattenings. If the flattenings have low distortion so will f. The converse is not true: it is possible to obtain a low distortion f lifted from high distortion flattenings Φ, Ψ, however we do not harness this fact in our algorithm. If Φ and/or Ψ possess high distortion (the cut meshes include high curvature areas in their interior), it is likely that f will have high distortion as-well. Such an example is displayed in the inset, where we marked only 3 landmarks on the bottom of the fish's body (bottom-left), forcing the fins and tail to be flattened with high distortion. Note the fins of the left fish are mapped to the body of the right fish, causing high distortion in the final bijection (see the yellow E4 square). On the right, the flattenings are shown, colored according to distortion.



Our method lends itself to many applications, such as transferring of various attributes (texture, normals and scientific data), parameterization onto non-flat surfaces, e.g. mapping to a sphere, in which case obtaining low distortion bijections can be beneficial, and, lastly, it enables mesh-transfer, i.e. transferring the topological structure of one mesh to the geometry of the second (see inset). One of our future goals is to generate databases of meshes with 1-to-1 vertex correspondence in the spirit of SCAPE.



The matter of visualization of surface maps also requires further exploration [Ovsjanikov et al. 2013]. The methods used in this paper provide much information regarding the map (e.g., smoothness, distortion, injectivity) but are sometimes hard to quickly grasp. An alternative technique is to transfer a characteristic color function, as shown above. This method lacks in detail but is easier to interpret. An interesting future research direction would be to create a hybrid of these two methods.

Acknowledgements This work was funded by the European Research Council (ERC Starting Grant "SurfComp"), the Israel Science Foundation (grant No. 1284/12) and the I-CORE program of the Israel PBC and ISF (Grant No. 4/11). The authors also thank the anonymous reviewers for their comments and suggestions.

References

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of all body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (proc. SIGGRAPH)*.

ANDERSEN, E. D., AND ANDERSEN, K. D. 1999. *The MOSEK interior point optimization for linear programming: an implementation of the homogeneous algorithm*. Kluwer Academic Publishers, 197–232.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., PANG, H., AND DAVIS, J. 2004. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. *Proc. of the Neural Information Processing Systems*.

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (July), 408–416.

BESL, P., AND MCKAY, N. 1992. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*.

BOYD, S. P., AND VANDENBERGHE, L. 2004. *Convex optimization*. Cambridge university press.

BRADLEY, D., POPA, T., SHEFFER, A., HEIDRICH, W., AND BOUBEKEUR, T. 2008. Markerless garment capture. *ACM Trans. Graph. 27*, 3 (Aug.), 99:1–99:9.

BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS)*.

BROWN, B. J., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-d scans. *ACM Trans. Graph. 26*, 3 (July).

CHUI, H., AND RANGARAJAN, A. 2003. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89*, 2-3, 114–141.

FLOATER, M. S., AND HORMANN, K. 2005. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*. Springer, 157–186.

FLOATER, M. 2003. One-to-one piecewise linear mappings over triangulations. *Mathematics of Computation 72*, 242, 685–696.

FLOATER, M. S. 2003. Mean value coordinates. *Computer Aided Geometric Design 20*, 1, 19–27.

GIORGI, D., BIASOTTI, S., AND PARABOSCHI, L. 2007. SHREC: SHape REtrieval Contest: Watertight models track. *http://watertight.ge.imati.cnr.it/*.

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. *ACM Trans. Graph. 21*, 3 (July), 355–361.

GU, X., WANG, Y., CHAN, T. F., THOMPSON, P. M., AND YAU, S.-T. 2004. Genus zero surface conformal mapping and its application to brain surface mapping. *Medical Imaging, IEEE Transactions on 23*, 8, 949–958.

HORMANN, K., AND GREINER, G. 2000. MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, P.-J. Laurent, P. Sablonnière, and L. L. Schumaker, Eds., Innovations in Applied Mathematics. Vanderbilt University Press, Nashville, TN, 153–162.

HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice video files associated with this course are available from the citation page. In *ACM SIGGRAPH 2007 Courses*, ACM, New York, NY, USA, SIGGRAPH '07.

HUANG, Q., ADAMS, B., WICKE, M., , AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proc. SGP 2008)*.

JAIN, V., ZHANG, H., AND VAN KAICK, O. 2007. Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling 13*, 1, 101–124.

KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. 2011. Blended intrinsic maps. *ACM Trans. Graph. 30*, 4 (July), 79:1–79:12.

KRAEVOY, V., AND SHEFFER, A. 2004. Cross-parameterization and compatible remeshing of 3d models. *ACM Trans. Graph. 23*, 3 (Aug.), 861–869.

LEE, A. W. F., DOBKIN, D., SWELDENS, W., AND SCHRÖDER, P. 1999. Multiresolution mesh morphing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 343–350.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum 27*, 5, 1421–1430.

LIN, J. L., CHUANG, J. H., LIN, C. C., AND CHEN, C. C. 2003. Consistent parametrization by quinary subdivision for remeshing and mesh metamorphosis. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, ACM, New York, NY, USA, GRAPHITE '03, 151–158.

LIPMAN, Y., AND FUNKHOUSER, T. 2009. Mobius voting for surface correspondence. *ACM Trans. Graph. 28*, 3 (July), 72:1–72:12.

LIPMAN, Y. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph. 31*, 4 (July), 108:1–108:13.

LÖFBERG, J. 2004. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*.

MARX, M. L. 1974. Extensions of normal immersions of S^1 into R^2. *Transactions of the American Mathematical Society 187*, 309–326.

MÉMOLI, F., AND SAPIRO, G. 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics 5*, 3, 313–347.

MICHIKAWA, T., KANAI, T., FUJITA, M., AND CHIYOKURA, H. 2001. Multiresolution interpolation meshes. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, 60–69.

OVSJANIKOV, M., MÉRIGOT, Q., MÉMOLI, F., AND GUIBAS, L. 2010. One point isometric matching with the heat kernel. In *Computer Graphics Forum (Proc. of SGP)*.

OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., AND GUIBAS, L. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July), 30:1–30:11.

OVSJANIKOV, M., BEN-CHEN, M., CHAZAL, F., AND GUIBAS, L. J. 2013. Analysis and visualization of maps between shapes. *Comput. Graph. Forum 32*, 6, 135–145.

PANOZZO, D., BARAN, I., DIAMANTI, O., AND SORKINE-HORNUNG, O. 2013. Weighted averages on surfaces. *ACM Trans. Graph. 32*, 4 (July), 60:1–60:12.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3d scan completion. In *Symposium on Geometry Processing*, 23–32.

PRAUN, E., SWELDENS, W., AND SCHRÖDER, P. 2001. Consistent mesh parameterizations. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 179–184.

SANDER, P. V., SNYDER, J., GORTLER, S. J., AND HOPPE, H. 2001. Texture mapping progressive meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 409–416.

SCHREINER, J., ASIRVATHAM, A., PRAUN, E., AND HOPPE, H. 2004. Inter-surface mapping. *ACM Trans. Graph. 23*, 3 (Aug.), 870–877.

SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis. 2*, 2 (Jan.), 105–171.

SOLOMON, J., NGUYEN, A., BUTSCHER, A., BEN-CHEN, M., AND GUIBAS, L. 2012. Soft maps between surfaces. *Comp. Graph. Forum 31*, 5 (Aug.), 1617–1626.

SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-distortion piecewise mesh parameterization. In *Proceedings of the Conference on Visualization '02*, IEEE Computer Society, Washington, DC, USA, VIS '02, 355–362.

SURAZHSKY, V., SURAZHSKY, T., KIRSANOV, D., GORTLER, S. J., AND HOPPE, H. 2005. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph. 24*, 3 (July), 553–560.

TEVS, A., BOKELOH, M., M.WAND, SCHILLING, A., AND SEIDEL, H.-P. 2009. Isometric registration of ambiguous and partial data. *In: Proc. IEEE Conference on Computer Vision and Pattern Recognition*.

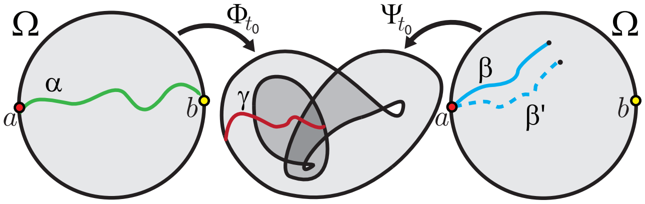
VAN KAICK, O., ZHANG, H., HAMARNEH, G., AND COHEN-OR, D. 2011. A survey on shape correspondence. *Computer Graphics Forum 30*, 6, 1681–1707.

ZENG, W., LUO, F., YAU, S.-T., AND GU, X. D. 2009. Surface quasi-conformal mapping by solving beltrami equations. In *Mathematics of Surfaces XIII*. Springer Berlin Heidelberg, 391–408.

Appendix

We provide a sketch of the proof of Theorem 1:
We start with reformulating the problem as follows. Let Φ, Ψ, ν be as stated in the theorem, and let ϕ, ψ be homeomorphic mappings of $\widehat{\mathbf{M}}, \widehat{\mathbf{N}}$ to a convex polygonal domain Ω with polygonal boundary $P = \partial\Omega$, such that $\phi(\mathbf{x}) = \psi(\nu(\mathbf{x}))$, for all $\mathbf{x} \in \partial\widehat{\mathbf{M}}$ (the existence of ϕ, ψ is guaranteed by Tutte's embedding theorem [Floater 2003a]). Let $\Phi' = \Phi \circ \phi^{-1}, \Psi' = \Psi \circ \psi^{-1}$. Φ, Ψ are liftable iff Φ', Ψ' are liftable, hence it is enough to consider the case $\widehat{\mathbf{M}} = \Omega = \widehat{\mathbf{N}}$ and mappings $\Phi', \Psi' : \Omega \to \mathbb{R}^2$ with the identity boundary map $\nu' : P \to P$, and prove that Φ', Ψ' are liftable.

Since Φ, Ψ satisfy the lifting conditions so do Φ', Ψ'. Let Φ_t, Ψ_t, $t \in [0, 1]$ be the relevant homotopies where $\Phi_1 = \Phi', \Psi_1 = \Psi'$. In this formulation Φ_t, Ψ_t are locally injective *extensions* to the boundary mapping $g_t : P \to \mathbb{R}^2$ defined by $g_t(\mathbf{x}) = \Phi_t(\mathbf{x}) = \Psi_t(\mathbf{x})$ for all $\mathbf{x} \in P$. Locally injective extensions of circle embeddings were studied e.g. in [Marx 1974].
For any fixed t, a sufficient condition for the existence of a homeomorphism $f : \Omega \to \Omega$ satisfying $\Phi_t = \Psi_t \circ f$ is that for any simple interior curve $\gamma \subset \Omega$ connecting two boundary points a and b in P, $\Psi_t^{-1}(\Phi_t(\gamma))$ contains a simple interior curve connecting a and b , and vice-versa. Assume toward a contradiction that there exists a first time t_0 where this condition fails to hold. It can be shown that in this case there must exist curves $\alpha, \beta \subset \Omega$ such that $\Phi_{t_0}(\alpha) = \Psi_{t_0}(\beta) = \gamma$, but α connects a, b while β starts at a but ends away from the boundary P.



Let $\gamma_t = \Phi_t\,(\alpha)$. By looking at time $t = t_0 - \epsilon$ of the homotopy for sufficiently small $\epsilon > 0$ one can show there is one, and only one, $\beta' \subset \Omega$ starting from a such that $\Phi_t(\alpha) = \gamma_t = \Psi_t(\beta')$. For a small enough ϵ, β' is as close to β as desired, namely cannot reach b. This means that the sufficient condition failed to hold already at time $t = t_0 - \epsilon$, contradicting the assumption that t_0 is the first time the sufficient condition fails.