Distortion-Minimizing Injective Maps Between Surfaces

PATRICK SCHMIDT, RWTH Aachen University
JANIS BORN, RWTH Aachen University
MARCEL CAMPEN, Osnabrück University
LEIF KOBBELT, RWTH Aachen University

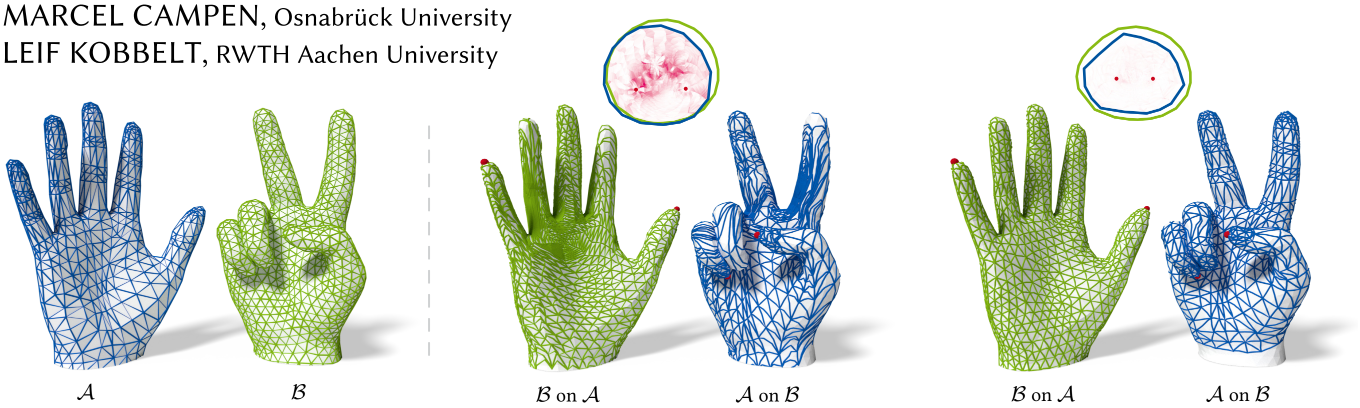

$\mathcal{A}$
$\mathcal{B}$
$\mathcal{B}$ on $\mathcal{A}$
$\mathcal{A}$ on $\mathcal{B}$
$\mathcal{B}$ on $\mathcal{A}$
$\mathcal{A}$ on $\mathcal{B}$

Fig. 1. Left: input meshes $\mathcal{A}$ and $\mathcal{B}$ of disk topology. Center and right: these meshes are continuously mapped onto each other via an intermediate flat domain (top) by composing two planar parametrizations. The map is constrained by just two landmarks (thumb and pinky). Center: both parametrizations are optimized for isometric distortion; the composed map, however, has high distortion (visualized in red on top). Right: our method directly optimizes the distortion of the composed map in an end-to-end manner, naturally aligning similarly curved regions as they map to each other with lower isometric distortion.

The problem of discrete surface parametrization, i.e. mapping a mesh to a planar domain, has been investigated extensively. We address the more general problem of mapping *between* surfaces. In particular, we provide a formulation that yields a map between two disk-topology meshes, which is continuous and injective by construction and which locally minimizes intrinsic distortion. A common approach is to express such a map as the composition of two maps via a simple intermediate domain such as the plane, and to independently optimize the individual maps. However, even if both individual maps are of minimal distortion, there is potentially high distortion in the composed map. In contrast to many previous works, we minimize distortion in an end-to-end manner, directly optimizing the quality of the composed map. This setting poses additional challenges due to the discrete nature of both the source and the target domain. We propose a formulation that, despite the combinatorial aspects of the problem, allows for a purely continuous optimization. Further, our approach addresses the non-smooth nature of discrete distortion measures in this context which hinders straightforward application of off-the-shelf optimization techniques. We demonstrate that, despite the challenges inherent to the more involved setting, discrete surface-to-surface maps can be optimized effectively.

CCS Concepts: • **Computing methodologies → Computer graphics**; **Mesh models**; **Mesh geometry models**; *Shape modeling*.

Additional Key Words and Phrases: Surface parametrization, inter-surface mapping, cross-parametrization, embedding, intersection mesh

ACM Reference Format:
Patrick Schmidt, Janis Born, Marcel Campen, and Leif Kobbelt. 2019. Distortion-Minimizing Injective Maps Between Surfaces. *ACM Trans. Graph.* 38, 6, Article 1 (November 2019), 15 pages. https://doi.org/10.1145/3355089.3356519

1 INTRODUCTION

Maps between surfaces are an important tool in Geometry Processing. They are required to transfer information (such as attributes, features, texture) between objects, to co-process multiple objects (such as shape collections, animation frames), to interpolate between objects (e.g. for shape morphing), or to embed and parametrize objects (e.g. for template fitting). We here consider the case of discrete surfaces (triangle meshes) that are of disk topology.

A special case is mapping between a surface and the plane, i.e. the problem of discrete surface parametrization. There is vast literature on this topic, with many improvements and extensions proposed each year. The general case of maps between (non-planar) surfaces, by contrast, has received less treatment—it is significantly harder to handle due to the aspect of combinatorial complexity incurred by both source and target domain being discrete. In the planar parametrization scenario (mapping a discrete surface to the continuous plane) this aspect does not have to be dealt with. However, the composition of two independently optimized surface-to-plane maps does not generally yield an optimal surface-to-surface map.

In many use cases it is, similar to the surface parametrization setting, desirable or required that the map between surfaces is

- continuous,
- bijective,
- low-distortion.

In some applications, the map is from a source to a target surface, while in others the setting is symmetric; in these cases, ideally a method to construct such maps should be

- symmetric

as well, i.e. independent of order. Many previous approaches to the problem of, under given constraints, constructing and optimizing maps between surfaces aim for one or more of these properties, but guarantees can rarely be given.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, https://doi.org/10.1145/3355089.3356519.

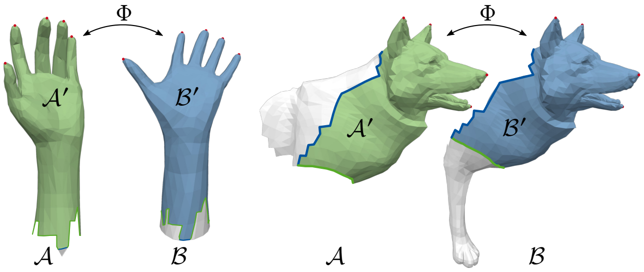


Fig. 2. We compute a bijective map between corresponding regions $\mathcal{A}'$ and $\mathcal{B}'$ of disk-topology surfaces. The extent of the regions covered by the map does not have to be fixed, but is subject to the optimization.

1.1 Contribution

We present a formulation that yields a map $\Phi : \mathcal{A}' \to \mathcal{B}'$ between parts $\mathcal{A}'$ and $\mathcal{B}'$ of disk-topology surfaces $\mathcal{A}$ and $\mathcal{B}$. The map is a continuous bijection between $\mathcal{A}'$ and $\mathcal{B}'$ and minimizes a distortion measure $E(\Phi) \to \min$. We say the map is injective when considering it with respect to one entire surface (e.g. $\mathcal{A}' \to \mathcal{B}$). Bijectivity between $\mathcal{A}'$ and $\mathcal{B}'$ is guaranteed by construction and the formulation is symmetric—the result is not inherently biased by the order of input surfaces, i.e. Φ and Φ^{-1} are on equal footing. The extent of $\mathcal{A}'$ and $\mathcal{B}'$ on the input surfaces does not have to be specified in advance; it is determined by our method based on $E(\Phi)$. We employ a second-order optimization method, tailored to the specific properties of the setting. The full algorithm is summarized in Appendix A.

1.2 Approach

Like several previous works, e.g. [Aigerman and Lipman 2015; Aigerman et al. 2014, 2015; Kanai et al. 1997; Lipman and Funkhouser 2009; Litke et al. 2005; Tierny et al. 2011], we make use of an intermediate non-discrete domain (the plane). Our map is composed of two maps, $\Phi = g^{-1} \circ f$, via this intermediate domain. The non-discrete nature of this domain allows handling the combinatorial aspect of the problem—which vertex maps into which face, which edge image crosses which edges—implicitly and thus efficiently.

Unlike many previous works, however, we formulate the objective end-to-end; aside from effects related to discretization and regularization, the result is oblivious of the intermediate domain. In other words, we directly optimize $E(\Phi)$, not $E(f)$ and $E(g)$. This is important because even minimal distortion of f and g does not imply low distortion of Φ. In fact, the optimal map Φ can be expressed via maps f and g of arbitrarily high distortion. It all depends on their interplay; distortion may mutually cancel or amplify (cf. Figure 1), as also recently pointed out by [Ezuz et al. 2019b]. The end-to-end optimization approach naturally favors alignment of similarly curved surface regions, as they can be mapped to each other with lower intrinsic distortion. This effect does not at all occur when maps f and g are constructed/optimized individually, as also described by [Ezuz et al. 2019b] and [Schreiner et al. 2004].

Using a planar intermediate domain allows us to compute maps between disk-topology shapes whose boundaries are not in direct correspondence to each other (e.g. when $\mathcal{A}$ and $\mathcal{B}$ are of different extent or have differently shaped boundary curves). While our method does provide control over the boundary of the embedding (by fixing boundary vertex correspondences), it demonstrates its full potential when the boundary embedding emerges naturally as output of the optimization (cf. Figure 2).

Our formulation is symmetric in two ways: (1) Due to the symmetry of the setup (i.e. no dependence on the order of $\mathcal{A}$ and $\mathcal{B}$), the optimization is oblivious to whether the map is used to pull information from $\mathcal{A}'$ to $\mathcal{B}'$ using Φ^{-1} (e.g. texture transfer), push from $\mathcal{A}'$ to $\mathcal{B}'$ using Φ (e.g. polyline transfer), or exchange information both ways. (2) We are able to optimize a symmetric distortion measure (such as the symmetric Dirichlet energy) that equally penalizes stretching and shrinking. This is of central importance to the quality of the map, even if it is eventually used to transport information one-way.

Few works have considered end-to-end optimization in the setting of continuous injective maps before (cf. Section. 2.1). Methods that operate in this setting typically use local distortion assessment and optimization [Schreiner et al. 2004] or some form of domain resampling [Litke et al. 2005]. Through our formulation we now enable direct and efficient global updates using second-order techniques, without any need for approximation or resampling. This direct global formulation brings our setting close to the surface-to-plane case, where enormous advances have been made in the optimization of geometric energies during the past few years.

Remark: In this paper we do not address the problem of shape correspondence or shape registration [Van Kaick et al. 2011], i.e. the problem of estimating correspondences given merely two independent surfaces as input. We assume either landmark constraints or a (possibly high-distortion) map initialization are given, and focus on the problem of accordingly optimizing a map between surfaces for low distortion in a continuous and bijective manner.

2 RELATED WORK

Taking the term "map" broadly, there is a vast number of works on maps between surfaces. We thus distinguish between methods that, like ours, consider continuous injective/bijective maps (homeomorphisms) of surfaces, and methods that do not, i.e. methods that consider other, looser forms of maps or correspondences:

2.1 Surface Homeomorphisms

A variety of ways, direct and indirect (via intermediate domains), have been used to define and represent homeomorphisms between discrete surfaces.

- Direct: [Schreiner et al. 2004] (map representation using barycentric vertex coordinates and explicit edge intersections); can be tailored to asymmetric (fine mesh – coarse mesh) setting [Guskov et al. 2000; Khodakovsky et al. 2003; Pietroni et al. 2010; Tarini et al. 2011].
- Indirect, via the plane: [Aigerman and Lipman 2015; Aigerman et al. 2014, 2015; Kanai et al. 1997; Kim et al. 2011; Lipman and Funkhouser 2009; Litke et al. 2005; Tierny et al. 2011], [Weber and Zorin 2014] (in this latter case used for planar shapes).
- Indirect, via the hyperbolic plane: [Aigerman and Lipman 2016; Shi et al. 2017].

- Indirect, via the sphere: [Aigerman et al. 2017; Alexa 2000; Asirvatham et al. 2005; Baden et al. 2018].
- Indirect, via a base mesh: [Kraevoy and Sheffer 2004; Praun et al. 2001].

End-to-End Distortion Minimization. To the best of our knowledge, the only descriptions of symmetric, bijective, and end-to-end distortion optimization of inter-surface maps are [Schreiner et al. 2004] and [Litke et al. 2005]. Similar to us, [Schreiner et al. 2004] represent the map via a mutual tessellation and consider the symmetric Dirichlet energy. Their method does not use a global intermediate domain and optimization is performed by locally moving single vertices along random search directions. Convergence to a desirable minimum inherently requires using a multi-resolution approach and is affected by numerical instabilities. The method by [Litke et al. 2005] employs a setup that is similar to ours (overlaying two parametrizations in the plane and considering an end-to-end distortion measure). However, optimization is performed by re-sampling the parameter domain on a regular grid and applying image matching techniques, which assumes that the surfaces can be mapped to the plane with relatively moderate distortion.

Other methods are tailored to the mapping between one fine and one coarse (base) mesh, e.g. [Guskov et al. 2000; Khodakovsky et al. 2003] (distortion reduction via local 1-ring relaxation), [Pietroni et al. 2010; Tarini et al. 2011] (distortion optimization using alternating semi-local optimization of a generic objective).

A specific scenario where minimal distortion of individual maps f and g to an intermediate domain does imply minimal distortion of the composed map $g^{-1} \circ f$, is that of zero-distortion maps. This is, however, possible only for either very restricted classes of shapes or very specific distortion measures. For instance, [Li et al. 2008a] and [Lipman and Funkhouser 2009] construct (quasi-)conformal maps between surfaces.

2.2 Other Surface Maps

Constructing and optimizing maps that are continuous and bijective is a hard problem in general. Depending on the use cases, one may be able to relax requirements in this regard and work with other, less strictly defined notions of maps. Consequently, a variety of techniques have been proposed to construct maps of various kinds.

Vertex-Based Maps. Instead of defining the map for each point on a surface, some representations focus on vertices and their images only. Examples are vertex-to-vertex maps, e.g. [Rodolà et al. 2015], or vertex-to-mesh maps, e.g. [Campen et al. 2012; Ezuz et al. 2019a,b; Yang et al. 2018]. These define the map natively for vertices only. Images of other points (on edges or in faces) would have to be deduced if needed. Performing this deduction (by some form of local but globally consistent parametrization) in an efficient, consistent, and robust manner is a non-trivial problem. Note that the vertex-to-mesh setting furthermore is inherently asymmetric; [Ezuz et al. 2019a,b] propose to optimize a forward and a backward map in tandem to symmetrize the construction. We compare to [Ezuz et al. 2019b] in Figure 12.

Non-Sharp Maps. Various kinds of non-sharp, i.e. not point-to-point, maps between surfaces have been described, for instance soft maps [Solomon et al. 2012], functional maps [Ovsjanikov et al. 2012], and optimal transport based maps [Mandad et al. 2017].

While techniques have been described to convert such distribution based maps to vertex-to-vertex maps [Ovsjanikov et al. 2012; Rodolà et al. 2015], conversion to continuous, guaranteed injective maps is an unsolved challenge. Thus, these methods are not immediately applicable to the injective mapping problem we are targeting.

Non-Rigid Registration. Rather than exploiting the 2-manifold nature of surfaces and defining the map accordingly, some methods employ maps in the 3-dimensional ambient space. Essentially, one surface is deformed so as to conform to the other. Examples are [Bouaziz et al. 2013; Huang et al. 2008; Li et al. 2008b; Sharf et al. 2006; Tam et al. 2013; Wu et al. 2007; Yang et al. 2018; Zhang et al. 2006]. Due to the use of a piecewise linear setting, the same issues as with the above vertex-to-mesh maps complicate the deduction of a homeomorphism. All methods performing non-rigid registration via spatial deformation eventually rely on some projection operation. Hence, they do not in general guarantee a continuous and injective map, which is one of our main objectives. Our method, by contrast, is fundamentally rooted in the 2-manifold nature of the problem and directly represents a continuous map.

2.3 Applications

Maps between surfaces are a key ingredient in a variety of application areas. We list prominent and recent examples in the following.

Shape Co-Processing. Surface maps are key to geometry processing methods that operate jointly or compatibly on multiple surfaces, whether collections of shapes or individual frames of an animation or deformation process. Examples are co-remeshing [Kraevoy and Sheffer 2004; Yang et al. 2018], co-quadrangulation [Meng and He 2016; Zhou et al. 2018], co-segmentation [Golovinskiy and Funkhouser 2009], co-analysis [Munsell et al. 2008], and co-field-synthesis [Azencot et al. 2017].

Information Transfer. The transfer of signals, attributes, or other data between surfaces naturally relies on maps between these. Examples are detail transfer [Biermann et al. 2002], texture transfer [Panozzo et al. 2013], deformation transfer [Sumner and Popović 2004], geometry transfer [Kraevoy and Sheffer 2005], mesh template transfer [Tierny et al. 2011], parametrization transfer (between levels of detail) [Bommes et al. 2013; Ebke et al. 2016; Sander et al. 2001].

Interpolation. For the purpose of shape interpolation or morphing, maps establishing the surface correspondence are an important component [Alexa 2002; Lee et al. 1999; Von-Tycowicz et al. 2015].

Embedding/Fitting. The embedding of some kind of graph structure, template, or base complex in surfaces is another use case requiring maps between manifolds. Instances are polycube mapping for texturing [Tarini et al. 2004] or meshing [Livesu et al. 2013], mesh layout embedding [Campen and Kobbelt 2014; Tarini et al. 2011], and model fitting [Allen et al. 2003; Blanz and Vetter 1999].

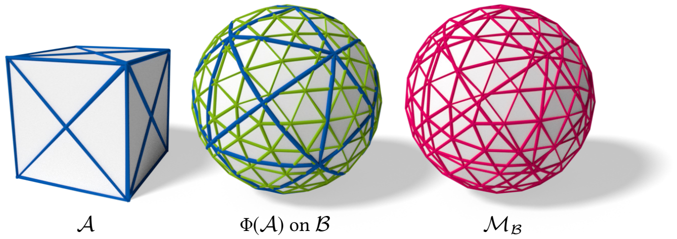

$\mathcal{A}$
$\Phi(\mathcal{A})$ on $\mathcal{B}$
$\mathcal{M}_\mathcal{B}$

Fig. 3. Left: input mesh $\mathcal{A}$. Center: the inter-surface map Φ embeds $\mathcal{A}$ on the surface of mesh $\mathcal{B}$. Right: the meta mesh $\mathcal{M}_\mathcal{B}$ is obtained by intersecting both mesh connectivities.

Note that continuity and injectivity or bijectivity of the map is either essential or beneficial for most of these applications. For instance, discontinuous or non-injective maps would lead to failures like invalid template embeddings, broken layout topology, torn textures, morphing artifacts, inconsistent or flipped parametrizations, and issues due to general non-invertibility and asymmetry.

3 BACKGROUND

We begin by reviewing a key object of interest—inter-surface maps—before we introduce our method in Sections 4 and 5.

3.1 Inter-Surface Maps

Consider a pair of 2-manifold surfaces $\mathcal{A}$ and $\mathcal{B}$ of the same genus. An *inter-surface* map [Schreiner et al. 2004] is a continuous bijective map $\Phi : \mathcal{A} \rightarrow \mathcal{B}$ between the points on both surfaces.

We are specifically interested in the discrete case in which $\mathcal{A}$ and $\mathcal{B}$ are triangle meshes. Applying Φ to the mesh $\mathcal{A}$ embeds its connectivity on the surface of mesh $\mathcal{B}$. Intuitively, each vertex of $\mathcal{A}$ is mapped to a point on $\mathcal{B}$ and each edge is mapped to a curve on the target surface. Intersecting the embedded connectivity $\Phi(\mathcal{A})$ with the mesh connectivity $\mathcal{B}$ forms the *meta mesh* $\mathcal{M}_\mathcal{B}$, embedded on $\mathcal{B}$ (cf. Figure 3). Its vertices consist of both original vertex sets as well as of an additional set of intersection vertices. Its edges are the edges of both $\Phi(\mathcal{A})$ and $\mathcal{B}$, split into segments at their intersections. Each face of $\mathcal{M}_\mathcal{B}$ corresponds to the overlap between a face of $\Phi(\mathcal{A})$ and a face of $\mathcal{B}$.

A symmetric construction—intersecting $\Phi^{-1}(\mathcal{B})$ with $\mathcal{A}$—yields $\mathcal{M}_\mathcal{A}$. Note that due to the bijectivity of Φ both $\mathcal{M}_\mathcal{A}$ and $\mathcal{M}_\mathcal{B}$ share the same connectivity and differ solely in their geometric embedding. We therefore write $\mathcal{M}$ when only the connectivity is of concern.

Adding the assumption that Φ is piecewise affine with respect to the faces of $\mathcal{M}$, leads to a number of beneficial properties:

- The edges of both $\mathcal{M}_\mathcal{A}$ and $\mathcal{M}_\mathcal{B}$ are straight line segments.
- The faces of $\mathcal{M}_\mathcal{A}$ and $\mathcal{M}_\mathcal{B}$ are polygonal cells.
- Each polygonal cell is planar, as it is a subset of the underlying target triangle.
- Only 3-, 4-, 5- and 6-gons occur, because each polygon is the 2D intersection between two triangles in the tangent space of the respective target triangle.
- For the same reason, each polygon is convex.
- Since the map is affine within each polygon, there is no need to triangulate $\mathcal{M}$.
- The map is fully specified by the vertex positions of $\mathcal{M}$ on $\mathcal{A}$ and $\mathcal{B}$.

The piecewise affine map Φ can be evaluated for a point $\mathbf{p}_\mathcal{A}$ on $\mathcal{A}$ using the following steps: (1) Look up the face of $\mathcal{M}_\mathcal{A}$ the point lies in, (2) express $\mathbf{p}_\mathcal{A}$ in a local 2D coordinate frame of this face, (3) apply the affine map, giving the mapped point $\Phi(\mathbf{p}_\mathcal{A})$ in the local 2D coordinate system of the corresponding face in $\mathcal{M}_\mathcal{B}$:

$$\Phi(\mathbf{p}_\mathcal{A}) = \mathbf{J} \cdot \mathbf{p}_\mathcal{A} + \mathbf{t}, \tag{1}$$

with a linear part $\mathbf{J} \in \mathbb{R}^{2\times 2}$ and a translation $\mathbf{t} \in \mathbb{R}^2$.

Maps via Planar Parametrizations. One way to locally obtain an inter-surface map is to overlay two injective parametrizations in a common planar domain. The map Φ is then defined as the composition of the first parametrization with the inverse of the second. If both parametrizations are piecewise affine, it follows that Φ is piecewise affine too and the above properties hold. The meta mesh $\mathcal{M}$ can be obtained by intersecting the mesh connectivities of $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^2$. This setting is particularly convenient because the edges of both $\mathcal{A}$ and $\mathcal{B}$ are straight in the parameter domain, i.e. the connectivity of $\mathcal{M}$ can be completely determined via line segment intersection tests.

3.2 Distortion Measures

A large number of well-known distortion measures from planar parametrization can be directly applied in the setting of inter-surface maps. For each affine piece of a map, orthonormal coordinate systems in both the source and target domain are set up. Given three independent points in the source coordinate system $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$, and their images $\mathbf{a}', \mathbf{b}', \mathbf{c}' \in \mathbb{R}^2$ in the target system, the linear part $\mathbf{J}$ of the map (Equation 1) is:

$$\mathbf{J} = \begin{bmatrix} \mathbf{b}' - \mathbf{a}' & \mathbf{c}' - \mathbf{a}' \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b} - \mathbf{a} & \mathbf{c} - \mathbf{a} \end{bmatrix}^{-1}. \tag{2}$$

The singular values $\sigma_{\min}$ and $\sigma_{\max}$ of $\mathbf{J}$ describe the minimum and maximum amount of local stretch and can be obtained via a closed-form solution [Smith and Schaefer 2015]. Isometric distortion measures penalize the deviation of both values from 1. Popular measures are the symmetric Dirichlet (SD) energy $\sigma_{\min}^2 + \sigma_{\max}^2 + \sigma_{\min}^{-2} + \sigma_{\max}^{-2} = \|\mathbf{J}\|_F^2 + \|\mathbf{J}^{-1}\|_F^2$ [Schreiner et al. 2004], the as-rigid-as-possible (ARAP) energy $(1 - \sigma_{\min})^2 + (1 - \sigma_{\max})^2$ [Liu et al. 2008] and a symmetrized version (SARAP) $(1 - \frac{1}{\sigma_{\min}})^2 + (1 - \sigma_{\max})^2$ [Poranne and Lipman 2016]. Due to the symmetry of the surface-to-surface setting, we are especially interested in energies satisfying $E(\Phi) = E(\Phi^{-1})$, such as the symmetric Dirichlet energy or the SARAP energy. An additional property of these two energies is that they are flip-preventing, i.e. they are natural barriers with respect to local non-injectivities in the map, diverging to ∞ as a piece of the map degenerates.

While conformal measures, e.g. the as-similar-as-possible (ASAP) energy $(\sigma_{\max} - \sigma_{\min})^2$, are equally supported, we focus on isometric measures because they are more sensitive towards misaligned geometric features [Schreiner et al. 2004]. In Figure 8 we demonstrate all four of the above energies with our method.

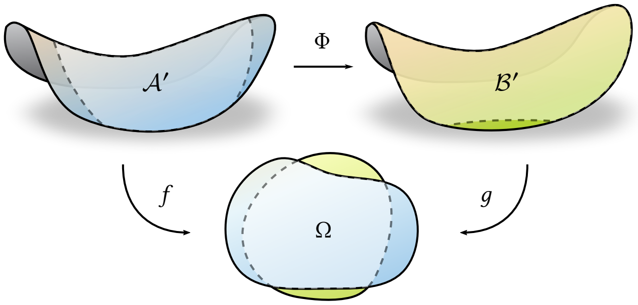


Fig. 4. We compute the inter-surface map Φ as the composition of a parametrization f with the inverse of a parametrization g. The distortion of the composed map Φ is agnostic to the intermediate domain Ω.

4 DISTORTION-MINIMIZING INTER-SURFACE MAPS

From here on, we consider as input two piecewise linear triangle meshes $\mathcal{A} = (\mathcal{V}^{\mathcal{A}}, \mathcal{E}^{\mathcal{A}}, \mathcal{T}^{\mathcal{A}})$ and $\mathcal{B} = (\mathcal{V}^{\mathcal{B}}, \mathcal{E}^{\mathcal{B}}, \mathcal{T}^{\mathcal{B}})$ of disk topology.

Overview. We construct a partial inter-surface map by overlaying parametrizations of $\mathcal{A}$ and $\mathcal{B}$ in a common planar domain. We then formulate an energy minimization problem in which the 2D images of all vertices of both $\mathcal{A}$ and $\mathcal{B}$ act as variables. Instead of considering distortion energies of the individual parametrizations, we directly optimize the end-to-end distortion of the map between the two input surfaces, effectively cancelling out the influence of the intermediate domain. Thus, the planar domain merely exists as an abstract space in which we express our variables.

4.1 Formulation

Let $f : \mathcal{A} \to \mathbb{R}^2$ and $g : \mathcal{B} \to \mathbb{R}^2$ be two globally injective, piecewise affine maps, defined at the vertices of the respective meshes. We refer to the intersection of their images $\Omega = f(\mathcal{A}) \cap g(\mathcal{B})$ as the *overlap* of $\mathcal{A}$ and $\mathcal{B}$ under f and g, respectively, and use f^{-1} and g^{-1} to denote the inverse maps, restricted to Ω. The inter-surface map Φ, defined as

$$\Phi = g^{-1} \circ f,$$

is a bijection between $\mathcal{A}' = f^{-1}(\Omega) \subseteq \mathcal{A}$ and $\mathcal{B}' = g^{-1}(\Omega) \subseteq \mathcal{B}$, as illustrated in Figure 4.

We measure the distortion of Φ via the symmetric Dirichlet energy, equally penalizing the L^2 stretch in both directions of the map. Each of the two stretch measures is integrated over the respective target surface [Schreiner et al. 2004]:

$$E_\Phi = \int_{\mathcal{B}'} \|\mathbf{J}_\Phi\|_F^2 \, \mathrm{d}A_{\mathcal{B}} + \int_{\mathcal{A}'} \|\mathbf{J}_\Phi^{-1}\|_F^2 \, \mathrm{d}A_{\mathcal{A}}, \tag{3}$$

with $\mathbf{J}_\Phi = \mathbf{J}_g^{-1} \cdot \mathbf{J}_f \in \mathbb{R}^{2\times 2}$ being the Jacobian of the map Φ. Note that the Dirichlet energy can easily be replaced by other measures based on $\mathbf{J}_\Phi$ or its singular values (cf. Section 3.2).

As $\mathcal{A}$ and $\mathcal{B}$ are triangle meshes, we can split the integral over an entire surface into integrals over its individual triangles:

$$E_\Phi = \sum_{t_{\mathcal{B}}} \int_{t_{\mathcal{B}}} \|\mathbf{J}_\Phi\|_F^2 \, \mathrm{d}A_{\mathcal{B}} + \sum_{t_{\mathcal{A}}} \int_{t_{\mathcal{A}}} \|\mathbf{J}_\Phi^{-1}\|_F^2 \, \mathrm{d}A_{\mathcal{A}},$$

where $t_{\mathcal{A}} \in \mathcal{A}'$ and $t_{\mathcal{B}} \in \mathcal{B}'$ are restricted to the parts of the triangles affected by the map. Since f is piecewise affine, its Jacobian $\mathbf{J}_f$ is piecewise constant in the triangulation of $\mathcal{A}$. Analogously, $\mathbf{J}_g$ is piecewise constant in $\mathcal{B}$.

As both triangulations differ from each other, the composed Jacobian $\mathbf{J}_\Phi = \mathbf{J}_g^{-1} \cdot \mathbf{J}_f$ is neither piecewise constant in $\mathcal{A}$ nor in $\mathcal{B}$, but in their mutual tessellation: the meta mesh $\mathcal{M}$, obtained by intersecting $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^2$ (cf. Section 3.1). Thus, the above triangle integrals can be split into sums over the polygonal faces of $\mathcal{M}$:

$$E_\Phi = \sum_{\rho \in \mathcal{M}} \|\mathbf{J}_\Phi\|_F^2 \cdot A_{\mathcal{B}}^{\rho} + \|\mathbf{J}_\Phi^{-1}\|_F^2 \cdot A_{\mathcal{A}}^{\rho}, \tag{4}$$

where $A_{\mathcal{A}}^{\rho}$ and $A_{\mathcal{B}}^{\rho}$ are the areas of the intersection polygon ρ on the respective surface. Note that Equation 4 is not a discrete approximation, but an exact realization of Equation 3 in the case of piecewise linear triangle meshes $\mathcal{A}$ and $\mathcal{B}$.

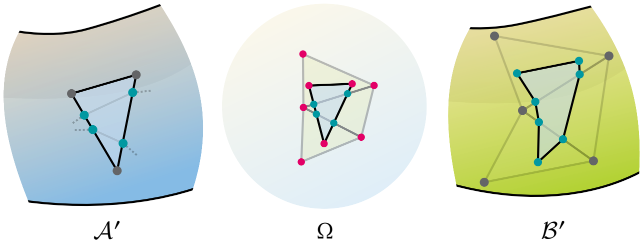


Fig. 5. A triangle of mesh $\mathcal{A}$ is mapped to the plane, where it is split according to the edges of mesh $\mathcal{B}$. The resulting pieces are lifted to the surface of $\mathcal{B}$. Variables in our system are the parameter images of vertices from $\mathcal{A}$ and $\mathcal{B}$ in Ω (magenta). Intersection vertices in Ω as well as images of vertices on the respective target surface are dependent (turquoise).

4.2 Choice of Optimization Variables

The intermediate flat domain allows a choice of optimization variables in one global 2D coordinate system. We choose the variable vector $\mathbf{x}$ as the concatenation of all vertex positions of $f(\mathcal{A})$ and $g(\mathcal{B})$ in $\mathbb{R}^2$:

$$\mathbf{x} = \left[\mathbf{p}_1^{\mathcal{A}}; \dots \mathbf{p}_{|\mathcal{V}^{\mathcal{A}}|}^{\mathcal{A}};\ \mathbf{p}_1^{\mathcal{B}}; \dots \mathbf{p}_{|\mathcal{V}^{\mathcal{B}}|}^{\mathcal{B}}\right] \in \mathbb{R}^n \tag{5}$$

is the column vector containing all parameter locations $\mathbf{p}^{\mathcal{A}}$ and $\mathbf{p}^{\mathcal{B}} \in \mathbb{R}^2$ and the number of variables is $n = 2|\mathcal{V}^{\mathcal{A}}| + 2|\mathcal{V}^{\mathcal{B}}|$.

In this setting, both discrete degrees of freedom (the connectivity of $\mathcal{M}$) as well as continuous ones (the affine transformation per polygon) of the inter-surface map Φ are parametrized by a single set of continuous variables (see Figure 5). In the following, we detail how the discrete energy $E_\Phi(\mathbf{x})$ in Equation 4 depends on the variables $\mathbf{x}$.

The Jacobian $\mathbf{J}_\Phi(\mathbf{x}) = \mathbf{J}_g(\mathbf{x})^{-1} \cdot \mathbf{J}_f(\mathbf{x})$ in a meta mesh polygon ρ depends on the individual Jacobians $\mathbf{J}_f(\mathbf{x})$ and $\mathbf{J}_g(\mathbf{x})$ of the corresponding triangles of $\mathcal{A}$ and $\mathcal{B}$. Both can be evaluated using Equation 2, where $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are constant and $\mathbf{a}', \mathbf{b}', \mathbf{c}'$ directly correspond to variable vertex positions in $\mathbb{R}^2$. While $\mathbf{J}_f(\mathbf{x})$ and $\mathbf{J}_g(\mathbf{x})$ are linear in $\mathbf{x}$, the Jacobian of the composed map $\mathbf{J}_\Phi(\mathbf{x}) = \mathbf{J}_g^{-1}(\mathbf{x}) \cdot \mathbf{J}_f(\mathbf{x})$ is not.

In contrast to the classical parametrization setting, the area weights $A_{\mathcal{A}}^{\rho}(\mathbf{x})$ and $A_{\mathcal{B}}^{\rho}(\mathbf{x})$ in Equation 4 also depend on $\mathbf{x}$. This is because

(1) the area of ρ in $\mathbb{R}^2$ changes with $\mathbf{x}$ and (2) the factor by which it is stretched out on $\mathcal{A}$ and $\mathcal{B}$ depends on $\mathbf{x}$ as well:

$$A^{\rho}_{\mathcal{A}}(\mathbf{x}) = \det \mathbf{J}_f^{-1}(\mathbf{x}) \cdot A^{\rho}_{\Omega}(\mathbf{x})$$
$$A^{\rho}_{\mathcal{B}}(\mathbf{x}) = \det \mathbf{J}_g^{-1}(\mathbf{x}) \cdot A^{\rho}_{\Omega}(\mathbf{x}).$$

While the Jacobians are again computed as above, the area of a k-gon ρ with vertices $\mathbf{p}_i$ in $\mathbb{R}^2$ is

$$A^{\rho}_{\Omega}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{k} \det \left[\mathbf{p}_i(\mathbf{x}) \;\; \mathbf{p}_{i+1}(\mathbf{x}) \right]$$

using $\mathbf{p}_{k+1} = \mathbf{p}_1$. Here, each $\mathbf{p}_i$ can either be a vertex of $\mathcal{A}$, a vertex of $\mathcal{B}$ or the straight line intersection between an edge of $\mathcal{A}$ and an edge of $\mathcal{B}$. In the first two cases, $\mathbf{p}_i$ directly corresponds to entries in $\mathbf{x}$. If $\mathbf{p}_i$ is an intersection vertex, its position is obtained from the closed-form solution of the 2×2 linear system

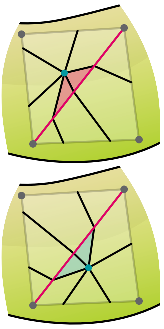
$$(1-\lambda)\mathbf{a}^{\mathcal{A}} + \lambda \mathbf{b}^{\mathcal{A}} = (1-\mu)\mathbf{c}^{\mathcal{B}} + \mu \mathbf{d}^{\mathcal{B}}, \tag{6}$$

in which $\mathbf{a}^{\mathcal{A}}(\mathbf{x}), \mathbf{b}^{\mathcal{A}}(\mathbf{x}), \mathbf{c}^{\mathcal{B}}(\mathbf{x}), \mathbf{d}^{\mathcal{B}}(\mathbf{x}) \in \mathbb{R}^2$ are vertex locations of $\mathcal{A}$ and $\mathcal{B}$ and as such directly entries of $\mathbf{x}$. The system always has a unique solution since, by construction of $\mathcal{M}$, intersection vertices only exist where two edge segments intersect in a single point.

4.3 Continuity of E_Φ

In a smooth setting, i.e. with $\mathcal{A}, \mathcal{B}$ being smooth surfaces and f, g smooth maps, the energy E_Φ is also a smooth function. The only discretization in our formulation is introduced by replacing $\mathcal{A}$ and $\mathcal{B}$ with piecewise linear surfaces. As long as the connectivity of $\mathcal{M}$ does not change, i.e. every mapped vertex stays in its current target triangle, the discrete E_Φ is still a smooth function in $\mathbf{x}$.

The meta mesh does change its connectivity when the image of a vertex $\Phi(v^{\mathcal{A}})$ moves across an edge $e^{\mathcal{B}}$, i.e switches from triangle $t_i^{\mathcal{B}}$ to $t_j^{\mathcal{B}}$ (see inset). In particular, a set $\mathcal{P}_{\text{old}}$ of meta mesh polygons, which are incident to both the vertex $\Phi(v^{\mathcal{A}})$ and the edge $e^{\mathcal{B}}$, disappears (red). After the vertex has moved across the edge, a new set $\mathcal{P}_{\text{new}}$ appears (blue). As $\mathcal{P}_{\text{old}}$ overlaps with $t_i^{\mathcal{B}}$ and $\mathcal{P}_{\text{new}}$ overlaps with $t_j^{\mathcal{B}}$, the pointwise distortion (in terms of the per-element Jacobians $\mathbf{J}_\Phi$) in $\mathcal{P}_{\text{new}}$ after the change can be quite different from the one in $\mathcal{P}_{\text{old}}$ before the change.



However, in such a transition the involved area terms in Equation 4 vanish while the Jacobians remain bounded (since the Jacobians of f and g do not depend on the meta mesh connectivity). As a result, $E_\Phi(\mathbf{x})$ is continuous across connectivity changes in the meta mesh. While the energy function itself is continuous, its derivatives are not, i.e. $E_\Phi(\mathbf{x})$ is C^0 but not C^1 continuous. Yet, under the assumption that the meshes $\mathcal{A}$ and $\mathcal{B}$ are approximations to smooth surfaces, we can expect E_Φ to be well-behaved as a piecewise approximation to a smooth energy functional.

4.4 Injectivity of Φ

We require the parametrizations f and g to be injective maps. As a consequence, both are invertible on their images in $\mathbb{R}^2$ and the composed map Φ, restricted to their overlap Ω, is a well-defined bijection. We maintain this property throughout the optimization by addressing local and global injectivity separately.

Local Injectivity. A parametrization f is locally injective if it is orientation preserving, i.e. $\det \mathbf{J}_f > 0$ for all triangles and the sum of angles around each interior vertex is 2π [Weber and Zorin 2014]. The first property is preserved by flip-preventing energies (cf. Section 3.2), which diverge to ∞ as a triangle degenerates. We make use of this barrier characteristic by adding the symmetric Dirichlet energies $E_f(\mathbf{x})$ and $E_g(\mathbf{x})$ of the individual parametrizations f and g with a small weight to our optimization. In addition, we check if $f(\mathbf{x})$ or $g(\mathbf{x})$ violate the angle sum criterion and return ∞ in that case. (While there is no continuous feasible path to such a configuration, a discrete step might reach it in rare cases.)

Note that since f and g are guaranteed to be locally injective, the composed map Φ is locally injective too, without making further assumptions about the properties of E_Φ. Therefore, non-flip-preventing energies, such as ARAP, can also be used as the main distortion measure.

Besides guaranteeing local injectivity, the individual barrier energies $E_f(\mathbf{x})$ and $E_g(\mathbf{x})$ also have an important regularizing effect: As Φ is restricted by Ω, there can be variable vertices in $\mathbf{x}$ that are outside of Ω and thus not affected by $E_\Phi(\mathbf{x})$, resulting in an underdetermined system. However, it is important that such vertices stay part of the system, as they should be able to freely move in and out of the overlap during the optimization. In other words, we do not wish to constrain the boundaries $f^{-1}(\partial\Omega)$ and $g^{-1}(\partial\Omega)$ of the map (unless desired). The regularization ensures a consistent movement of vertices in $\mathbb{R}^2 \setminus \Omega$.

Global Injectivity. In addition, f is globally injective if it is bijective on its boundary [Lipman 2014], i.e. the boundary of $f(\mathcal{A})$ does not intersect itself. If either $f(\mathcal{A})$ or $g(\mathcal{B})$ threatens to violate global injectivity, the method of [Jiang et al. 2017] can be used, which reduces this problem to the one of local injectivity: A scaffold mesh is created by filling the empty space between the boundary of the parametrization and a larger bounding box using a constrained Delaunay triangulation. Again, using the symmetric Dirichlet energy with a low weight prevents the scaffold triangles from degenerating. As a result, the parametrization boundary cannot overlap itself.

4.5 Position Constraints

Our method allows adding hard positional constraints in a straightforward manner. Intuitively, a vertex of one mesh can be constrained to remain at its current position on the other surface as in using a fixing pin.

Consider w.l.o.g. a vertex $v^{\mathcal{A}}$ of $\mathcal{A}$ being pinned to its location on $\mathcal{B}$. Although this fixes $\Phi(v^{\mathcal{A}})$, the position $f(v^{\mathcal{A}})$ in $\mathbb{R}^2$ is still free to move around, as long as the corresponding point of $\mathcal{B}$ performs the same movement under g. We therefore constrain $\mathbf{p}^{\mathcal{A}} := f(v^{\mathcal{A}})$ via its barycentric coordinates in the triangle of $g(\mathcal{B})$:

$$\mathbf{p}^{\mathcal{A}} = \alpha \mathbf{a}^{\mathcal{B}} + \beta \mathbf{b}^{\mathcal{B}} + \gamma \mathbf{c}^{\mathcal{B}},$$

with $\alpha+\beta+\gamma = 1$, where $\mathbf{a}^{\mathcal{B}}, \mathbf{b}^{\mathcal{B}}, \mathbf{c}^{\mathcal{B}} \in \mathbb{R}^2$ are the parameter locations of the triangle vertices. As $\mathbf{p}^{\mathcal{A}}, \mathbf{a}^{\mathcal{B}}, \mathbf{b}^{\mathcal{B}}, \mathbf{c}^{\mathcal{B}} \in \mathbb{R}^2$ are variables of the system and the barycentric coordinates $0 < \alpha, \beta, \gamma < 1$ are

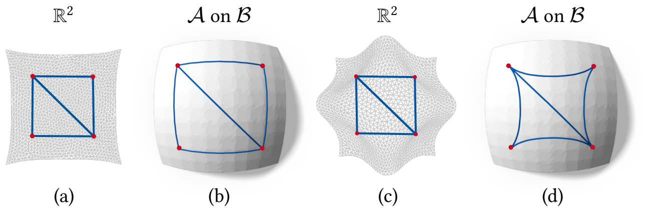

$\mathbb{R}^2$
$\mathcal{A}$ on $\mathcal{B}$
$\mathbb{R}^2$
$\mathcal{A}$ on $\mathcal{B}$
(a)
(b)
(c)
(d)

Fig. 6. We show two different maps of a single quad $\mathcal{A}$ to a fine mesh $\mathcal{B}$. While all edges are straight in the parameter domain (a, c), their embedding on the target surface can be curved (b, d). Here, the edges of $\mathcal{A}$ (blue) are bent by deforming the parametrization of $\mathcal{B}$ (gray).

constants, this results in a linear equality constraint. We satisfy such constraints by variable elimination. Thus, in the following we refer by $\mathbf{x}$ to the reduced variable vector and by $E_\Phi(\mathbf{x})$ to the objective function in which all occurrences of pinned vertices $\mathbf{p}^{\mathcal{A}}$ have been replaced by the above linear term.

4.6 Degrees of Freedom

Again, suppose a setting with smooth surfaces $\mathcal{A}, \mathcal{B}$ and smooth maps f, g into the plane. In this setting, the composed map $\Phi = g^{-1} \circ f$ is invariant to any bijective deformation $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ applied to both parametrizations, as it cancels out in the composition $(\psi \circ g)^{-1} \circ (\psi \circ f) = g^{-1} \circ f$. It would therefore be sufficient to only optimize one of the two maps into the plane, keeping the other one fixed. This does not limit the degrees of freedom, as for any desired deformation of $g(\mathcal{B})$ the inverse can be applied to $f(\mathcal{A})$, resulting in the same composition Φ.

In our discrete setting however, the space of deformations the map Φ is invariant to is smaller. Since the positions of the meta mesh vertices are defined by straight line intersections, any deformation of $\mathbb{R}^2$ that does not preserve ratios, i.e. that is not affine, can change Φ. The other way around, (non-degenerate) affine transformations both preserve the relative location of intersection vertices and cancel out in the per-element composition of both maps. Consequently, Φ is invariant exactly to the non-singular affine deformations applied to both parametrizations $f(\mathcal{A})$ and $g(\mathcal{B})$.

Thus, in contrast to the continuous setting, there is a large space of mutual deformations in $\mathbb{R}^2$, which *do* represent degrees of freedom for the map Φ. Hence, it is important to include the vertices of both meshes in the optimization. Figure 6 shows how the embedding of an edge of $\mathcal{A}$ on $\mathcal{B}$ can be altered by deforming the underlying parametrization of $\mathcal{B}$.

Removing Redundant Degrees of Freedom. The only remaining degrees of freedom leading to an underconstrained system in our setting are the global affine transformations applied to both $f(\mathcal{A})$ and $g(\mathcal{B})$ simultaneously. The distortion terms E_f and E_g of the individual parametrizations (Section 4.4) already regularize any global stretching. However, they are invariant to rigid transformations—the only remaining class of transformations that needs to be fixed. We eliminate the redundant translations by fixing the most central vertex of $g(\mathcal{B})$ to the origin of $\mathbb{R}^2$. Similarly, the rotation is removed by fixing a boundary vertex to one of the coordinate axes. Both constraints are realized via variable elimination.

5 OPTIMIZATION

In the above formulation, we wish to directly minimize the distortion of the inter-surface map Φ, subject to injectivity constraints as well as positional hard constraints. While the former are realized via barrier terms (Section 4.4), the latter are simple variable eliminations (Section 4.5), leaving us with the following unconstrained minimization problem:

$$\min_{\mathbf{x}} \quad E_\Phi(\mathbf{x}) \; + \; \omega_{\mathcal{R}} \left(E_f(\mathbf{x}) + E_g(\mathbf{x}) \right).$$

Here, $E_\Phi(\mathbf{x})$ is the integrated distortion energy of the inter-surface map (Equation 4) and $E_f(\mathbf{x}) + E_g(\mathbf{x})$ are the flip-preventing and regularizing distortion energies of the individual parametrizations (Section 4.4), weighted by a small positive factor $\omega_{\mathcal{R}}$ (10^{-8} in all examples). Lastly, $\mathbf{x} \in \mathbb{R}^n$ is the continuous variable vector (Equation 5). In the following, we refer to the above sum as our objective function $E(\mathbf{x})$.

We employ a second-order optimization algorithm which follows the basic scheme of Newton's method: In each iteration we consider a second-order approximation of E around the current point $\mathbf{x}$, given in terms of the gradient $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^n$ and Hessian matrix $\mathbf{H}(\mathbf{x}) \in \mathbb{R}^{n \times n}$. We compute the direction vector $\mathbf{d}(\mathbf{x}) \in \mathbb{R}^n$, pointing to the minimum of the quadratic approximation and take a step $s\mathbf{d}$, with $s > 0$, such that the new point $\mathbf{x} + s\mathbf{d}$ has a lower energy.

In the following, we tailor the optimization method to the properties of our objective $E(\mathbf{x})$. Specifically, $E(\mathbf{x})$ is a non-convex function whose dependency on $\mathbf{x}$ is significantly more involved than in the classical parametrization setting. In addition, it violates a precondition of Newton's method by only possessing C^0 continuity. We show that despite both properties, we can effectively minimize $E(\mathbf{x})$ with large step sizes. The resulting algorithm is summarized again in Appendix A.

5.1 Non-Convexity

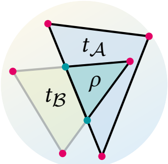
The discrete inter-surface distortion $E_\Phi(\mathbf{x})$ includes three main operators: (1) distortion measure, (2) map composition and (3) edge-edge intersection.

1. Typical isometric distortion measures, e.g. the symmetric Dirichlet, ARAP or SARAP, are generally non-convex [Golla et al. 2018].
2. The distortion measure itself is a function of the per-polygon Jacobians $\mathbf{J}_\Phi$ of the composed map. While the individual per-triangle Jacobians $\mathbf{J}_f$ and $\mathbf{J}_g$ are linear in the vertex parameter locations, the entries of $\mathbf{J}_\Phi = \mathbf{J}_g^{-1} \cdot \mathbf{J}_f$ are non-convex functions, due to the 2×2 matrix inverse.
3. Furthermore, the area weights in $E_\Phi(\mathbf{x})$ require computing intersection vertices between variable line segments by solving a 2×2 linear system (Equation 6). Its closed-form solution has a non-convex dependency on the variables $\mathbf{x}$.

As a result, we can expect E_Φ to be non-convex, i.e. its Hessian to have negative eigenvalues. One might fear that the non-convexity is more severe than in the classical parametrization setting, rendering the optimization less efficient. In our experiments however, we observe negative eigenvalues of the Hessian of E_Φ with similar or even lower magnitude than in E_f and E_g.

Nonetheless, Newton's method requires a positive definite Hessian matrix to produce a descent direction [Boyd and Vandenberghe 2004]. Replacing $\mathbf{H}$ with any positive-definite matrix $\mathbf{H}_+$ does result in a descent direction, as $\mathbf{g}^\mathsf{T}\mathbf{H}_+\mathbf{g} > 0$. While computing the eigendecomposition of $\mathbf{H}$ and clamping the negative eigenvalues to a small positive number gives the closest positive definite matrix to $\mathbf{H}$ in the Frobenius norm, this method is impractical for large matrices [Chen and Weber 2017]. Instead, we use the approach of [Teran et al. 2005], who compute the Hessian matrix for each surface element, apply the projection via eigendecomposition to each one, and obtain $\mathbf{H}_+$ as a sum of positive-definite matrices. We clamp negative eigenvalues of the per-element Hessians to $\varepsilon = 10^{-3}$ in all our examples.

Assembly of the Hessian Matrix. In case of $E_\Phi(\mathbf{x})$, this means computing the per-element Hessian $\mathbf{H}_\rho$ for each polygon ρ of the meta mesh. Recall that ρ is the intersection of two triangles $t^\mathcal{A}$ and $t^\mathcal{B}$ in the parameter domain. The integrated energy in this polygon depends on $\mathbf{x}$ via the triangle Jacobians $\mathbf{J}_f$ and $\mathbf{J}_g$, as well as through its area A^ρ (cf. Section 4.1). The Jacobians are determined by the vertices in $\mathbb{R}^2$ of $t^\mathcal{A}$ and $t^\mathcal{B}$ respectively, i.e. by a total of 12 variables. The intersection area A^ρ in the parameter domain depends on the same set of variables, as each vertex of ρ is either a vertex of $t^\mathcal{A}$, a vertex of $t^\mathcal{B}$ or an intersection between edges of $t^\mathcal{A}$ and $t^\mathcal{B}$. Thus, disregarding variable elimination, each $\mathbf{H}_\rho$ is a dense symmetric 12×12 matrix.



In case of a pin constraint, a vertex is replaced by a combination of three other vertices, resulting in a slightly larger matrix. If a vertex is constrained to a fixed 2D location, the matrix is smaller.

The additional energies $E_f(\mathbf{x})$ and $E_g(\mathbf{x})$ consider both meshes independently and thus result in simpler 6×6 Hessians per triangle.

After projecting all per-element matrices to positive-definite matrices, we inflate each one to size $n \times n$ and compute their sum $\mathbf{H}_+$. The full Hessian matrix is symmetric and follows a global block structure in which two diagonal blocks encode dependencies within a mesh and the off-diagonal block encodes dependencies between the two meshes. The sparsity pattern in the off-diagonal block largely depends on the tessellation of the input meshes $\mathcal{A}$ and $\mathcal{B}$. In case of e.g. two similarly uniform tessellations, the off-diagonal block is relatively sparse, since each triangle intersects only a small number of triangles in the other mesh. If however $\mathcal{A}$ is a coarse layout and $\mathcal{B}$ a high-resolution surface, the off-diagonal block has few but comparably dense rows, as one triangle of $\mathcal{A}$ covers many triangles of $\mathcal{B}$. This density is desirable, as it directly encodes dependencies between a large number of vertices of $\mathcal{B}$ in the quadratic approximation, allowing a more coordinated movement of those vertices

With the positive definite matrix $\mathbf{H}_+$ in place, it is now possible to obtain a descent direction $\mathbf{d}$ by solving the linear system $\mathbf{H}_+\mathbf{d} = -\mathbf{g}$.

5.2 Meta Mesh Combinatorics and Line Search

Using a flat intermediate domain allows us to express the combinatorial degrees of freedom (which vertex maps to which triangle, which edge of $\mathcal{A}$ intersects which edge of $\mathcal{B}$) solely in terms of a continuous variable vector. This comes at the cost of having to infer the discrete degrees of freedom from the continuous ones.

Specifically, for each point $\mathbf{x}$ we wish to evaluate, we have to compute the meta mesh $\mathcal{M}$, by overlaying the connectivites of $f(\mathcal{A})$ and $g(\mathcal{B})$ in the parameter domain and computing their mutual tessellation. We decide to recompute $\mathcal{M}$ at each $\mathbf{x}$ instead of sequentially tracking its changes during optimization because we target large step sizes, which would incur a long series of connectivity changes to be performed explicitly per iteration. Further, we wish to employ a backtracking line search that starts by testing large steps and decreases the step size if necessary. Lastly, recomputing the meta mesh at different $\mathbf{x}$ independently is trivially parallelizable.

Given a current point $\mathbf{x}$ inside the feasible region (corresponding to injective maps f and g) and a descent direction $\mathbf{d}$, we first compute the maximum feasible step size $s_{\max}$ following [Smith and Schaefer 2015]. This is the step size for which the first triangle of the parametrizations $f(\mathcal{A})$ or $g(\mathcal{B})$ degenerates. Starting from $s = 0.99 \cdot s_{\max}$, we successively decrease s by a factor of 0.8. In addition, we always check $s = 1$ if $1 < s_{\max}$, as it marks the minimum of our quadratic approximation and indeed often yields the best improvement in practice. We parallelize the backtracking line search in blocks of 8 step sizes. For each step size s in the current block, we compute $\mathcal{M}(\mathbf{x}+s\mathbf{d})$ and evaluate $E(\mathbf{x}+s\mathbf{d})$ in parallel. If one or more step sizes fulfill the Armijo condition (cf. [Nocedal and Wright 2006]), we pick the one yielding the largest improvement and stop the procedure. If no step size in the current block fulfills the condition, we proceed to the next block.

One implication of the varying meta mesh connectivity is that the sparsity pattern of the Hessian matrix $\mathbf{H}$ changes between iterations. Therefore, we cannot compute a symbolic factorization of $\mathbf{H}$ as it is usually possible in the classical parametrization setting.

5.3 Handling C^1 Discontinuities

As discussed in Section 4.3, $E_\Phi(\mathbf{x})$ is C^0 but not C^1 whenever a vertex v moves from one underlying triangle into another. At such an event both $\mathbf{g}$ and $\mathbf{H}$ can jump and, as a consequence, the search direction $\mathbf{d}$ also changes abruptly. A problematic situation arises when the search direction $\mathbf{d}_v \in \mathbb{R}^2$ of the vertex v repeatedly turns by close to 180° and v oscillates around an underlying edge. In addition, it is possible that the direction component $\mathbf{d}_v$ of v is of large magnitude, because a potentially strong increase in E_Φ across the discontinuity is not reflected in local information. In this case, the line search has to globally reduce the step size s, allowing a single vertex to slow down the optimization immensely.

A simple, yet effective way to mitigate the problem lies in employing a momentum-based approach. In particular, we apply temporal smoothing directly to the discontinuous quantities $\mathbf{g}$ and $\mathbf{H}_+$, by blending their current values with accumulated ones by a factor of 0.5 in each iteration. As the sparsity pattern of $\mathbf{H}_+$ changes between iterations, we prevent the accumulated matrix from becoming too dense by considering only a limited history (of size 6 in all our examples).

While smoothing $\mathbf{H}_+$ preserves positive-definiteness, smoothing $\mathbf{g}$ can lead to a non-descent direction $\mathbf{d}$ when solving $\mathbf{H}_+\mathbf{d} = -\mathbf{g}$. In that case (which we observe only very rarely), we reset the gradient and Hessian history and re-solve for $\mathbf{d}$ using just the most recent data, thereby guaranteeing a descent direction.

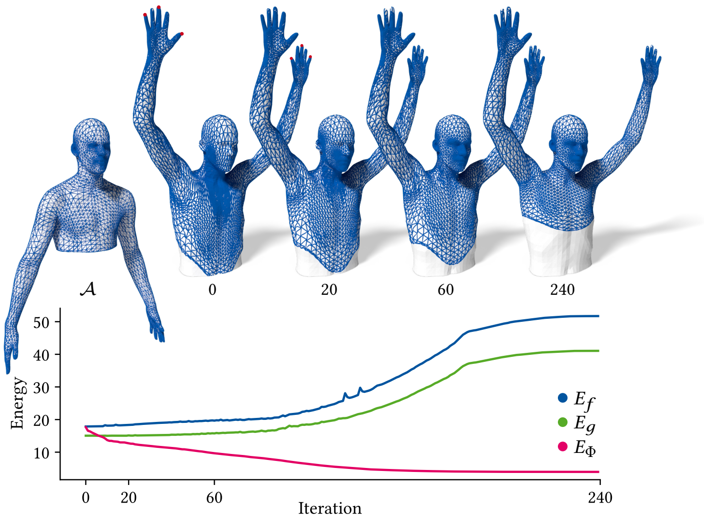


Fig. 7. We map between two different poses of a human model. Landmarks on 6 fingers were used for initialization only. In the initial map, the head of $\mathcal{A}$ is located on the chest of $\mathcal{B}$. Our algorithm successfully recovers from this situation and converges to the desired minimum in 240 iterations. While the end-to-end energy E_Φ is minimized, the individual parametrization energies E_f and E_g increase.

Intuitively, the smoothed $\mathbf{g}$ and $\mathbf{H}_+$ lead to a smoothed $\mathbf{d}$ in which sudden changes of per-vertex directions have the chance to cancel out. Most importantly, when a vertex jumps across an underlying edge, the quadratic approximation of E_Φ improves as information from both sides of the C^1 discontinuity is collected. In that sense, the local approximation of E_Φ, based on a smoothed $\mathbf{g}$ and $\mathbf{H}_+$, more closely mimics that of a function which is in itself smooth. We demonstrate the effect of this modification in Section 6.

5.4 Preconditioning

While the optimization algorithm described so far is able to effectively minimize $E(\mathbf{x})$, it can take many iterations to do so, especially when vertices have to travel long distances on the target surface. Since the overall energy is very sensitive to the distortion of single triangles, fine-level adjustments can easily dominate the system. This prohibits consistent large-scale movements of entire mesh regions in few iterations. We therefore wish to bias the optimization towards performing low-frequency updates (e.g. large scale alignment of similarly curved regions) first and deferring high-frequency updates (e.g. small scale fine tuning of individual elements) to later iterations.

We make use of a squared Laplacian preconditioner that comes with an intuitive interpretation. Specifically, we modify the second-order Taylor approximation of E around $\mathbf{x}$,

$$\hat{E}(\mathbf{x}+\mathbf{d}) = E(\mathbf{x}) + \mathbf{g}(\mathbf{x})^\mathsf{T}\mathbf{d} + \frac{1}{2}\mathbf{d}^\mathsf{T}\mathbf{H}_+(\mathbf{x})\mathbf{d},$$

by adding the additional term

$$\frac{1}{2}\omega_\mathcal{L}\mathbf{d}^\mathsf{T}\mathbf{L}^\mathsf{T}\mathbf{M}\mathbf{L}\mathbf{d}.$$

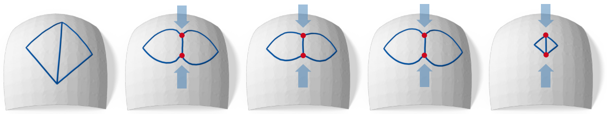


Fig. 8. A single quad is mapped to a finely tessellated surface (left). The embedded edges are polygonal lines whose resolution is determined by the underlying mesh. We pinch two vertices of the quad and show how different energies compensate the constraint.

Here, $\mathbf{L}$ is the combined Laplace-Beltrami operator of both meshes, i.e. the $n \times n$ matrix having the cotangent-weighted Laplace-Beltrami operators of $\mathcal{A}$ and $\mathcal{B}$ as diagonal blocks. Similarly, $\mathbf{M}$ is the lumped mass matrix of both meshes, i.e. the diagonal matrix containing the Voronoi areas of all vertices in $\mathcal{A}$ and $\mathcal{B}$. The modification is realized by replacing the Hessian matrix with $\mathbf{H}_+(\mathbf{x}) + \omega_\mathcal{L}\mathbf{L}^\mathsf{T}\mathbf{M}\mathbf{L}$, which is again positive definite.

We can interpret the update step $\mathbf{d}$ as a vector field defined at the vertices of $f(\mathcal{A})$ and $g(\mathcal{B})$ in $\mathbb{R}^2$. Then $\mathbf{L}\mathbf{d}$ is the vector field describing for each vertex the deviation in $\mathbf{d}$ from its neighborhood. Thus, the inner product $\langle \mathbf{L}\mathbf{d}, \mathbf{L}\mathbf{d}\rangle_{\mathcal{A}\mathcal{B}} = \mathbf{d}^\mathsf{T}\mathbf{L}^\mathsf{T}\mathbf{M}\mathbf{L}\mathbf{d}$ is a quadratic measure for the smoothness of $\mathbf{d}$. By adding this measure to $\hat{E}$, we penalize the deviation of $\mathbf{d}$ from being harmonic.

Note that we do not modify the objective E, but merely its quadratic approximation $\hat{E}$, resulting in a different optimization path.

The strength of the modification is determined by the weight $\omega_\mathcal{L}$, which we control by a simple adaptive strategy, mimicking a coarse-to-fine behavior. Starting from a high initial value (10^9 in all examples), we try the current, a smaller and a larger value in each iteration. From $\frac{1}{\alpha}\omega_\mathcal{L}$, $\omega_\mathcal{L}$ and $\alpha\omega_\mathcal{L}$ ($\alpha = 1.25$ in most examples), we choose the value yielding the best improvement in E and update $\omega_\mathcal{L}$ accordingly. In our experiments, this preconditioner improves the convergence rate of our optimization significantly, as shown in Section 6.

6 RESULTS

Implementation. We implemented the algorithm using OpenMesh [Botsch et al. 2002] and Eigen3 [Guennebaud et al. 2010]. To robustly obtain the meta mesh, we compute the 2D mesh overlay of $\mathcal{A}$ and $\mathcal{B}$ using exact predicates [Shewchuk 1996]. Per-element gradients and Hessians of $E(\mathbf{x})$ are computed using automatic differentiation with ADOL-C [Walther and Griewank 2012]. Further, we use the SLIM implementation in libIGL [Jacobson et al. 2018] for initialization.

Initialization. In our experiments, we start from bijective Tutte embeddings of $\mathcal{A}$ and $\mathcal{B}$, and initially minimize E_f and E_g separately using SLIM [Rabinovich et al. 2017]. We use penalty terms [Lévy 2001] to align landmark vertices of $\mathcal{A}$ to the corresponding vertices of $\mathcal{B}$ in $\mathbb{R}^2$. From thereon, our method maintains point-to-point constraints exactly as they are provided by the initialization.

Other forms of initialization, e.g. previous injective mapping methods via the plane, which do not optimize distortion end-to-end (Section 2.1), could be used as well. If required by the application, exact satisfaction of landmark constraints can be achieved by using [Kraevoy et al. 2003] for initialization. This method adds Steiner vertices if necessary to guarantee a feasible parametrization of mesh $\mathcal{A}$.

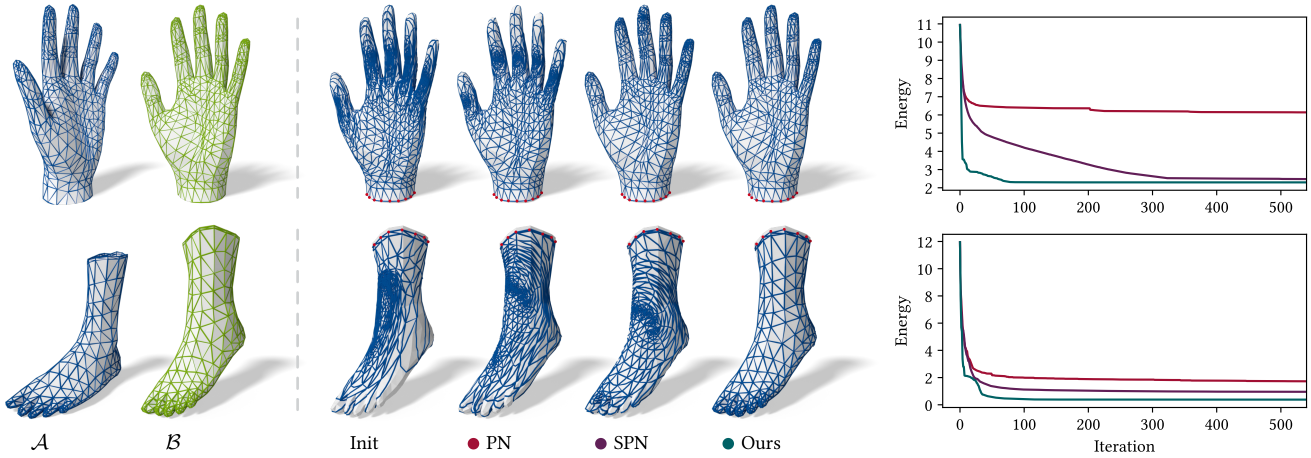


Fig. 9. Left: input meshes $\mathcal{A}$ and $\mathcal{B}$ of similar geometry. Center: embeddings of $\mathcal{A}$ on $\mathcal{B}$ are optimized, starting from challenging initial solutions. For each variant of the algorithm, we show the resulting maps after 550 iterations (if not converged earlier). Our formulation using a projected Newton method (PN), as described up to Section 5.2, initially removes some fine-level distortion, but then takes prohibitively small steps on the C^1 discontinuous objective function. Smoothing the quadratic approximation of E_Φ (Section 5.3, SPN) performs the expected movement in one example (hand) but slows down prematurely in the other (foot). Our full method, including the Laplacian preconditioner (Section 5.4), reaches the desired minimum in less than 100 (hand) and 150 (foot) iterations.

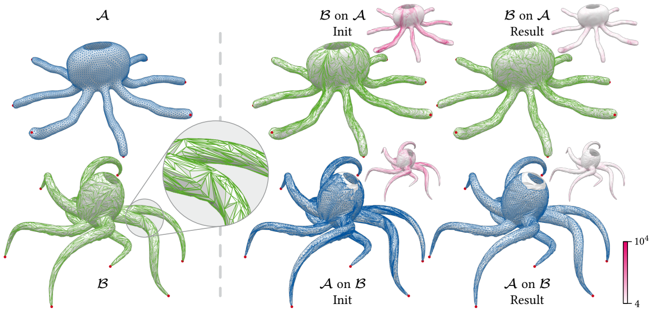

$\mathcal{A}$
$\mathcal{B}$ on $\mathcal{A}$
Init
$\mathcal{B}$ on $\mathcal{A}$
Result
$\mathcal{B}$
$\mathcal{A}$ on $\mathcal{B}$
Init
$\mathcal{A}$ on $\mathcal{B}$
Result
10^4
4

Fig. 10. We map between meshes of different triangulation quality. While $\mathcal{A}$ is uniformly tessellated, we reduced the triangulation quality of $\mathcal{B}$ by a series of edge flips (43% of triangles have an inner angle of less than 10°). Our method effectively optimizes the initially distorted map and converges within 400 iterations. A logarithmic heat map shows the symmetric Dirichlet energy per polygon before and after the optimization.

Comparison to Surface-to-Plane Optimization. In Figure 1, we demonstrate that optimizing two surface-to-plane maps can be quite different from optimizing their composition. Specifically, we optimize E_f and E_g under landmark constraints, as described in Section 4.5, and compare to our results. Figure 7 plots the energies E_Φ, E_f and E_g for a different example and shows that in fact E_f and E_g increase when optimizing for low distortion in the end-to-end map.

Distortion Measures. While we use the symmetric Dirichlet energy in our examples unless stated otherwise, our formulation supports a variety of other distortion measures (cf. Section 3.2). In Figure 8, we demonstrate the behavior of different measures in the presence of external constraints. In contrast to the classical parametrization setting, embedded edges can bend to the extent permitted by the resolution of the target domain.

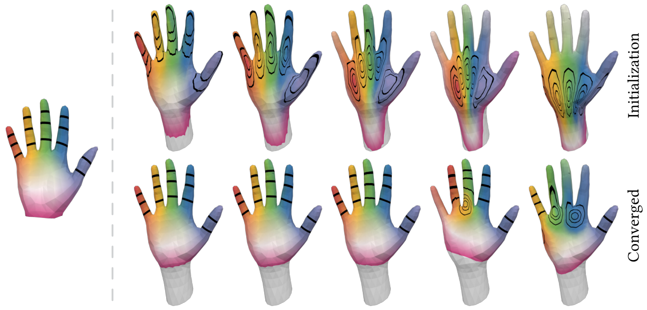


Fig. 11. We compute maps between two hands, starting from increasingly challenging initial maps (top row). Our optimization (bottom row) recovers the expected map as long as the initialization is reasonably close.

Dependence on Initialization. Solving a non-convex optimization problem with a potentially large number of local minima naturally bears the risk of converging to an undesired minimum. Such minima arise in particular when similarly curved regions align locally—producing low distortion—while the geometry is still misaligned on a more global scale. In our experiments, we observe that a small number of landmark constraints is enough to guide our algorithm to the correct minimum (see Figure 13). Depending on the application, we maintain these constraints during the optimization (Figure 1, Figure 12) or use them for initialization only (Figure 7, Figure 11, Figure 15).

Meta Mesh Complexity. We observed that in all our examples the number of polygons in $\mathcal{M}$ is bounded by a factor of 10 compared to the sum of triangles in $\mathcal{A}$ and $\mathcal{B}$; even if the input meshes show considerable differences in tessellation quality. A similar observation has already been reported by [Praun et al. 2001]. In Figure 10 we successfully optimize a map between a uniform and a highly anisotropic triangulation.

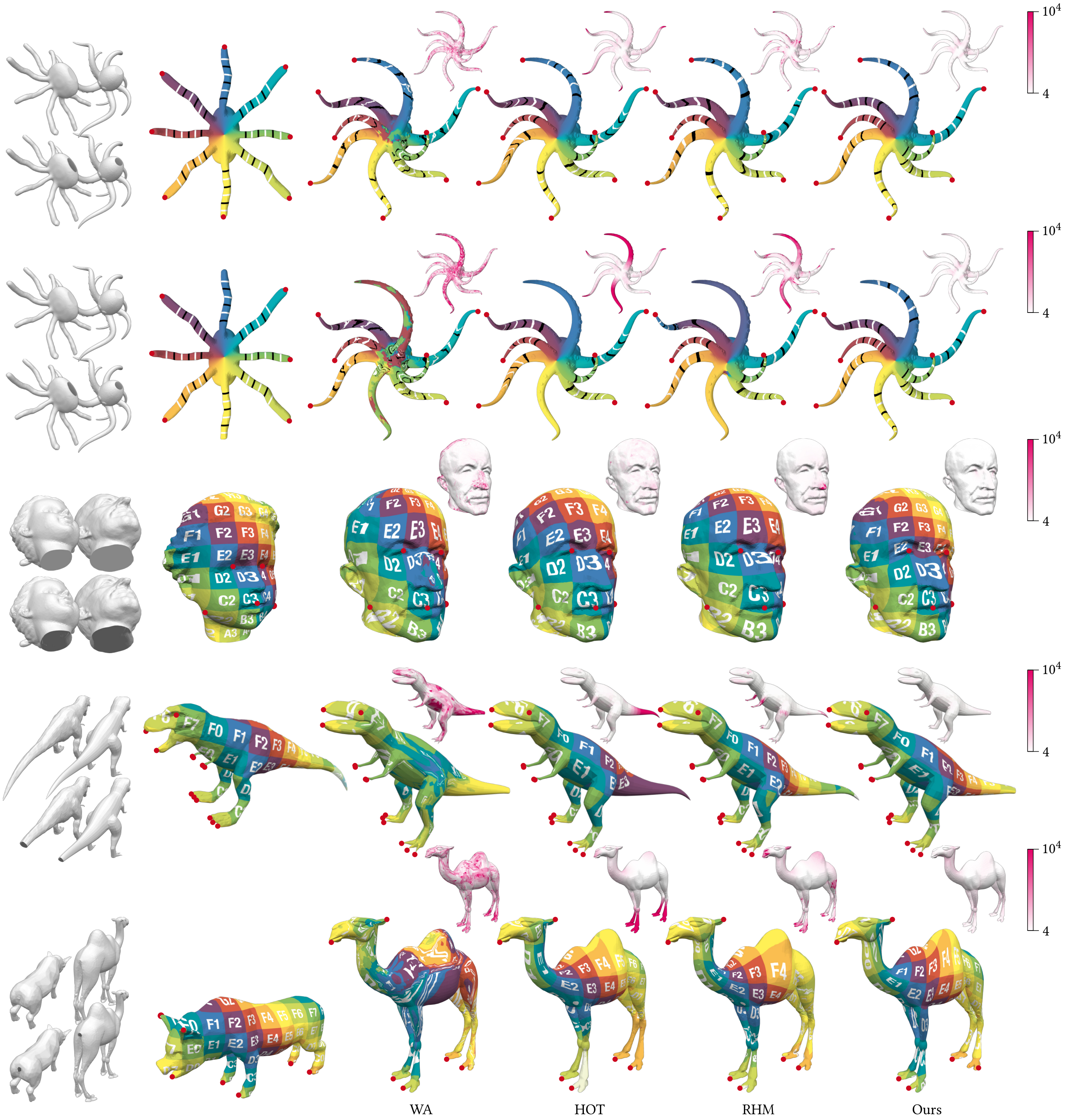


Fig. 12. Comparison to related methods on differently challenging inputs. While we run HOT and RHM on closed surfaces, we add a hole to each input mesh for WA and our method. Landmark constraints are shown in red and kept fixed during the optimization. Logarithmic heat maps show the symmetric Dirichlet energy per polygon. Due to the low number of landmarks, WA creates a consistent map only in the vicinity of constraint vertices or in flat regions. HOT always produces a continuous bijective map, but has no incentive to align similarly curved regions away from landmarks (see arms of the octopus and tail of the dino). RHM does optimize for map distortion, but suffers from projection artifacts and thus does not produce injective maps (note how the color singularity on the octopus in the second row is mapped twice). While our method only operates on disk-topology meshes, it always produces an injective map, distributes the inevitable distortion evenly and aligns features far away from landmarks (see e.g. the arms of the octopus and the correctly mapped top of the camel's hump).

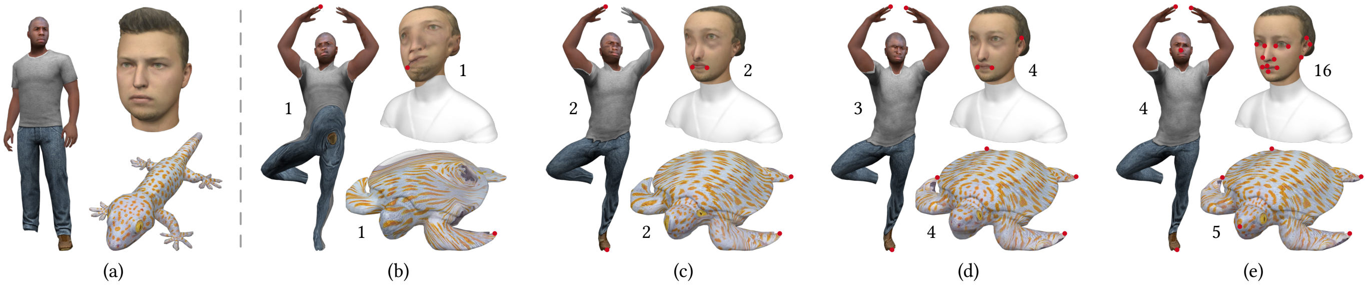


Fig. 13. Maps between source shapes (a) and target shapes (b-e) with successively more landmark constraints (red). We show the converged results of our optimization, with numbers next to models indicating the amount of landmarks used. While insufficient constraints can lead to undesired local optima (b, c), a low number of landmarks is usually enough find the intended distortion minimum (d). Additional landmarks can be added to fine-tune the map in presence of geometric differences between the shapes (e).

Meta Mesh Geometry. As polygons of the meta mesh are intersections between pairs of input triangles, their shape does become arbitrarily bad and often near-degenerate, even for high-quality input meshes. Still, we can determine the meta mesh connectivity robustly using exact predicates. Further, note that our objective function depends on the shape of meta mesh polygons only in terms of their area (cf. Section 4.2). No map Jacobians are computed using the geometry of meta mesh polygons. As a result, numerical inaccuracies due to near-degenerate polygons only have a minuscule effect on the overall objective function; we never encountered negative effects in our experiments.

6.1 Modified Newton Method

Our optimization algorithm modifies Newton's method. Projected Newton (PN) makes use of an element-wise positive definite Hessian projection (Section 5.1) and allows optimizing a non-convex function. We refer to the method that, in addition, uses a smoothed quadratic approximation (Section 5.3) as Smoothed Projected Newton (SPN). Finally, adding the Laplacian preconditioner (Section 5.4) yields our method. We compare the behavior of these variants in Figure 9.

Per-Iteration Cost. The algorithm requires fewer than 400 iterations in all our examples. On our largest mesh pair (Pig–Camel, Figure 12, 26,000 and 19,000 input triangles), the algorithm converged in 169 iterations, with a per-iteration run time of 38 seconds. This is mainly due to our current implementation using algorithmic differentiation, which enables flexible prototyping. With 30 seconds per iteration, this step dominates the process and we expect significant speedups using other forms of differentiation. On the same example, solving the linear system takes 5 seconds and 3 seconds are spent in the line search. This contains computing the multiple instances of the meta mesh in parallel, each of which takes about 0.3 seconds using exact predicates.

6.2 Comparison to Other Mapping Approaches

We compare our method to *Weighted Averages on Surfaces* (WA) [Panozzo et al. 2013], *Hyperbolic Orbifold Tutte Embeddings* (HOT) [Aigerman and Lipman 2015] and *Reversible Harmonic Maps* (RHM) [Ezuz et al. 2019b], for which implementations provided by the authors are available. While HOT and RHM operate on closed surfaces, our algorithm requires disk topology meshes and WA can handle both settings. To enable a fair comparison in terms of map quality, we cut a small hole in each input mesh for our method and WA. We then visualize the resulting maps and their distortion in areas away from the boundary (see Figure 12).

WA expresses a point as a weighted average of the surrounding landmark vertices and synthesizes a corresponding point on the other surface. While the approach is flexible in its application, it is not well suited for scenarios with complex shapes with few landmarks, as the resulting maps exhibit high distortion and a large number of discontinuities.

HOT maps both meshes to a hyperbolic orbifold, while introducing cuts and cone singularities at the landmark vertices. Both maps are optimized separately and the composed map is guaranteed to be a continuous bijection. In contrast to our method (which composes high-distortion maps f and g into the plane to a low-distortion map Φ), HOT chooses an intermediate domain that allows for low-distortion maps f and g in the first place. Due to the absence of an end-to-end distortion measure, however, the method is in general oblivious to misalignment of similarly curved regions.

RHM maps vertices of both meshes to the embedding space of the respective target mesh. It optimizes for both vertex positions and their projections to the target surface. Bijectivity of the resulting map is favored via a quadratic term but cannot be enforced. In fact, the resulting maps show various discontinuities.

Our method produces a bijection between the corresponding parts of both meshes by construction. As a result of the end-to-end distortion optimization, matching regions align automatically, even far away from landmarks.

6.3 Application: Template Embedding

The method allows embedding different types of template meshes in a target surface. Depending on the scenario, the template may be a finely tessellated mesh (e.g. for compatible remeshing) or a coarse (quadrilateral, polygonal) complex or layout (e.g. for morphable models, polycube mapping). In Figure 14 we embed a coarse artist-generated quad layout on a finely tessellated surface. In contrast to many other methods, our embedding does not rely on a projection operator, i.e. does not risk invalidating layout topology. Further, given a rough initial alignment, the method does not inherently require any type of descriptor or correspondence information but can operate purely based on the distortion of the embedding.

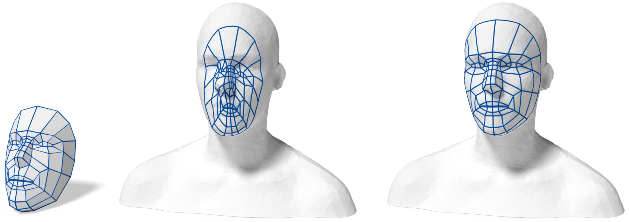


Fig. 14. We use our algorithm to embed a quad template (left) on a target surface. From a rough initial alignment (center), our method finds a minimum-distortion embedding (right). We obtain a continuous and injective embedding of the entire template, including its edges and faces. (Note that, as the template mesh has quadrilateral elements, it is virtually triangulated for the optimization process.)

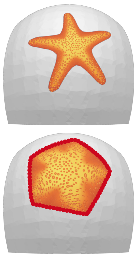
6.4 Application: Texture Transfer

Our maps can transfer various attribute fields (continuous or not) between surfaces. In Figures 13 and 15, we demonstrate texture transfer between differently shaped meshes. While the maps in Figure 15 require significant stretch, the inevitable distortion is evenly distributed over the stretched areas and landmark correspondences are only needed for initialization. Since our maps are continuous, they will never tear a texture or introduce additional seams. As the maps are injective, textures cannot fold over.

Especially due to those two properties, our method is also well suited to transfer surface parametrizations, e.g. integer-grid-maps in the context of quadrangulation, between meshes.

7 LIMITATIONS & FUTURE WORK

The most obvious limitation of our current formulation is the focus on disk-topology surfaces. It will naturally be interesting to investigate options for generalization to closed and higher-genus surfaces. For instance, the use of locally injective global parametrizations with cuts and cones for the intermediate maps could be an alternative. A challenge in the context of end-to-end optimization will be transitions across cuts that need to be taken into account. In contrast to switching to non-planar intermediate domains (like the sphere, the hyperbolic plane, etc., cf. Sec. 2.1) this would allow preserving the Euclidean, piecewise linear setting, typically enabling simpler optimization setups.



A main feature of our method is its free-boundary setting. If one, however, wishes to constrain the coverage region, boundary vertices can be constrained to fixed locations (e.g. as in Figure 9). The inset shows another example of free boundary (top, the starfish assumes its natural shape during optimization) versus fixed boundary (bottom). Note that this, however, not only predetermines the region to be covered but also the map along its boundary curve. Depending on the use case, it may be interesting to let this boundary curve parametrization be part of the optimization (i.e. enable sliding of boundary vertices along the prescribed curve).

Of course, we are dealing with a non-linear optimization problem. The result is thus a local optimum. Combining our approach with some form of multi-scale technique in order to achieve better optima would be an interesting orthogonal endeavor. A related direction is that of integration of multi-resolution / multi-grid schemes for the purpose of speed-up.

In the context of surface-to-plane mapping, a number of techniques were recently introduced that go beyond classical Newton-style optimization. Using domain-specific or measure-specific modifications, significant improvements in terms of convergence behavior can be achieved [Claici et al. 2017; Kovalsky et al. 2016; Liu et al. 2018; Rabinovich et al. 2017; Shtengel et al. 2017]. As these are not trivially applicable in our setting (with elementary objective terms formulated based on the polygonal meta mesh), investigating the potential of the underlying ideas in the inter-surface context is a highly interesting challenge for future work.

A bijective map between surfaces expresses a one-to-one correspondence between them. Natural questions are whether and how inter-surface map optimization can be used in the context of the shape correspondence / registration problem. Isometry-related objectives, as used here, incentivize the intrinsic alignment in terms of Gaussian curvature distribution only. To obtain semantically meaningful results, further measures (e.g. based on extrinsic information and higher-level descriptors) would have to be taken into account. Investigating the potential of such augmentations is another interesting avenue.

ACKNOWLEDGMENTS

We thank the authors of the AIM@SHAPE Project and the FAUST, TOSCA and SHREC shape datasets. Additional meshes are courtesy of Marc Pauly (octopus), subatasos on TurboSquid (T-Rex), Nark Russom (giraffe), ptaxa999 on 3DExport.com (goat), Keenan Crane (cow), the Digital Life Project (gecko, turtle), Tucho Fernández Calo (man) and Ruslan Vasylev (head). We further thank Jan Möbius [2010] for maintaining the OpenFlipper framework.

This work was supported by the European Research Council under the European Union's Seventh Framework Programme, ERC grant agreement 340884 (ACROSS), by the Excellence Initiative of the German federal and state governments (CompSE), as well as by the Gottfried-Wilhelm Leibniz Programme of the Deutsche Forschungsgemeinschaft (DFG). Financial support from the DFG through grant IRTG-2379 is gratefully acknowledged.

REFERENCES

Noam Aigerman, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Spherical Orbifold Tutte Embeddings. *ACM Transactions on Graphics* 36, 4 (2017).

Noam Aigerman and Yaron Lipman. 2015. Orbifold Tutte Embeddings. *ACM Transactions on Graphics* 34, 6 (2015).

Noam Aigerman and Yaron Lipman. 2016. Hyperbolic Orbifold Tutte Embeddings. *ACM Transactions on Graphics* 35, 6 (2016).

Noam Aigerman, Roi Poranne, and Yaron Lipman. 2014. Lifted Bijections for Low Distortion Surface Mappings. *ACM Transactions on Graphics* 33, 4 (2014).

Noam Aigerman, Roi Poranne, and Yaron Lipman. 2015. Seamless Surface Mappings. *ACM Transactions on Graphics* 34, 4 (2015).

Marc Alexa. 2000. Merging Polyhedral Shapes with Scattered Features. *The Visual Computer* 16, 1 (2000).

Marc Alexa. 2002. Recent Advances in Mesh Morphing. In *Computer Graphics Forum*, Vol. 21.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. In *ACM Transactions on Graphics*, Vol. 22.

Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2005. Consistent Spherical Parameterization. In *International Conference on Computational Science*. Springer.

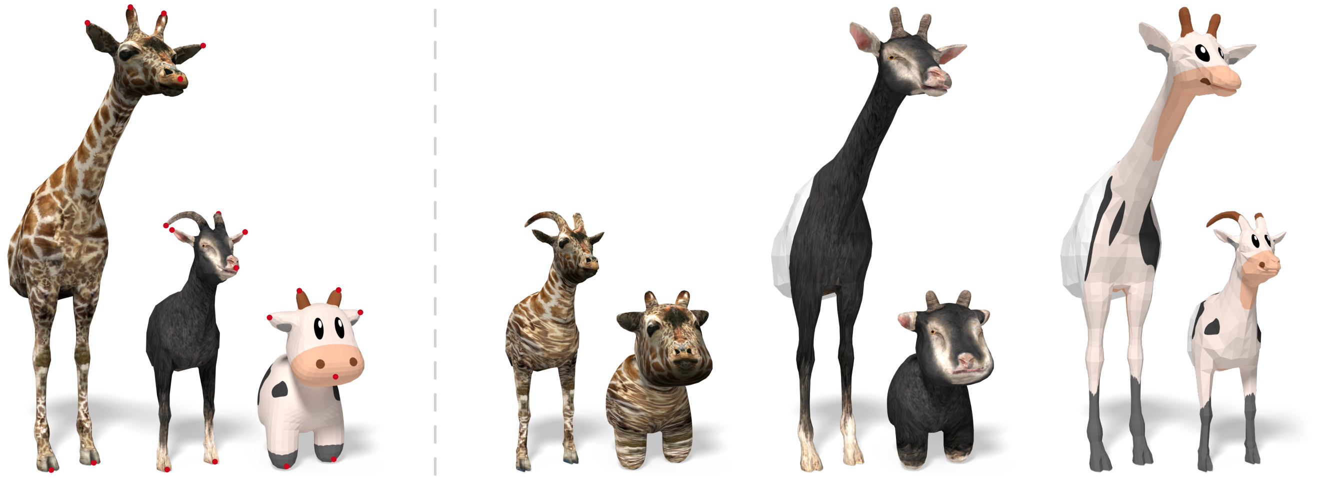


Fig. 15. Continuous maps between three disk topology surfaces are used for mutual texture transfer. The maps were computed by our method using landmark correspondences only for initialization (indicated by red dots on the left). By construction, the textures never invert or degenerate anywhere. The partially uncovered giraffe surfaces on the right are due to the use of an injective map; a bijective one can be obtained as well by constraining the boundary.

Omri Azencot, Etienne Corman, Mirela Ben-Chen, and Maks Ovsjanikov. 2017. Consistent Functional Cross Field Design for Mesh Quadrangulation. *ACM Transactions on Graphics* 36, 4 (2017).

Alex Baden, Keenan Crane, and Misha Kazhdan. 2018. Möbius Registration. *Computer Graphics Forum* (2018).

Henning Biermann, Ioana Martin, Fausto Bernardini, and Denis Zorin. 2002. Cut-and-Paste Editing of Multiresolution Surfaces. *ACM Transactions on Graphics* 21, 3 (2002).

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of SIGGRAPH '99*. ACM Press / Addison-Wesley.

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013. Integer-Grid Maps for Reliable Quad Meshing. *ACM Transactions on Graphics* 32, 4 (2013).

Mario Botsch, Stefan Steinberg, Stefan Bischoff, and Leif Kobbelt. 2002. OpenMesh: A Generic and Efficient Polygon Mesh Data Structure. *OpenSG Symposium* (2002).

Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. 2013. Sparse Iterative Closest Point. In *Symposium on Geometry Processing*. Eurographics Association.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York.

Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual Loops Meshing: Quality Quad Layouts on Manifolds. *ACM Transactions on Graphics* 31, 4 (2012).

Marcel Campen and Leif Kobbelt. 2014. Quad Layout Embedding via Aligned Parameterization. *Computer Graphics Forum* 33, 8 (2014).

Renjie Chen and Ofir Weber. 2017. GPU-Accelerated Locally Injective Shape Deformation. *ACM Transactions on Graphics* 36, 6 (2017).

Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. 2017. Isometry-Aware Preconditioning for Mesh Parameterization. In *Computer Graphics Forum*, Vol. 36.

Hans-Christian Ebke, Patrick Schmidt, Marcel Campen, and Leif Kobbelt. 2016. Interactively Controlled Quad Remeshing of High Resolution 3D Models. *ACM Transactions on Graphics* 35, 6 (2016).

Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. 2019a. Elastic Correspondence between Triangle Meshes. *Computer Graphics Forum* 38, 2 (2019).

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019b. Reversible Harmonic Maps between Discrete Surfaces. *ACM Transactions on Graphics* 38, 2 (2019).

Björn Golla, Hans-Peter Seidel, and Renjie Chen. 2018. Piecewise Linear Mapping Optimization Based on the Complex View. In *Computer Graphics Forum*, Vol. 37.

Aleksey Golovinskiy and Thomas Funkhouser. 2009. Consistent Segmentation of 3D Models. *Computers & Graphics* 33, 3 (2009).

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Igor Guskov, Kiril Vidimce, Wim Sweldens, and Peter Schröder. 2000. Normal Meshes. In *Proceedings of SIGGRAPH '00*.

Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. 2008. Non-Rigid Registration Under Isometric Deformations. In *Symposium on Geometry Processing*.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A Simple C++ Geometry Processing Library. http://libigl.github.io/libigl/.

Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial Complex Augmentation Framework for Bijective Maps. *ACM Transactions on Graphics* 36, 6 (2017).

Takashi Kanai, Hiromasa Suzuki, and Fumihiko Kimura. 1997. 3D Geometric Metamorphosis Based on Harmonic Map. In *Proceedings of Pacific Graphics '97*. IEEE.

Andrei Khodakovsky, Nathan Litke, and Peter Schröder. 2003. Globally Smooth Parameterizations with Low Distortion. *ACM Transactions on Graphics* 22, 3 (2003).

Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended Intrinsic Maps. *ACM Transactions on Graphics* 30, 4.

Shahar Z. Kovalsky, Meirav Galun, and Yaron Lipman. 2016. Accelerated Quadratic Proxy for Geometric Optimization. *ACM Transactions on Graphics* 35, 4 (2016).

Vladislav Kraevoy and Alla Sheffer. 2004. Cross-Parameterization and Compatible Remeshing of 3D Models. *ACM Transactions on Graphics* 23, 3 (2004).

Vladislav Kraevoy and Alla Sheffer. 2005. Template-Based Mesh Completion. In *Symposium on Geometry Processing*, Vol. 385.

Vladislav Kraevoy, Alla Sheffer, and Craig Gotsman. 2003. Matchmaker: Constructing Constrained Texture Maps. *ACM Transactions on Graphics* 22, 3 (2003).

Aaron W. F. Lee, David Dobkin, Wim Sweldens, and Peter Schröder. 1999. Multiresolution Mesh Morphing. In *Proceedings of SIGGRAPH '99*.

Bruno Lévy. 2001. Constrained Texture Mapping for Polygonal Meshes. In *Proceedings of SIGGRAPH '01*. ACM.

Hao Li, Robert W Sumner, and Mark Pauly. 2008b. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library.

Xin Li, Yunfan Bao, Xiaohu Guo, Miao Jin, Xianfeng Gu, and Hong Qin. 2008a. Globally Optimal Surface Mapping for Surfaces with Arbitrary Topology. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008).

Yaron Lipman. 2014. Bijective Mappings of Meshes with Boundary and the Degree in Mesh Processing. *SIAM Journal on Imaging Sciences* 7, 2 (2014).

Yaron Lipman and Thomas Funkhouser. 2009. Möbius Voting for Surface Correspondence. *ACM Transactions on Graphics* 28, 3 (2009).

Nathan Litke, Marc Droske, Martin Rumpf, and Peter Schröder. 2005. An Image Processing Approach to Surface Matching. In *Symposium on Geometry Processing*, Vol. 255.

Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive Parameterizations. *ACM Transactions on Graphics* 37, 4 (2018).

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local / Global Approach to Mesh Parameterization. In *Symposium on Geometry Processing*.

Marco Livesu, Nicholas Vining, Alla Sheffer, James Gregson, and Riccardo Scateni. 2013. PolyCut: Monotone Graph-Cuts for PolyCube Base-Complex Construction. *ACM Transactions on Graphics* 32, 6 (2013).

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-minimizing Transport Plans for Inter-Surface Mapping. *ACM Transactions on Graphics* 36, 4 (2017).

Min Meng and Ying He. 2016. Consistent Quadrangulation for Shape Collections via Feature Line Co-extraction. *Computer-Aided Design* 70, C (2016).

Jan Möbius and Leif Kobbelt. 2010. OpenFlipper: An Open Source Geometry Processing and Rendering Framework. In *International Conference on Curves and Surfaces*.

Brent C. Munsell, Pahal Dalal, and Song Wang. 2008. Evaluating Shape Correspondence for Statistical Shape Analysis: A Benchmark Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 11 (2008).

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (2nd ed. ed.). Springer, New York.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics* 31, 4 (2012).

Daniele Panozzo, Ilya Baran, Olga Diamanti, and Olga Sorkine-Hornung. 2013. Weighted Averages on Surfaces. *ACM Transactions on Graphics* 32, 4 (2013).

Nico Pietroni, Marco Tarini, and Paolo Cignoni. 2010. Almost Isometric Mesh Parameterization through Abstract Domains. *IEEE Transactions on Visualization and Computer Graphics* 16, 4 (2010).

Roi Poranne and Yaron Lipman. 2016. *Simple Approximations of Planar Deformation Operators.* Technical Report.

Emil Praun, Wim Sweldens, and Peter Schröder. 2001. Consistent Mesh Parameterizations. In *Proceedings of SIGGRAPH '01.*

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Transactions on Graphics* 36, 4 (2017).

Emanuele Rodolà, Michael Möller, and Daniel Cremers. 2015. Point-wise Map Recovery and Refinement from Functional Correspondence.

Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. 2001. Texture Mapping Progressive Meshes. In *Proceedings of SIGGRAPH '01.* ACM.

John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2004. Inter-Surface Mapping. *ACM Transactions on Graphics* 23, 3 (2004).

Andrei Sharf, Marina Blumenkrants, Ariel Shamir, and Daniel Cohen-Or. 2006. SnapPaste: An Interactive Technique for Easy Mesh Composition. *The Visual Computer* 22, 9 (2006).

Jonathan Richard Shewchuk. 1996. Robust Adaptive Floating-Point Geometric Predicates. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry.* Association for Computing Machinery.

Rui Shi, Wei Zeng, Zhengyu Su, Jian Jiang, Hanna Damasio, Zhonglin Lu, Yalin Wang, Shing-Tung Yau, and Xianfeng Gu. 2017. Hyperbolic Harmonic Mapping for Surface Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 5 (2017).

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Geometric Optimization via Composite Majorization. *ACM Transactions on Graphics* 36, 4 (2017).

Jason Smith and Scott Schaefer. 2015. Bijective Parameterization with Free Boundaries. *ACM Transactions on Graphics* 34, 4 (2015).

Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas. 2012. Soft Maps Between Surfaces. In *Computer Graphics Forum*, Vol. 31.

Robert W. Sumner and Jovan Popović. 2004. Deformation Transfer for Triangle Meshes. *ACM Transactions on Graphics* 23, 3 (2004).

Gary K. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank Langbein, Yonghuai Liu, A. David Marshall, Ralph Martin, Xianfang Sun, and Paul Rosin. 2013. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (2013).

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. Polycube-Maps. In *ACM Transactions on Graphics*, Vol. 23.

Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. 2011. Simple Quad Domains for Field Aligned Mesh Parametrization. *ACM Transactions on Graphics* 30, 6 (2011).

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *SCA '05.*

Julien Tierny, Joel Daniels, Luis G. Nonato, Valerio Pascucci, and Claudio T. Silva. 2011. Inspired Quadrangulation. *Computer-Aided Design* 43, 11 (2011).

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A Survey on Shape Correspondence. In *Computer Graphics Forum*, Vol. 30.

Christoph Von-Tycowicz, Christian Schulz, Hans-Peter Seidel, and Klaus Hildebrandt. 2015. Real-Time Nonlinear Shape Interpolation. *ACM Transactions on Graphics* 34, 3 (2015).

Andrea Walther and Andreas Griewank. 2012. Getting Started with ADOL-C. In *Combinatorial Scientific Computing*, Uwe Naumann and Olaf Schenk (Eds.). CRC Press, Taylor & Francis Group.

Ofir Weber and Denis Zorin. 2014. Locally Injective Parametrization with Arbitrary Fixed Boundaries. *ACM Transactions on Graphics* 33, 4 (2014).

Huai-Yu Wu, Chunhong Pan, Qing Yang, and Songde Ma. 2007. Consistent Correspondence Between Arbitrary Manifold Surfaces. In *ICCV 2007.* IEEE.

Yang Yang, Xiao-Ming Fu, Shuangming Chai, Shiwei Xiao, and Ligang Liu. 2018. Volume-Enhanced Compatible Remeshing of 3D Models. *IEEE Transactions on Visualization and Computer Graphics* (2018).

Lei Zhang, Ligang Liu, Zhongping Ji, and Guojin Wang. 2006. Manifold Parameterization. In *Advances in Computer Graphics.* Springer.

Jiaran Zhou, Marcel Campen, Denis Zorin, Changhe Tu, and Claudio T. Silva. 2018. Quadrangulation of Non-Rigid Objects Using Deformation Metrics. *Computer-Aided Geometric Design* 62 (2018).

A PSEUDOCODE

We supply a comprehensive summary of our method from an implementation perspective in Algorithm 1.

In presence of position constraints (Section 4.5), vertex positions in Line 6 are substituted with the respective linear combination. In that case, the dimensions of the per-element gradients $\mathbf{g}_\Phi^\rho, \mathbf{g}_f^{t^\mathcal{A}}, \mathbf{g}_g^{t^\mathcal{B}}$ and Hessians $\mathbf{H}_\Phi^\rho, \mathbf{H}_f^{t^\mathcal{A}}, \mathbf{H}_g^{t^\mathcal{B}}$ can differ. Further, note that $\mathbf{g}^{(i)}$ and $\mathbf{H}^{(i)}$ refer to the global gradient and Hessian in iteration i, which are implicitly omitted in sums if $i < 0$.

ALGORITHM 1: Inter-Surface Map Optimization

```
   Input: $\mathbf{x}^{(0)}$ (2D vertex coordinates of $\mathcal{A}$ and $\mathcal{B}$ in the plane)
1  Compute the combined Laplacian $\mathbf{L} \in \mathbb{R}^{n \times n}$ and
     mass matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ of $\mathcal{A}$ and $\mathcal{B}$ (Sec. 5.4).
2  for iteration $i = 0 \ldots i_{\max}$ do
3      Compute meta mesh polygons $\mathcal{M}(\mathbf{x}^{(i)})$.
4      foreach polygon $\rho = t^\mathcal{A} \cap t^\mathcal{B}$ in $\mathcal{M}(\mathbf{x}^{(i)})$ do
5          Compute $E_\Phi^\rho, {E_f}^{t^\mathcal{A}}, {E_g}^{t^\mathcal{B}}$ per element (Eq. 4 & Sec. 4.4).
6          Compute (using auto-differentiation) the gradients and
             Hessians w.r.t. the 6 vertices of $t^\mathcal{A}$ and $t^\mathcal{B}$, namely:
                 $\mathbf{g}_\Phi^\rho \in \mathbb{R}^{12}$ and $\mathbf{g}_f^{t^\mathcal{A}}, \mathbf{g}_g^{t^\mathcal{B}} \in \mathbb{R}^6$
                 $\mathbf{H}_\Phi^\rho \in \mathbb{R}^{12 \times 12}$ and $\mathbf{H}_f^{t^\mathcal{A}}, \mathbf{H}_g^{t^\mathcal{B}} \in \mathbb{R}^{6 \times 6}$.
7          Project $\mathbf{H}_\Phi^\rho, \mathbf{H}_f^{t^\mathcal{A}}$ and $\mathbf{H}_g^{t^\mathcal{B}}$ to s.p.d. matrices via eigen
             decomposition, i.e. $\mathbf{H}_{\bullet +}^\bullet := \mathbf{V}\mathbf{D}_+\mathbf{V}^\top$ (Sec. 5.1).
8      Assemble the global gradient $\mathbf{g}^{(i)} \in \mathbb{R}^n$ and sparse Hessian
         matrix $\mathbf{H}^{(i)} \in \mathbb{R}^{n \times n}$ (Sec. 5.1).
9      $\mathbf{g} := \sum_{j=1}^{6} 0.5^j \cdot \mathbf{g}^{(i-j+1)}$ and $\mathbf{H} := \sum_{j=1}^{6} 0.5^j \cdot \mathbf{H}^{(i-j+1)}$ (Sec 5.3)
10     foreach $\ell \in \{-1, 0, 1\}$ do
11         $\omega_\ell := \alpha^\ell \cdot \omega^{(i-1)}$ (multiple choice, Sec. 5.4)
12         Solve $(\mathbf{H} + \omega_\ell \mathbf{L}^\top \mathbf{M} \mathbf{L})\mathbf{d} = -\mathbf{g}$ (Sec. 5.4).
13         if $\mathbf{d}^\top \mathbf{g}^{(i)} > 0$ then
14             Discard all $\mathbf{g}^{(<i)}, \mathbf{H}^{(<i)}$ and goto line 9 (reset history).
15         Compute max. step size $s_{\max}$ [Smith and Schaefer 2015].
16         $s := 0.99 \cdot s_{\max}$
17         for $k = 0 \ldots$ do
18             $\mathbf{x}_{k,\ell} := \mathbf{x}^{(i)} + s\mathbf{d}$ (line search)
19             Compute meta mesh polygons $\mathcal{M}(\mathbf{x}_{k,\ell})$.
20             Evaluate $E(\mathbf{x}_{k,\ell})$.
21             Break if $E(\mathbf{x}_{k,\ell}) \leq E(\mathbf{x}^{(i)}) + 10^{-4} \cdot s\mathbf{d}^T\mathbf{g}$ (Armijo rule).
22             $s := 0.8 \cdot s$
23     $k^*, \ell^* := \mathrm{argmin}_{k,\ell}\, E(\mathbf{x}_{k,\ell})$
24     $\omega^{(i)} := \omega_{\ell^*}$
25     $\mathbf{x}^{(i+1)} := \mathbf{x}_{k^*,\ell^*}$
```