Symmetric Volume Maps: Order-invariant Volumetric Mesh Correspondence with Free Boundary

S. MAZDAK ABULNAGA, Massachusetts Institute of Technology, USA
ODED STEIN, Massachusetts Institute of Technology, USA and University of Southern California, USA
POLINA GOLLAND and JUSTIN SOLOMON, Massachusetts Institute of Technology, USA

Although shape correspondence is a central problem in geometry processing, most methods for this task apply only to two-dimensional surfaces. The neglected task of *volumetric* correspondence—a natural extension relevant to shapes extracted from simulation, medical imaging, and volume rendering—presents unique challenges that do not appear in the two-dimensional case. In this work, we propose a method for mapping between volumes represented as tetrahedral meshes. Our formulation minimizes a distortion energy designed to extract maps symmetrically, i.e., without dependence on the ordering of the source and target domains. We accompany our method with theoretical discussion describing the consequences of this symmetry assumption, leading us to select a symmetrized ARAP energy that favors isometric correspondences. Our final formulation optimizes for near-isometry while matching the boundary. We demonstrate our method on a diverse geometric dataset, producing low-distortion matchings that align closely to the boundary.

CCS Concepts: • **Computing methodologies** → **Volumetric models**; **Shape analysis;**

Additional Key Words and Phrases: Correspondence, volumes, tetrahedral meshes, as-rigid-as-possible, symmetry

ACM Reference format:
S. Mazdak Abulnaga, Oded Stein, Polina Golland, and Justin Solomon. 2023. Symmetric Volume Maps: Order-invariant Volumetric Mesh Correspondence with Free Boundary. *ACM Trans. Graph.* 42, 3, Article 25 (April 2023), 20 pages.
https://doi.org/10.1145/3572897

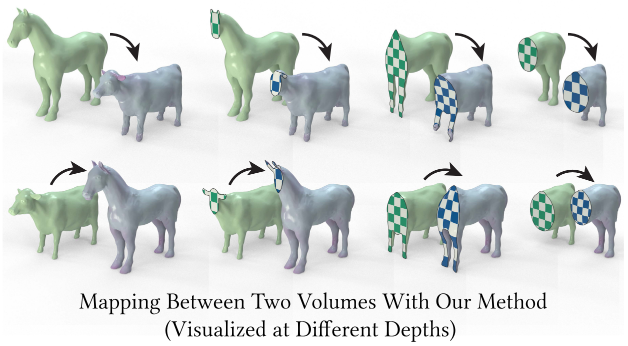

Mapping Between Two Volumes With Our Method
(Visualized at Different Depths)

Fig. 1. Our method produces low-distortion correspondences between volumes, visualized as checkerboard textures through the sliced volumes.

1 INTRODUCTION

Shape correspondences are at the core of many applications in graphics and geometry processing, including texture and segmentation transfer, animation, and statistical shape analysis. The central objective of these applications is to compute a dense map between two input shapes, facilitating semantically meaningful information transfer with minimal distortion.

The vast majority of shape correspondence algorithms focus on mapping two-dimensional surfaces. These approaches leverage geometric properties that are unique to surfaces. For example, key shape properties like curvature are defined over the entire surface domain, rather than only on the boundary as in the volumetric case. As a result, one can even find reasonable correspondences by matching geometric features directly, without incorporating any notion of distortion [Ovsjanikov et al. 2010]. Other methods use Tutte's embedding or notions of discrete conformality specific to surfaces to achieve key properties like invertibility [Lipman and Funkhouser 2009; Schmidt et al. 2019].

In contrast, here, we consider the problem of mapping *volumes* to *volumes* rather than surfaces to surfaces. Volumetric correspondence is beneficial for several tasks. In graphics and CAD, boundary representations of shapes are used to represent objects, so even the input geometry used to evaluate surface-to-surface mapping techniques typically expresses a volumetric domain. Hence, finding volumetric correspondences may improve correspondences of these boundary representations, since volumetric reasoning is needed to preserve thin features and prevent volumetric collapse; for example, to prevent the candy-wrapper artifact, where regions twist about a point and change orientation. In these cases, surface area is roughly maintained while volume degenerates. See Figure 2 (top) for an illustration using the surface mapping

This work is funded in part by NIH NIBIB NAC P41EB015902, NIH NICHD R01HD100009, Wistron Corporation, Army Research Office grants W911NF2010168 and W911NF2110293, of Air Force Office of Scientific Research award FA9550-19-1-031, of National Science Foundation grants IIS-1838071 and CHS-1955697, from the CSAIL Systems that Learn program, from the MIT–IBM Watson AI Laboratory, from the Toyota–CSAIL Joint Research Center, from a gift from Adobe Systems, the Swiss National Science Foundation's Early Postdoc.Mobility fellowship, the NSF Graduate Research Fellowship, the NSERC Postgraduate Scholarship – Doctoral, and the MathWorks Fellowship.

Authors' addresses: S. M. Abulnaga, P. Golland, and J. Solomon, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02143; emails: abulnaga@mit.edu, polina@csail.mit.edu, jsolomon@mit.edu; O. Stein, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02143 and University of Southern California, 941 Bloom Walk, Los Angeles, CA, 90089; email: ostein@mit.edu.

0730-0301/2023/04-ART25 $15.00
https://doi.org/10.1145/3572897

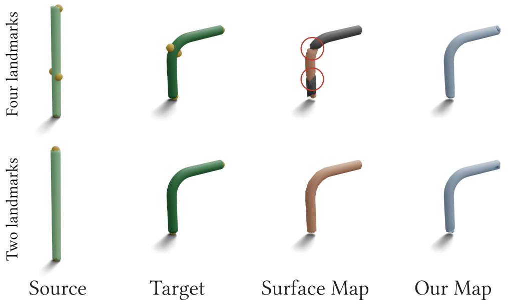


Fig. 2. Illustration of possible map degeneration when using a surface-mapping approach. Top row: Mapping using the surface-based approach of Ezuz et al. [2019] initialized with four landmark points (yellow spheres) leads to the candy-wrapper artifact, where regions of the mapped shape twist 180°, causing a change in orientation accompanied by a collapse in volume (red circles). The dark gray regions of the surface map show the backs of the triangles. Bottom row: mapping with two landmarks at the ends of the rods corrects the issue. In both cases, our volumetric approach maintains volumetric integrity and preserves orientation.

approach of Ezuz et al. [2019]. From a surface isometry perspective, the candy-wrapper artifact has little distortion, as only few edges have deformed. However, from a volumetric perspective, the shape's volume has completely degenerated. In other applications, such as medical imaging, data is acquired in a regular 3D grid and shape correspondence is used for volumetric texture transfer or alignment. Consequently, extending surface correspondences to the interior of volumetric shapes is nontrivial, so volumetric mapping approaches are needed.

Volumes do not share many of the geometric properties that have enabled mapping techniques for surfaces, so new approaches are needed. The closest existing methods to volumetric mapping tackle volumetric deformation and parameterization. In these applications, one starts with a volume in its rest pose and deforms the volume to a target domain or to conform to a set of target handle positions in a fashion that minimizes distortion. These approaches differ from volumetric mapping in several ways. First, volumetric deformation and parameterization methods typically assume a reasonable initial guess (e.g., the source shape) and flexibility in the target domain (e.g., unconstrained geometry away from the handles) or specialize to a single target (e.g., a ball). In contrast, in mapping, the source and target domains are geometrically distinct shapes, so a reasonable initialization is not given. One may need to start with a coarse map to a known set of landmarks [Aigerman et al. 2014; Ezuz et al. 2019]. Furthermore, mapping problems are typically symmetric, in the sense that the computed map should be invariant to the ordering of the source and target domains; there is no notion of a "rest pose" typical in deformation. Consequently, we seek a distortion energy that is symmetric with respect to the source and target.

We propose an algorithm for mapping between volumes represented as tetrahedral meshes. Our method draws insight from 2D surface mapping and 3D deformation. It builds on the discretization of maps used in a state-of-the-art surface mapping algorithm [Ezuz et al. 2019] but requires new objective functions and optimization methods to be effective. In particular, we propose a set of *symmetrized* distortion energies that are invariant to the domain over which the map is applied and aim to produce inversion-free, low-distortion matchings that conform closely to the boundary (Figure 1).

Contributions. This article contributes the following:

- We present a method for computing volumetric correspondences between far-from-isometric shapes by minimizing a symmetric distortion energy.
- We analyze the concept of a *symmetric* distortion energy, which is agnostic to the ordering of source and target domains, and provide a recipe for *symmetrizing* a distortion energy. We propose a set of desirable properties for a symmetric distortion energy and analyze well-known measures of distortion within our framework.
- We demonstrate our method on a diverse dataset of examples, showing that our method reliably extracts correspondences with low distortion.

1.1 Approach

We find a dense correspondence between two volumetric shapes M_1 and M_2 represented as tetrahedral meshes. Our algorithm simultaneously optimizes for a map $\phi : M_1 \to M_2$ and its (approximate) inverse $\psi \approx \phi^{-1} : M_2 \to M_1$, which both take vertices of one mesh to (interiors or boundaries of) tetrahedra in the other. Our approach handles meshes of differing connectivity and facilitates finding maps between far-from-isometric shapes.

Existing volumetric mapping methods use deformation techniques to place or repair interior tetrahedra, given a fixed map between the boundaries ∂M_1 and ∂M_2. In contrast, we include the boundary map as a variable. Our method can repair poorly initialized surface maps and compute maps using only landmark correspondences as initialization.

Our formulation is *symmetric* in that the computed map is invariant to the labeling of the "source" and the "target" among M_1 and M_2. The motivation for symmetry comes from several applications where the selection of a source or target shape is unnecessary. For example, in medical imaging, one is interested in finding correspondences between brain shapes extracted from magnetic resonance images (MRI) to perform comparisons of local cortical (brain tissue) thickness [Aganj et al. 2015]. Similar symmetry arises when seeking a correspondence between two humans standing in the same pose, and in general for applications seeking to align two shapes. The arbitrary choice of the source shape is a consequence of algorithm design rather than application need. Consequently, this choice can influence the correspondence result, introducing bias. As shown in Figure 3, an asymmetric method like Kovalsky et al. [2015] may result in unequal performance dictated by the choice of map direction. Further, the asymmetry of previous approaches in medical imaging have introduced bias in estimating the effects of Alzheimer's disease [Fox et al. 2011; Hua et al. 2011; Yushkevich et al. 2010].

A reasonable expectation is to produce the same map—up to inversion—regardless of the choice of the source and target shape, i.e., the ordering of M_1 and M_2. One way to achieve this is to use a symmetric energy. An energy E is symmetric if $E(\phi) = E(\phi^{-1})$ [Cachier and Rey 2000; Schmidt et al. 2019]. Since ϕ^{-1} is

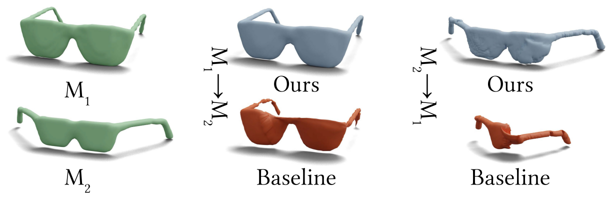


Fig. 3. Comparison between our symmetric approach and an asymmetric baseline. A symmetric approach is necessary when there is no clear source or target shape to produce high-quality bidirectional maps.

challenging to compute in practice and does not exist for maps initialized with flipped tetrahedra, we introduce $\psi \approx \phi^{-1}$ and propose a symmetric approach by optimizing $E(\phi) + E(\psi)$. Optimizing with this pair of maps is a common way of guaranteeing symmetry [Cachier and Rey 2000; Christensen and Johnson 2001; Ezuz et al. 2019; Schmidt et al. 2019; Schreiner et al. 2004], and we show via change-of-variables that optimizing this sum is *equivalent* to optimizing a different distortion energy $E^{\text{Sym}}(\phi)$ on just the forward map ϕ.

Key to computing a high-quality map is the proper choice of distortion energy E or its symmetrized counterpart E^{Sym}. We analyze the effect of symmetrizing several widely used distortion energies, showing that several symmetrized energies violate typical desiderata used to design mapping algorithms. For example, several symmetrized energies no longer favor local isometry. Following this analysis, we select the symmetrized ARAP energy as our distortion measure, eliminating solutions that locally favor collapsing or shrinking maps.

2 RELATED WORK

Volumetric correspondence poses a new set of challenges that has not been addressed in surface-based methods. Although relatively few works consider precisely the problem tackled in this article, we draw insights from volumetric parameterization, volumetric deformation, and surface mapping and focus our review on relevant work on these topics.

Volumetric parameterization and deformation. Parameterization and deformation algorithms provide means of deforming tetrahedral meshes into prescribed poses or domains with minimal distortion.

A *parameterization* is a deformation of a volume to a simpler domain, such as a topological ball [Abulnaga et al. 2022; Garanzha et al. 2021; Paillé and Poulin 2012; Wang et al. 2003; Yueh et al. 2019] or a polycube [Aigerman and Lipman 2013; Fu and Liu 2016; Li et al. 2021; Paillé and Poulin 2012; Wang et al. 2008b; Xia et al. 2010]. The better-studied instance of parameterization in graphics maps, possibly with cuts, two-dimensional surfaces (rather than volumes) into the plane; see Floater and Hormann [2005], Fu et al. [2021], and Sheffer et al. [2007] for discussion of this broad area of research.

In *deformation*, the task is to deform a volume by moving a set of handles to a set of target positions. These methods are often based on physical models of strain [Irving et al. 2004] and aim to produce elastic deformations minimizing a prescribed energy choice [Chao et al. 2010; Fu et al. 2015; Irving et al. 2004; Kovalsky et al. 2014; Müller et al. 2002; Sahillioğlu and Kavan 2015; Smith et al. 2018, 2019]. In the 2D case, both skeleton-based [Lewis et al. 2000] and physical models [Nealen et al. 2006] can be used. See Gain and Bechmann [2008], Selim and Koomullil [2016], and Sieger et al. [2015] for general discussion.

In both problems above, one computes a deformation from the rest pose to the target. Optimization methods are used to match the target while minimizing distortion, where the distortion is measured using an energy that quantifies the deformation of the Jacobian matrix of each tetrahedron. Since these models start with a good initialization, namely, the rest pose, one can optimize using a combination of energies with flip-free barriers and a constrained line search, arriving at solutions that are both flip-free and have low distortion; see, e.g., Smith and Schaefer [2015] for a representative example. In contrast to these past works, we produce maps between far-from-isometric domains without an obvious effective initialization. Consequently, our choice of energies is designed to be resilient to poor initial maps that are not foldover-free.

Volumetric mappings. Some methods consider the task of computing correspondences between volumetric shapes. To our knowledge, all past methods can be understood as special cases of the deformation methods where the task is to extend a fixed boundary map to the interior of a volume.

Kovalsky et al. [2015] present a local-global alternating algorithm targeting maps with bounded distortion. Their method takes an initial surface map and computes a similar map with bounded condition number. They demonstrate their algorithm on two volumetric correspondence examples and show one example (their Figure 11) where relaxing prescribed boundary constraints at the end of the optimization procedure can help recover from minor artifacts. Su et al. [2019] also target computation of foldover-free volumetric maps with prescribed boundary; they extend the method of Kovalsky et al. [2015] by automatically finding a suitable bound on the condition number. Their method has impressive levels of efficiency but targets a specific notion of conformal distortion. Stein et al. [2021] propose an operator splitting technique to optimize nonconvex distortion energies to yield a flip-free parameterization; they demonstrate a few examples of volumetric correspondence.

The approaches above require a prescribed boundary map and minimize distortion of the interior. In contrast, our method optimizes the boundary map to minimize global distortion and does not need a bijective, orientation-preserving boundary map as an initializer. Indeed, it is not always obvious how to design a boundary map so the induced volumetric correspondence has low isometric disortion. We also optimize an alternative objective function that targets symmetry and isometry rather than bounded distortion or conformal structure preservation.

A few mapping methods reduce a mapping problem between volumetric domains to a sequence of surface-mapping problems between leaves of foliations of the two domains. Campen et al. [2016] propose a volumetric parameterization approach relying on a foliation. Their algorithm requires the domain to be a topological ball whose tetrahedral mesh is *bishellable*. Cohen and Ben-Chen [2019] describe an alternative method to compute foliations of more-general volumetric domains using a Hele-Shaw flow along a potential function from a Möbius inversion of the domain boundary to a sphere. Unlike these methods that decompose the domain into surfaces, our method does end-to-end optimization of a mapping over an entire volume at once.

Symmetric maps. Symmetric mapping methods are invariant to the ordering of the source and target shapes. Several works in 2D surface mapping do so by optimizing for the average of the forward and reverse map distortion [Ezuz et al. 2019; Hass and Koehl 2017; Schmidt et al. 2019; Schreiner et al. 2004]. In medical imaging, mapping is referred to as registration, where the problem is to learn a displacement field defined on a 3D grid. Symmetry, or "inverse-consistency" [Christensen and Johnson 2001] is achieved using a similar approach of averaging the map distortions [Aganj et al. 2015; Cachier and Rey 2000; Leow et al. 2005; Sabuncu et al. 2009] or by optimizing in a mid-space between the two images [Avants et al. 2008; Joshi et al. 2004]. Many of these works demonstrate that symmetry improves consistency of mapping, improves accuracy, and eliminates bias.

We use a similar formulation to achieve symmetry. We analyze several common distortion energies symmetrized in this way and show that—surprisingly—the choice of energy can have counterintuitive consequences. In particular, distortion energies that favor isometry in one direction may not do so when optimizing their symmetrized counterparts. To prevent this undesired behavior, Hass and Koehl [2017] developed a symmetric distortion energy that measures the distance of a conformal map from an isometry. Their distortion energy is restricted to conformal maps between genus-0 surfaces. Extending it to the volumetric case is nontrivial due to the lack of conformal maps in 3D.

We develop the concept of a *symmetric* energy that is invariant to the choice of optimization domain over which it is taken, in the sense that the energy of the inverse map matches that of the forward map. Although it is a sensible choice in our setting, we note the term "symmetric" is somewhat overloaded in the parameterization and mapping literature. Several distortion measures have been deemed symmetric, because they equally penalize scaling and shrinking, such as the symmetric Dirichlet energy [Schreiner et al. 2004; Smith and Schaefer 2015] and the symmetric ARAP energy [Shtengel et al. 2017]. Our analysis shows that in fact these energies do not necessarily satisfy our notion of symmetry.

Surface maps. Two-dimensional surface mapping can generally be divided into (at least) three sets of approaches: methods that use an intermediate domain, methods that rely on descriptors, and methods that directly extract a map from one mesh into another. We refer the reader to one of several surveys for a broad overview [Li and Iyengar 2014; Sahillioğlu 2020; Van Kaick et al. 2011].

The first two groups of approaches cannot be directly extended to the volumetric case. In particular, while Tutte's parameterization provides a natural means of mapping surfaces bijectively to an intermediate domain and thus provides a natural means of initializing maps in the first category, no such canonical parameterization exists for volumes. Moreover, volumetric geometry descriptors do not appear to be sufficiently reliable for correspondence tasks.

Methods that find correspondences through an intermediate domain employ a bijective parameterization of each input to a simple domain such as the plane [Kraevoy and Sheffer 2004], the sphere [Gotsman et al. 2003; Haker et al. 2000; Lee and Kazhdan 2019], or a quotient manifold [Aigerman and Lipman 2015, 2016; Aigerman et al. 2014, 2015; Bright et al. 2017; Schmidt et al. 2019]. We also note methods such as Kim et al. [2011] and Lipman and Funkhouser [2009], which average multiple maps computed in a similar fashion. These approaches admit no obvious extension to volumes. First, the existence of a bijection to a simpler intermediate domain does not always exist. Second, many of these methods require introducing cutting seams [Aigerman et al. 2015], which becomes substantially more difficult in three dimensions. Furthermore, these may not result in low-distortion maps, as minimizing the composition of the maps in the intermediate domain may result in high distortion in the final surface-to-surface map.

The second set of methods computes maps that match descriptors, possibly with added regularization. Descriptors are often distance-based [Bronstein et al. 2008b; Huang et al. 2008], spectral [Jain et al. 2007; Mateus et al. 2008; Ovsjanikov et al. 2010; Vestner et al. 2017], extrinsic [Ankerst et al. 1999; Salti et al. 2014], or a combination [Dubrovina and Kimmel 2011; Kim et al. 2011; Litman and Bronstein 2013]. Many correspondence methods in this category are built on the functional maps framework [Ovsjanikov et al. 2012, 2016], which finds correspondences by matching functions defined on the shapes. Relatively few descriptors are available for volumetric geometry, whose structure is still inherited from the boundary surface.

The third class of approaches directly optimize for inter-surface maps. These methods compute a map between surfaces by matching features or landmarks while minimizing distortion [Ezuz et al. 2019; Mandad et al. 2017; Schreiner et al. 2004; Solomon et al. 2012, 2016].

Ezuz et al. [2019] produce a map between surfaces by minimizing the geodesic Dirichlet energy of the forward and reverse map and encouraging bijectivity through a reversibility energy. Our algorithm extends many of their ideas to the volumetric case. In our case, however, a new algorithm is required.

Medical image registration. Medical image registration is a form of volumetric shape correspondence in Euclidean space. Here, the task is to find correspondences between two volumes defined on a dense 3D grid. The correspondence is driven by matching voxel signal intensities, for example, using mutual information [Klein et al. 2007] or cross-correlation [Avants et al. 2008]. The optimization seeks to find a displacement field defined at the grid coordinates. Similar to our formulation, the transformation is governed by any of several regularization terms, for example, to compute a diffeomorphic transformation [Beg et al. 2005]. We refer readers to the surveys by Oliveira and Tavares [2014], Sotiras et al. [2013], and Viergever et al. [2016]. While both our approach and registration methods aim to find volumetric correspondences, the techniques used in medical image registration are not applicable, as they operate on a dense Euclidean grid and are driven by intensity rather than geometry.

3 MAPPING PROBLEM

We develop a volumetric mapping method that is symmetric, in that the resulting maps are invariant to the ordering of the source and target shapes. We compute the map by minimizing an objective function that measures distortion symmetrically while satisfying a set of constraints. In this section, we investigate the consequences of the symmetry assumption on our algorithmic design.

3.1 Preliminaries

Given two bounded volumes $M_1, M_2 \subset \mathbb{R}^3$ with smooth boundaries $\partial M_1, \partial M_2$, we seek a map $\phi : M_1 \to M_2$. Several considerations inform our choice of ϕ, detailed below. Note that this problem is not the same as *deformation* (sometimes referred to as "mapping" in past literature), which aims to find a low-distortion deformation of $M_1 \subset \mathbb{R}^3$ given prescribed target positions for a few handles rather than the geometry of M_2.

Many algorithms for mapping and deformation can be viewed as optimizing a distortion energy of the form

$$E_f[\phi] := \int_{M_1} f(J_\phi(\mathbf{x}))\, dV(\mathbf{x}), \tag{1}$$

where $J_\phi \in \mathbb{R}^{3\times 3}$ is the map Jacobian and $dV(\mathbf{x})$ is the volume form on M_1.

The distortion function $f : \mathbb{R}^{3\times 3} \to \mathbb{R}_{\geq 0}$ usually measures local deviation of the map from isometry. Typical choices favor rigidity [Rabinovich et al. 2017]. For example, the as-rigid-as-possible distortion function (ARAP) [Liu et al. 2008] measures the deviation of the Jacobian from the set of rotation matrices SO(3):

$$f_{\mathrm{ARAP}}(J) = \min_{R\in \mathrm{SO}(3)} \|J - R\|_F^2.$$

In contrast, the Dirichlet energy functional

$$f_{\mathrm{D}}(J) = \|J\|_F^2$$

favors the as-constant-as-possible map [Schreiner et al. 2004]. Selection of the distortion function is application-dependent. For example, one might choose f to model physical strain for deformation. Alternatively, one might select f to encourage injectivity.

In almost all applications, f is chosen to be *rotation invariant*, reflecting the fact that rigid motions of M_1 and M_2 should not affect the computed map. In this case, $f(J)$ is a function of the singular values $\boldsymbol{\sigma}(J)$, the elements of the diagonal matrix Σ in the singular value decomposition (SVD) $J = U\Sigma V^\top$. In a slight abuse of notation, in our subsequent discussion, we will use f to denote both a function on matrices in $\mathbb{R}^{3\times 3}$ and vectors of singular values in $\mathbb{R}^3$, with $f(J) := f(\boldsymbol{\sigma}(J))$.

In addition to finding a map with low distortion, we are concerned with finding one that satisfies a desired set of constraints. For example, we can constrain the boundary of the source volume to be mapped to the boundary of the target, i.e., $\phi(\partial M_1) = \partial M_2$. We use $\mathcal{P}$ to denote the constrained *feasible set*. One might imagine other constraints, for example, ensuring a set of landmark points are mapped to the pre-specified locations, further restricting $\mathcal{P}$. Moreover, regularizing objective terms, Reg[ϕ] could be added. So, our optimization problem becomes

$$\begin{aligned} \underset{\phi}{\arg\min} \quad & \int_{M_1} f(J_\phi(\mathbf{x}))\, dV(\mathbf{x}) + \mathrm{Reg}[\phi] \\ \text{subject to} \quad & \phi \in \mathcal{P}. \end{aligned} \tag{2}$$

3.2 Symmetrized Energy Functions

For correspondence problems where there is no clear distinction between the rest pose and the target pose, it is desirable for a volumetric correspondence method to be *symmetric*, meaning that it is invariant to the ordering of the "source" domain M_1 and "target" domain M_2. Symmetry requires $E_f[\phi] = E_f[\phi^{-1}]$. In this section, we arrive at a set of conditions on f to check if an energy is symmetric and propose a symmetrization procedure to obtain the symmetrized form of a distortion function f. We later investigate the effects on computing a map using the symmetrized form of f.

Following Cachier and Rey [2000], Christensen and Johnson [2001], Ezuz et al. [2019], Schmidt et al. [2019], and Schreiner et al. [2004], one simple way to achieve symmetry is to optimize the average of the distortion energy of a map with the distortion energy of its inverse. Ezuz et al. [2019] and Schreiner et al. [2004] use the simplest choice of energies to symmetrize—the Dirichlet energy—while Schmidt et al. [2019] use the symmetric Dirichlet energy to prevent foldovers. Below, we analyze the consequences of using these energies and other possible choices of f not considered in prior work. Surprisingly, our analysis will show that the Dirichlet energy and several other seemingly reasonable choices do *not* yield an effective notion of distortion after symmetrization, leading us to employ an alternative in our technique.

We start by deriving conditions on f that ensure the distortion energy E_f is invariant to the ordering of the source and target. Let M_1 and M_2 be open subsets of $\mathbb{R}^n$ and $\phi : M_1 \to M_2$ a diffeomorphism between them. For simplicity, assume M_1 and M_2 are normalized to have volume 1. We can compute the distortion of the map ϕ by applying Equation (1) in both directions:

$$E_f[\phi] = \int_{M_1} f\left(J_\phi(\mathbf{x})\right) dV_1(\mathbf{x}) \tag{3}$$

$$E_f[\phi^{-1}] = \int_{M_2} f\left(J_{\phi^{-1}}(\mathbf{y})\right) dV_2(\mathbf{y}). \tag{4}$$

Pulling back the integral in Equation (4) to M_1, we use a change of variables to $\mathbf{y} = \phi(\mathbf{u})$ to show

$$E_f[\phi^{-1}] = \int_{M_1} f\left(J_{\phi^{-1}}(\phi(\mathbf{u}))\right) \left|\det J_\phi(\mathbf{u})\right| dV_1(\mathbf{u}). \tag{5}$$

By the inverse function theorem,

$$E_f[\phi^{-1}] = \int_{M_1} f\left(\left(J_\phi(\mathbf{u})\right)^{-1}\right) \left|\det J_\phi(\mathbf{u})\right| dV_1(\mathbf{u}). \tag{6}$$

For invariance with respect to the integration domain, Equation (3) must agree with Equation (6). Matching the integrands,

$$\boxed{f(J) = |\det J|\, f\left(J^{-1}\right)}, \tag{7}$$

is sufficient for this equivalence. In terms of the singular values, we obtain

$$f(\boldsymbol{\sigma}) = \left|\prod_{i=1}^{n} \sigma_i\right| f\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right). \tag{8}$$

Here and in our subsequent discussion, we will use n to refer to the dimensionality of the domains M_1, M_2 when the result under discussion applies to maps in any dimension; $n = 3$ in our application. This condition was first proposed by Cachier and Rey [2000]

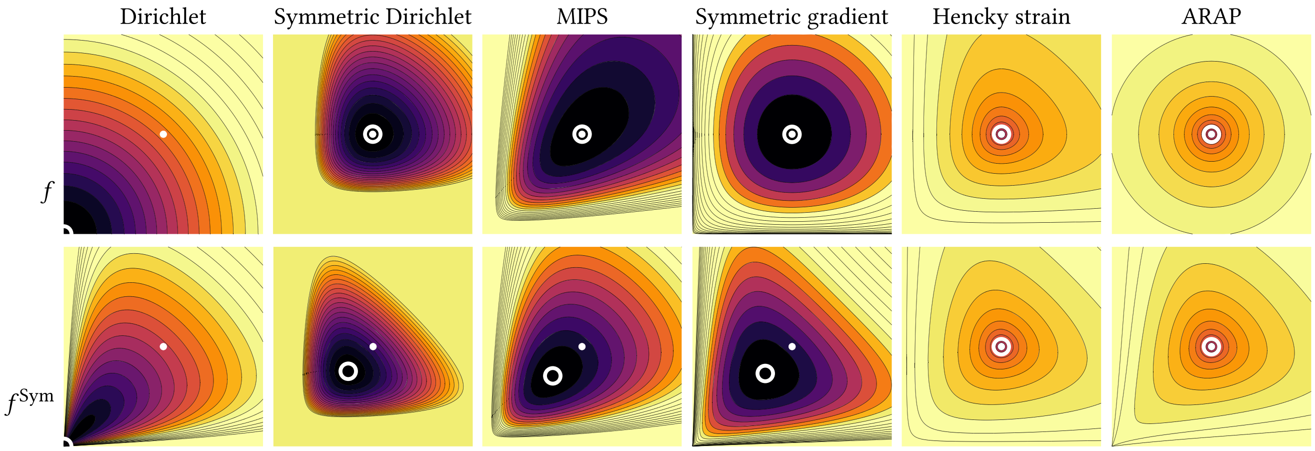


Fig. 4. Level sets of distortion functions f (top) and their symmetrized counterparts f^{Sym} (bottom) evaluated at $(\sigma_1, \sigma_2, 1)$ for $(\sigma_1, \sigma_2) \in [0, 2]^2$. We mark $(1, 1)$ as a white dot and the location of the minimum as a circle. In the parlance of Section 3.3, all energies except the Dirichlet energy preserve structure (f minimized at $(1, 1, 1)$), while only the Hencky strain and ARAP energies favor isometry (f^{Sym} minimized at $(1, 1, 1)$). Only Dirichlet and ARAP are nonsingular, since the level sets do not diverge as singular values approach 0.

to propose symmetrization by averaging the distortion function in both mapping directions. This motivates the following definition:

Definition 3.1 (Symmetric Energy). A distortion energy E_f whose distortion function f satisfies Equation (7)—or Equation (8) in terms of singular values—is a *symmetric energy.*

Our symmetric energy condition is both necessary and sufficient for symmetric distortion measures, in the following sense:

Proposition 3.2. *$E_f[\phi] = E_f[\phi^{-1}]$ for all M_1, M_2, and ϕ as defined above if and only if f is a symmetric energy.*

Proof. Substituting Equation (7) into Equation (6) shows that any f satisfying Equation (7) automatically satisfies $E_f[\phi] = E_f[\phi^{-1}]$. We now show the converse. Since $E_f[\phi] = E_f[\phi^{-1}]$ $\forall$ M_1, M_2, ϕ as defined above, we can choose $M_1 = B_1(\mathbf{0}) \subset \mathbb{R}^n$, the open ball of radius 1. Consider any invertible $J \in \mathbb{R}^{n\times n}$, and define a map $\phi(\mathbf{x}) := J\mathbf{x}$, whose Jacobian is given by $J_\phi(\mathbf{x}) = J$. Take $M_2 := \phi(M_1)$. Applying Equation (3),

$$E_f[\phi] = f(J) \cdot \mathrm{vol}\,(B_1(\mathbf{0})). \tag{9}$$

Similarly, applying Equation (6) yields

$$E_f[\phi^{-1}] = f(J^{-1})\,|\det J| \cdot \mathrm{vol}\,(B_1(\mathbf{0})). \tag{10}$$

Equating Equations (9) and (10) and dividing by $\mathrm{vol}\,(B_1(\mathbf{0}))$ completes the proof. □

Not all distortion energies are symmetric, but there is a simple procedure to construct a symmetric distortion function f^{Sym} from any distortion function f. For any distortion function f, we can obtain a corresponding f^{Sym} fulfilling Equation (7) by—in effect—computing $\frac{1}{2}E_f[\phi] + \frac{1}{2}E_f[\phi^{-1}]$ via our symmetrization procedure:

$$f^{\mathrm{Sym}}(J) = \frac{1}{2} f(J) + \frac{1}{2}\,|\det J|\, f(J^{-1}), \tag{11}$$

or in terms of singular values,

$$f^{\mathrm{Sym}}(\boldsymbol{\sigma}) = \frac{1}{2} f(\boldsymbol{\sigma}) + \frac{1}{2}\left|\prod_{i=1}^{n} \sigma_i\right| f\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right). \tag{12}$$

For example, suppose $f_{\mathrm{D}}(J) = \|J\|_F^2$ is the distortion function of the Dirichlet energy. Then, the average of the Dirichlet energy of the forward map and of the inverse map yields the distortion function:

$$f_{\mathrm{D}}^{\mathrm{Sym}}(J) = \frac{1}{2}\|J\|_F^2 + \frac{1}{2}|\det J|\|J^{-1}\|_F^2, \tag{13}$$

or for $n = 3$,

$$f_{\mathrm{D}}^{\mathrm{Sym}}(\sigma_1, \sigma_2, \sigma_3) = \frac{1}{2}\sum_{i=1}^{3} \sigma_i^2 + \frac{1}{2}\,(\sigma_1\sigma_2\sigma_3)\left(\sum_{j=1}^{3} \sigma_j^{-2}\right). \tag{14}$$

This is not the "symmetric" Dirichlet energy from past work on parameterization [Rabinovich et al. 2017; Smith and Schaefer 2015], which has the form $\frac{1}{2}\|J\|_F^2 + \frac{1}{2}\|J^{-1}\|_F^2$. Incidentally, in 2D, the second term in Equation (13) is the objective function of the inverse harmonic mapping problem used to obtain foldover-free mappings by Garanzha et al. [2021]. This term is also known as the inverse Dirichlet energy [Knupp 1995].

Equation (13) is a model for the objective function for mapping surfaces in Schreiner et al. [2004] and Ezuz et al. [2019], and one could reasonably attempt to reuse the same formulation for volumes. More careful examination of this function, however, indicates some undesirable properties. In particular, as illustrated in Figure 4, the distortion function $f_{\mathrm{D}}^{\mathrm{Sym}}(\boldsymbol{\sigma})$ is not minimized at $(1, 1, 1)$, the singular values of a rigid map. That is, the distortion function of the symmetrized Dirichlet energy f_{D}^{Sym} favors nonisometric maps, even though it is symmetric.

The counterintuitive behavior of energies like in Equation (13) suggests that algorithms optimizing the sum of the distortion of a map and the distortion of its inverse can have unpredictable behavior, even for standard choices of distortion functions. We examine this effect empirically in Section 6.6.

3.3 Designing Symmetric Distortion Energies

In this section, we extend the previous analysis to compute the symmetrized form of several commonly used distortion functions and examine their behavior in computing a volumetric map. We propose a list of desiderata to guide the selection of a desirable distortion function f.

Several properties are desirable when selecting f:

- Favors isometry: f^{Sym} is minimized at $(1, 1, 1)$.
- Preserves structure: f is minimized at $(1, 1, 1)$.
- Nonsingular: f is defined for all matrices.

Favoring isometry and preserving structure are similar but not identical conditions, and they are desirable for different reasons. Distortion energy functions that favor isometry are the typical choice for geometry processing applications, and this condition simply expresses a preference for maps ϕ that are rigid. However, structure-preserving choices of f facilitate optimization routines like ours that alternate between estimating ϕ and ψ, ensuring that both alternating steps work toward a common goal. Similarly, nonsingular functions f avoid the need for barrier optimization techniques and feasible initialization.

The following proposition provides a necessary condition that can be used to rule *out* many standard choices of f when considering the properties above:

Proposition 3.3. *Suppose a differentiable function* $f : \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$ *favors isometry and preserves structure, i.e.,* $f(\boldsymbol{\sigma})$ *and* f^{Sym} *are minimized at* $(1, 1, 1)$*. Then,* $f(1, 1, 1) = 0$ *and* $\nabla f(1, 1, 1) = (0, 0, 0)$*.*

Proof. Structure preservation immediately implies $\nabla f(1, 1, 1) = (0, 0, 0)$, since $(1, 1, 1)$ is a local minimum. Similarly, to favor isometry, we must have that $\nabla f^{\text{Sym}}(1, 1, 1) = (0, 0, 0)$. Taking the derivative of Equation (12) in one singular value σ_i, we find

$$\frac{\partial f^{\text{Sym}}}{\partial \sigma_i} = \frac{1}{2}\frac{\partial f}{\partial \sigma_i} + \frac{1}{2}\left|\prod_{j \neq i} \sigma_j\right| \left[f\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right) - \frac{1}{|\sigma_i|}\frac{\partial f}{\partial \sigma_i}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right)\right].$$

Substituting $\sigma_1 = \cdots = \sigma_n = 1$,

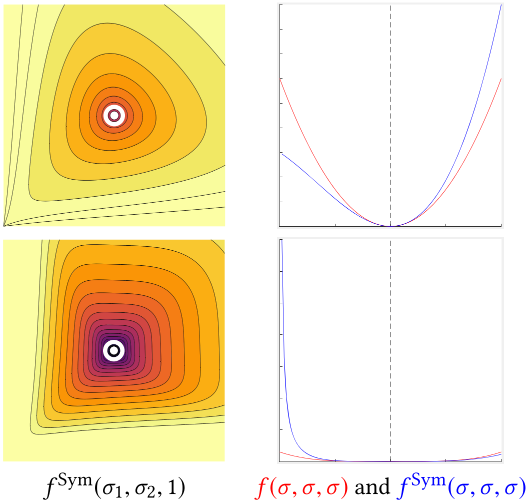
$$0 = \frac{\partial f^{\text{Sym}}}{\partial \sigma_i}(1, \ldots, 1) = \frac{1}{2} f(1, \ldots, 1).$$

This expression yields our first condition. □

The result above may feel somewhat counterintuitive, since constant shifts in f affect whether f favors isometry. But, adding a constant to f changes the effect of the volume form on the distortion energy, explaining the result above.

In Table 1, we list several distortion functions $f(J)$, their equivalent forms in terms of the Jacobian J's singular values $f(\boldsymbol{\sigma})$, and their symmetrized forms $f^{\text{Sym}}(J)$, $f^{\text{Sym}}(\boldsymbol{\sigma})$. We check if the symmetrized distortion functions satisfy the isometry favoring property above by examining the behavior of $\boldsymbol{\sigma}_{\min}$, the singular values that minimize $f^{\text{Sym}}(\boldsymbol{\sigma})$. We verify the other properties in a similar way by studying $f(\boldsymbol{\sigma})$. Table 2 summarizes the result. Figure 4 visualizes these properties by showing level sets of f and f^{Sym} for examples drawn from Table 1.

Tables 1 and 2 reveal several valuable properties that can inform our choice of f. None of the distortion energies in Table 1 is symmetric in its standard form. A surprising result is that, after symmetrization, no distortion energy except for ARAP and Hencky strain favors isometry. Despite the fact that minimizing these energies in the forward or reverse direction independently would lead to an isometry, minimizing for the average of the two does not (see Figure 4). For example, the symmetric Dirichlet energy and the AMIPS energy after symmetrization prefer maps that tend to shrink ($\sigma_{\min} < 1$). We also observe that the symmetrized Dirichlet, the symmetrized 3rd-order Dirichlet, and the symmetrized MIPS energies favor maps that collapse, that is, they are minimized close to $\sigma_{\min} \approx (0, 0, 0)$. While the (asymmetric) Dirichlet energy favors maps with $\boldsymbol{\sigma} = \mathbf{0}$, the MIPS energy does not. The 3rd-order Dirichlet energy is used in 3D for C^1 continuity [Iwaniec and Onninen 2010].


$f^{\text{Sym}}(\sigma_1, \sigma_2, 1)$

$f(\sigma, \sigma, \sigma)$ and $f^{\text{Sym}}(\sigma, \sigma, \sigma)$

Fig. 5. Mathematical boundary case: Comparison of symmetrized ARAP energy $\sum_i (\sigma_i - 1)^2$ to symmetrized fourth-power ARAP energy $\sum_i (\sigma_i - 1)^4$, using level sets similar to Figure 4 (left) and by plotting the diagonal where $\sigma = \sigma_1 = \sigma_2 = \sigma_3$ (right). As discussed in Section 3.3 (Remark), the fourth-power alternative blows up when approaching $(0, 0, 0)$ from *any* direction, while conventional ARAP admits a path to $(0, 0, 0)$ where the energy density remains finite.

From Table 2, only the symmetrized ARAP energy, which we will refer to as *sARAP*, satisfies all the desired properties. To implement the sARAP energy, we optimize the average of the ARAP energy of the forward and reverse maps. This objective function has the added benefit of removing the requirement of a flip-free initialization, which is often not available for correspondence tasks.

If M_1 and M_2 have different volumes, then the forward and backward terms in Equations (3) and (4) might prefer distortion of one direction over another. In practice, we normalize our models to have volume 1, so the integrals in Equations (3) and (4) measure average local distortion of the two maps; Schreiner et al. [2004] equivalently rescales the forward and backward terms.

Remark (Avoiding Zero Singular Values). The symmetric Dirichlet energy [Smith and Schaefer 2015], symmetric gradient energy [Stein et al. 2021], and others used for bijective parameterization *blow up* as singular values approach zero; this property provides a barrier ensuring existence of a locally optimal parameterization without collapsed or inverted elements. Our *nonsingular* property actually prefers the opposite of this scenario, allowing inverted Jacobians so we can recover from poor initialization, but this is a property of f—employed during optimization—rather than f^{Sym}, the actual distortion energy being optimized in the symmetrized formulation.

A nonsingular f can actually admit a function f^{Sym} that blows up as singular values approach 0, as is the case for the ARAP and Dirichlet energies. This property suggests that even a nonsingular choice of f can favor orientation-preserving symmetric maps.

For completeness, we note that $f^{\text{Sym}}_{\text{ARAP}}$ is not a *perfect* barrier, in the following sense (also illustrated in Figure 5): For $\sigma_1 = 1$ and $\sigma_2, \sigma_3 \to 0$, we have $f^{\text{Sym}}_{\text{ARAP}}(\boldsymbol{\sigma}) \to 1$. This technicality can be addressed using an f that grows faster than cubically in the singular values, e.g., $f(\boldsymbol{\sigma}) = \sum_i (\sigma_i - 1)^4$, but in practice such an adjustment did not yield better maps.

Remark (Role of Boundary Conditions). Several prior works optimize symmetric energies *without* the desired properties at the beginning of this section [Ezuz et al. 2019; Schmidt et al. 2019; Schreiner et al. 2004]. Although their distortion energies do not promote isometry directly, these methods are still able to find low-distortion and even bijective correspondences. Indeed, the symmetrized energy analysis above does not tell the whole story. In particular, these methods include energy terms, boundary conditions, and other constraints that favor bijectivity and semantic correspondences. These constraints counteract the energy's unexpected local properties and can affect the resulting map quality. For example, optimizing the symmetrized Dirichlet energy in the space of surjective or bijective maps will prevent the map from collapsing, but the map quality is essentially upheld by the boundary condition rather than the constitutive model used in the objective function. We hypothesize that the success of these methods lies in balancing competing terms and constraints. We leave detailed theoretical analysis of these intriguing global questions to future work.

3.4 Symmetric Optimization Problem

Following the previous section's analysis, we revise the the generic formulation of our optimization problem in Equation (2) to be symmetric. We optimize an energy of the form $\frac{1}{2}E_f[\phi] + \frac{1}{2}E_f[\psi]$, where we maintain separate estimates of the map $\phi : M_1 \to M_2$ and its inverse $\psi \approx \phi^{-1} : M_2 \to M_1$. This is done for practical reasons: The existence of a flip-free initial map is not guaranteed, so ϕ^{-1} may not exist to start. Additionally, this form is advantageous, as f is necessarily nonsingular for initializations with flipped elements, while f^{Sym} can be orientation-preserving as is the case for sARAP. Finally, even if f is not symmetric, the resulting energy is roughly of the form in Equation (11) and hence our analysis in Section 3.2 applies. This leads to the modified problem:

$$\begin{aligned}\operatorname*{arg\,min}_{\phi,\psi} \quad & \frac{1}{2}\int_{M_1} f_{\text{ARAP}}(J_\phi(\mathbf{x}))\, dV(\mathbf{x}) \\ & + \frac{1}{2}\int_{M_2} f_{\text{ARAP}}(J_\psi(\mathbf{y}))\, dV(\mathbf{y}) + \text{Reg}[\phi,\psi] \\ \text{subject to} \quad & \phi \in \mathcal{P}, \psi \in \mathcal{Q},\end{aligned} \tag{15}$$

where $\mathcal{Q}$ denotes the constraint $\psi(\partial M_2) \subset \partial M_1$. In practice, the constraints that define $\mathcal{P}$ and $\mathcal{Q}$ can be made soft and modeled in $\text{Reg}[\phi,\psi]$. The estimate $\psi \approx \phi^{-1}$ can be enforced as a soft or hard constraint. In practice, we use a soft constraint modeled in $\text{Reg}[\phi,\psi]$, as described in Section 4.3.

4 DISCRETIZATION AND MODEL

We build on our analysis in Section 3.2 and Section 3.3 to discretize the optimization problem in Equation (15) and develop an algorithm to compute a volumetric map that is invariant to the ordering of the source and target shapes. In this section, we define our map discretization and map constraints and develop the objective function used in the optimization.

4.1 Notation

We represent volumetric shapes as tetrahedral meshes. We let $\mathcal{V}_i$, $\mathcal{E}_i$, $\mathcal{F}_i$, $\mathcal{T}_i$ denote the sets of vertices, edges, faces, and tetrahedra of mesh M_i, for $i \in \{1,2\}$. We represent the coordinates of $\mathcal{V}_i$ as a matrix $V_i \in \mathbb{R}^{n_i \times 3}$, where n_i denotes the number of vertices in mesh M_i. We represent tetrahedron k in mesh i as the matrix $V_i^{T_k} \in \mathbb{R}^{4\times 3}$ whose rows are the coordinates of the vertices of tetrahedron k. We use ∂ to denote the boundary of a mesh, and $\partial\mathcal{V}_i, \partial\mathcal{E}_i, \partial\mathcal{F}_i, \partial\mathcal{T}_i$ denote sets of boundary vertices, edges, faces, and tetrahedra, respectively. Boundary tetrahedra are those that contain one or more boundary faces.

We use a piecewise linear discretization to model the maps ϕ and ψ, with each tetrahedron being mapped affinely. The map on each tetrahedron is determined by its transformed vertex coordinates. We use matrix $X_i \in \mathbb{R}^{n_1\times 3}$ to denote the coordinates of the transformed vertices of mesh M_i, and $X_i^{T_k} \in \mathbb{R}^{4\times 3}$ to denote the transformed tetrahedron k of mesh M_i. The Jacobian matrix

$$J\left(X_i^{T_k}\right) = \left(BX_i^{T_k}\right)\left(BV_i^{T_k}\right)^{-1} \tag{16}$$

defines the map differential of tetrahedron k based on the transformed coordinates $X_i^{T_k}$. The constant matrix $B \in \mathbb{R}^{3\times 4}$ extracts vectors parallel to the edges of the tetrahedron.

4.2 Map Representation

We wish to constrain each map to lie within the target shape, i.e., $\phi(M_1) \subset M_2$ and $\psi(M_2) \subset M_1$. We extend the strategy of Ezuz et al. [2019] to tetrahedral meshes to enforce these constraints.

We represent the map ϕ as a matrix $P_{12} \in [0,1]^{n_1\times n_2}$ and the map ψ as $P_{21} \in [0,1]^{n_2\times n_1}$. Matrices P_{12} and P_{21} use barycentric coordinates to encode the vertex-to-tetrahedron map and ensure the mapped vertices lie in the target mesh. This representation is also beneficial to map between meshes with differing connectivity. Suppose P_{12} maps vertex i of mesh M_1 into tetrahedron $T_k = (a,b,c,d) \in \mathcal{T}_2$ in mesh M_2, where $(a,b,c,d) \in \{1,\ldots,n_2\}$ are the indices of the vertices of T_k. Then, row i of P_{12} contains the barycentric coordinates of the image of vertex i in columns a,b,c,d, and zeros elsewhere. Map P_{21} is constructed analogously. We can enforce the constraint that boundary vertices are mapped to boundary faces by constraining the sparsity patterns of P_{12} and P_{21}. A limitation in the discretization is that we are unable to enforce that the interior of boundary faces and edges are mapped inside the target shape, since our map representation is vertex-based. In practice, this effect is minimized using high-resolution meshes.

We denote the set of all feasible maps satisfying the boundary constraints as $\mathcal{P}^{\star}_{ij}$; we use $\mathcal{P}_{ij}$ to denote the set of feasible maps that may map the boundary ∂M_i to the interior of M_j.

We use half-quadratic splitting [Geman and Yang 1995] to express our problem in a form that is amenable to efficient

Table 1. Several Distortion Measures and Their Symmetrized Forms

Name	$f(J)$	$f(\sigma)$	$f^{\text{Sym}}(J)$	$f^{\text{Sym}}(\boldsymbol{\sigma})$	$\boldsymbol{\sigma}_{\min}$
Dirichlet	$\lVert J\rVert_F^2$	$\sum_{i=1}^n \sigma_i^2$	$\frac{1}{2}\lVert J\rVert_F^2 + \frac{1}{2}(\det J)\left(\lVert J^{-1}\rVert_F^2\right)$	$\frac{1}{2}\sum_{i=1}^n \sigma_i^2$ $+\frac{1}{2}\left(\prod_{j=1}^n \sigma_j\right)\left(\sum_{k=1}^n \sigma_k^{-2}\right)$	$\approx (0, 0, 0)$
Dirichlet (3rd order)	$\lVert J\rVert_F^3$	$\sum_{i=1}^3 \sigma_i^3$	$\frac{1}{2}\lVert J\rVert_F^3 + \frac{1}{2}(\det J)\left(\lVert J^{-1}\rVert_F^3\right)$	$\frac{1}{2}\sum_{i=1}^3 \sigma_i^3$ $+\frac{1}{2}\left(\prod_{j=1}^3 \sigma_j\right)\left(\sum_{k=1}^3 \sigma_k^{-3}\right)$	$\approx (0, 0, 0)$
Symmetric Dirichlet	$\lVert J\rVert_F^2 + \lVert J^{-1}\rVert_F^2$	$\sum_{i=1}^n \left(\sigma_i^2 + \sigma_i^{-2}\right)$	$\frac{1}{2}(\det J + 1)\left(\lVert J\rVert_F^2 + \lVert J^{-1}\rVert_F^2\right)$	$\frac{1}{2}\left(\prod_{i=1}^n \sigma_i + 1\right)\left(\sum_{j=1}^n \left(\sigma_j^2 + \sigma_j^{-2}\right)\right)$	$\approx (0.77, 0.77, 0.77)$
MIPS (3D)	$\frac{1}{8}\left(\lVert J\rVert_F^2 \cdot \lVert J^{-1}\rVert_F^2 - 1\right)$	$\frac{1}{8}\prod_{i=1}^3 \left(\frac{\sigma_i}{\sigma_{i+1}} + \frac{\sigma_{i+1}}{\sigma_i}\right)$	$\frac{1}{16}(\det J + 1)\left(\lVert J\rVert_F^2 \cdot \lVert J^{-1}\rVert_F^2 - 1\right)$	$\frac{1}{16}\left(1 + \prod_{i=1}^3 \sigma_i\right)\left(\prod_{j=1}^3 \left(\frac{\sigma_{j+1}}{\sigma_j} + \frac{\sigma_j}{\sigma_{j+1}}\right)\right)$	$\approx (0, 0, 0)$
AMIPS (3D)	$\frac{1}{16}\left(\lVert J\rVert_F^2 \cdot \lVert J^{-1}\rVert_F^2 - 1\right)$ $+\frac{1}{2}\left(\det J + (\det J)^{-1}\right)$	$\frac{1}{16}\prod_{i=1}^3 \left(\frac{\sigma_i}{\sigma_{i+1}} + \frac{\sigma_{i+1}}{\sigma_i}\right)$ $+\frac{1}{2}\left(\prod_{j=1}^3 \sigma_j + \prod_{k=1}^n \sigma_k^{-1}\right)$	$\frac{\det J + 1}{32}\left(\lVert J\rVert_F^2 \cdot \lVert J^{-1}\rVert_F^2 - 1\right)$ $+\frac{1}{4}\left(\det J + (\det J)^{-1}\right)$ $+\frac{1}{4}\left((\det J)^2 + 1\right)$	$\frac{1}{32}\left(1 + \prod_{i=1}^3 \sigma_i\right)\left(\prod_{j=1}^n \left(\frac{\sigma_j}{\sigma_{j+1}} + \frac{\sigma_{j+1}}{\sigma_j}\right)\right)$ $+\frac{1}{4}\left(\prod_{k=1}^3 \sigma_k + \prod_{l=1}^n \sigma_l^{-1}\right)$ $+\frac{1}{4}\left(\prod_{m=1}^3 \sigma_m^2\right)$	$\approx (0.8, 0.8, 0.8)$
Conformal AMIPS	$\frac{\operatorname{tr}\left(J^T J\right)}{(\det J)^{\frac{2}{3}}}$	$\left(\prod_{j=1}^3 \sigma_j^{-\frac{2}{3}}\right)\left(\sum_{i=1}^3 \sigma_i^2\right)$	$\frac{1}{2}(\det J)^{-\frac{2}{3}} \operatorname{tr}\left(J^T J\right)$ $+\frac{1}{2}(\det J)^{\frac{1}{3}} \operatorname{tr}\left(J^{-T} J^{-1}\right)$	$\frac{1}{2}\left(\prod_{i=1}^3 \sigma_i^{-\frac{2}{3}}\right)\left(\sum_{j=1}^3 \sigma_j^2\right)$ $+\frac{1}{2}\left(\prod_{k=1}^3 \sigma_k^{-\frac{1}{2}}\right)\left(\sum_{l=1}^3 \sigma_l^{-2}\right)$	$\approx (0.032, 0.032, 0.032)$
Symmetric gradient	$\frac{1}{2}\lVert J\rVert_F^2 - \log(\det J)$	$\frac{1}{2}\sum_{j=1}^n \sigma_j^2 - \log\left(\prod_{i=1}^n \sigma_i\right)$	$\frac{1}{4}\lVert J\rVert_F^2 - \frac{1}{2}\log(\det J)$ $+\frac{1}{4}\det J \cdot \lVert J^{-1}\rVert_F^2$ $+\frac{1}{8}\det J \cdot \log(\det J)$	$\frac{1}{4}\sum_{i=1}^n \sigma_i^2 - \frac{1}{2}\log\left(\prod_{j=1}^n \sigma_j\right)$ $+\frac{1}{4}\left(\prod_{k=1}^n \sigma_k\right)\Big[\sum_{l=1}^n \sigma_l^{-2}$ $+\frac{1}{2}\log\left(\prod_{m=1}^n \sigma_m\right)\Big]$	$\approx (0.61, 0.61, 0.61)$
Hencky strain	$\lVert \log J^T J\rVert_F^2$	$\sum_{i=1}^n \log^2(\sigma_i)$	$\frac{1}{2}\lVert \log J^T J\rVert_F^2$ $+\frac{1}{2}\det J \cdot \lVert \log J^{-T} J^{-1}\rVert_F^2$	$\frac{1}{2}\sum_{i=1}^n \log^2(\sigma_i)$ $+\frac{1}{2}\left(\prod_{j=1}^n \sigma_j\right)\left(\sum_{k=1}^n \log^2(\sigma_k)\right)$	$(1, 1, 1)$
ARAP	$\lVert J - R\rVert_F^2$	$\sum_{i=1}^n (\sigma_i - 1)^2$	$\frac{1}{2}\lVert J - R\rVert_F^2$ $+\frac{1}{2}\det J \cdot \lVert J^{-1} - R\rVert_F^2$	$\frac{1}{2}\sum_{i=1}^n (\sigma_i - 1)^2 +$ $\frac{1}{2}\left(\prod_{j=1}^n \sigma_j\right)\left(\sum_{k=1}^n (\sigma_k^{-1} - 1)^2\right)$	$(1, 1, 1)$

In this table, we consider orientation-preserving maps, so $|\det J| = \det J$. We use an interior-point method constrained to search over non-negative σ to compute the set of singular values $\boldsymbol{\sigma}_{\min}$ that minimize the symmetrized energy $f^{\text{Sym}}(\boldsymbol{\sigma})$.

Table 2. Summary of Distortion Energy Function Properties

Name	Favors isometry	Preserves structure	Nonsingular
Dirichlet	✗	✗	✓
Dirichlet (3rd order)	✗	✗	✓
Symm. Dirichlet	✗	✓	✗
MIPS (3D)	✗	✓	✗
AMIPS (3D)	✗	✓	✗
Conformal AMIPS	✗	✓	✗
Symm. Gradient	✗	✓	✗
Hencky strain	✓	✓	✗
ARAP	✓	✓	✓

optimization [Ezuz et al. 2019; Wang et al. 2008a; Zoran and Weiss 2011]. In particular, we introduce the auxiliary variable X_{ij} to model the image of vertices $\mathcal{V}_i$ under the map to mesh M_j, where $X_{ij} \approx P_{ij}V_j$.

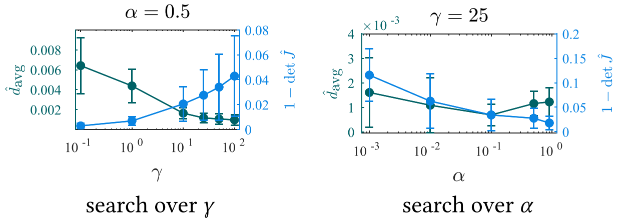
4.3 Objective Terms

We define several objective terms used to find the correspondence and model the soft constraints on the map.

4.3.1 Auxiliary and Reversibility Energy Functions. Our first two terms are adapted from Ezuz et al. [2019] and extended for volumetric meshes. The first term is the auxiliary energy that encourages $X_{ij} \approx P_{ij}V_j$:

$$E_Q[P_{12}, P_{21}, X_{12}, X_{21}] = \sum_{\substack{i,j \in \{1,2\} \\ i \neq j}} \frac{1}{c_i c_j} \left\lVert X_{ij} - P_{ij}V_j \right\rVert_{M_i}^2, \tag{17}$$

where c_i, c_j are the total volumes of meshes M_i and M_j, and $\lVert \cdot \rVert_{M_i}^2$ denotes the Frobenius norm with respect to M_i. For a matrix G, $\lVert G\rVert_{M_i}^2 = \operatorname{tr}(G^T C_i G)$, where C_i is the lumped diagonal vertex mass matrix of M_i.



search over γ

search over α

Fig. 6. Parameter sweep over γ and α, comparing the tradeoff between $\hat{d}_{avg}$ and $1 - \det \hat{J}$, where $\det \hat{J}$ is the normalized determinant of the Jacobian. We select $\alpha = 0.5$, $\gamma = 25$, as they achieve a reasonable tradeoff between conforming to the target boundary while maintaining map quality.

The second term is the reversibility energy that encourages bijectivity:

$$E_R[P_{12}, P_{21}, X_{12}, X_{21}] = \sum_{\substack{i,j \in \{1,2\} \\ i \neq j}} \frac{1}{c_i^2} \lVert P_{ij}X_{ji} - V_i\rVert_{M_i}^2. \tag{18}$$

This energy measures the distance between the original vertex positions V_i and the back projection of their image under the map P_{ij}, X_{ij}.

4.3.2 ARAP Energy. Central to the computation of a volumetric map is the proper selection of a distortion energy. From our analysis in Section 3.3, we select the sARAP energy, as it is both symmetric and promotes rigidity.

We use $\frac{1}{2}E_{ARAP}[\phi] + \frac{1}{2}E_{ARAP}[\psi]$ to approximate $E_{sARAP}[\phi]$. We approximate the integral over the volumetric domain by measuring the distortion energy per tetrahedron. For tetrahedron k of mesh i, the ARAP distortion function is given by

$$f_{ARAP}\left(J\left(X_{ij}^{T_k}\right)\right) = \sum_{j=1}^{3}(\sigma_{k,j} - 1)^2, \tag{19}$$

where $\sigma_{k,j}$ is the jth *signed* singular value of $J(X_{ij}^{T_k})$. We use the convention laid out by Irving et al. [2004] to define the signed singular value decomposition unambiguously. For $J = U\Sigma V^T$, this convention allows the sign of the smallest singular value σ_{min} to be negative, $\text{sign}(\sigma_{min}) = \text{sign}(\det J)$, and $U, V \in \text{SO}(3)$.

The total ARAP energy is then

$$E_{ARAP}\left[X_{12}, X_{21}\right] = \sum_{\substack{i,j\in\{1,2\}\\ i\neq j}} \frac{1}{2c_i} \sum_{T_k\in\mathcal{T}_i} v(T_k) f_{ARAP}\left(J\left(X_{ij}\right)^{T_k}\right), \tag{20}$$

where $v(T_k)$ denotes the volume of tetrahedron k.

4.3.3 Projection Energy. We encourage preserving the boundary of the source and target meshes by using forward and backward projection energies. We compute the forward projection energy $E_{P,f}$ as

$$E_{P,f}[X_{12}, X_{21}] = \sum_{\substack{i,j\in\{1,2\}\\ i\neq j}} \frac{1}{s_i} \left\| \left(X_{ij}\right)_{\partial M_i} - \text{proj}\left((X_{ij})_{\partial M_i}, \partial M_j\right) \right\|^2_{\partial M_i}, \tag{21}$$

where $\text{proj}((X_{ij})_{\partial M_i}, \partial M_j)$ denotes the Euclidean projection of the boundary vertices of ∂M_i with coordinates X_{ij} onto the boundary mesh ∂M_j, s_i denotes the total surface area of ∂M_1, and $\|\cdot\|^2_{\partial M_i}$ denotes the Frobenius norm with respect to boundary triangle mesh ∂M_i.

The backward projection energy $E_{P,b}$ is given by

$$E_{P,b}[X_{12}, X_{21}] = \sum_{\substack{i,j\in\{1,2\}\\ i\neq j}} \frac{1}{s_i} \left\| V_i - \text{proj}\left(V_i, \partial F_j\left(X_{ji}\right)\right) \right\|^2_{\partial M_i}, \tag{22}$$

where $\partial F_j(X_{ji})$ denotes the boundary of mesh M_j with vertices given by X_{ji}.

The full projection energy is then

$$E_P[X_{12}, X_{21}] = E_{P,f}[X_{12}, X_{21}] + E_{P,b}[X_{12}, X_{21}]. \tag{23}$$

4.4 Optimization Problem

Combining the distortion and regularization energies, our optimization problem becomes

$$\begin{aligned} \underset{P_{12},P_{21},X_{12},X_{21}}{\arg\min} \quad & E[P_{12}, P_{21}, X_{12}, X_{21}] \\ \text{subject to} \quad & P_{12} \in \mathcal{P}_{12}, P_{21} \in \mathcal{P}_{21}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} & E[P_{12}, P_{21}, X_{12}, X_{21}] \\ &= \sum_{\substack{i,j\in\{1,2\}\\ i\neq j}} \alpha E_{ARAP}[X_{ij}] + (1-\alpha)E_R[P_{ij}, X_{ji}] \\ &\quad + \gamma E_P[X_{ij}] + \beta E_Q[X_{ij}, P_{ij}]. \end{aligned} \tag{25}$$

Several parameters govern the strength of the distortion energies and soft constraints. The parameter $\alpha \in [0, 1]$ models the tradeoff between a reversible map (small $\alpha \to 0$) and one that maintains the rest shape ($\alpha \to 1$). The parameter $\gamma \in \mathbb{R}_{\geq 0}$ weighs the projection term that models the soft constraint for matching to the target boundary. The parameter β controls the soft constraint on the auxiliary variables. As recommended by Ezuz et al. [2019] and Wang et al. [2008a], β should use an update schedule tailored per application. In our experiments, since we start with a coarse initialization of the interior, we initialize $\beta = 0.25$ and increase β linearly to 5 over 20 iterations. We found our approach to be insensitive to the update schedule.

In this formulation, we use a soft constraint measured by E_P to map to the target boundary. While we could use a hard constraint by setting $\gamma = 0$ and requiring $P_{12} \in \mathcal{P}^{\star}_{12}, P_{21} \in \mathcal{P}^{\star}_{21}$, we did not find that this hard constraint had a substantial effect on our final output.

5 OPTIMIZATION

In this section, we outline our optimization procedure. We discuss strategies for initializing the map and propose an approach to uninvert tetrahedra. We conclude by presenting our algorithm for minimizing Equation (24) using block coordinate descent.

5.1 Initialization

Objective function (24) includes four variables: P_{12}, P_{21}, X_{12}, and X_{21}. In this section, we provide strategies for initializing the variables P_{ij} before running our optimization procedure. We initialize the X_{ij} variables via $X_{ij} \leftarrow P_{ij}V_j$.

Landmark-based initialization. If we are given landmark pairs $(\mathbf{p}_i, \mathbf{q}_i)$, where $\mathbf{p_i} \in M_1, \mathbf{q_i} \in M_2$, then we can initialize each landmark's target by copying the target of its closest landmark.

2D surface map initialization. A second approach is to initialize the boundaries of M_1, M_2 using an existing surface-to-surface mapping approach. We initialize the interior vertices identically to landmark-based initialization, where we consider *every* boundary vertex to be a landmark.

We do not hold the landmark or surface map vertices fixed during the optimization.

5.2 Alternating Minimization

We use coordinate descent, alternating between optimizing over X_{ij} and P_{ij}. Our multi-step optimization procedure ensures strong conformation to the boundary while avoiding inverted tetrahedra.

Optimizing for X_{ij}. Optimizing for X_{ij} while holding the P_{ij} variables fixed is a smooth optimization problem, for which we use the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [Zhu et al. 1997].

We compute the gradient of each energy term in Equation (25). The gradients for E_P, E_Q are straightforward, as they are matrix norms. We compute the gradient of E_{ARAP} using the chain rule. First, we compute the gradient of $f_{ARAP}(J)$ with respect to a Jacobian J, $\nabla_J f_{ARAP}[J] = U\text{diag}\left(\nabla_\sigma f_{ARAP}(\sigma)\right)V^T$. Using the chain rule, we then compute the gradient with respect to the elements of tetrahedron $T_k \in \mathcal{T}_i$, with coordinates $X_{ij}^{T_k}$,

$$\frac{\partial f_{ARAP}(X_{ij}^{T_k})}{\partial(X_{ij}^{T_k})} = \left(\left(BV_i^{T_k}\right)^{-T} B\right)\left(U\text{diag}\left(\nabla_\sigma f_{ARAP}(\sigma)\right)V^T\right)^T. \tag{26}$$

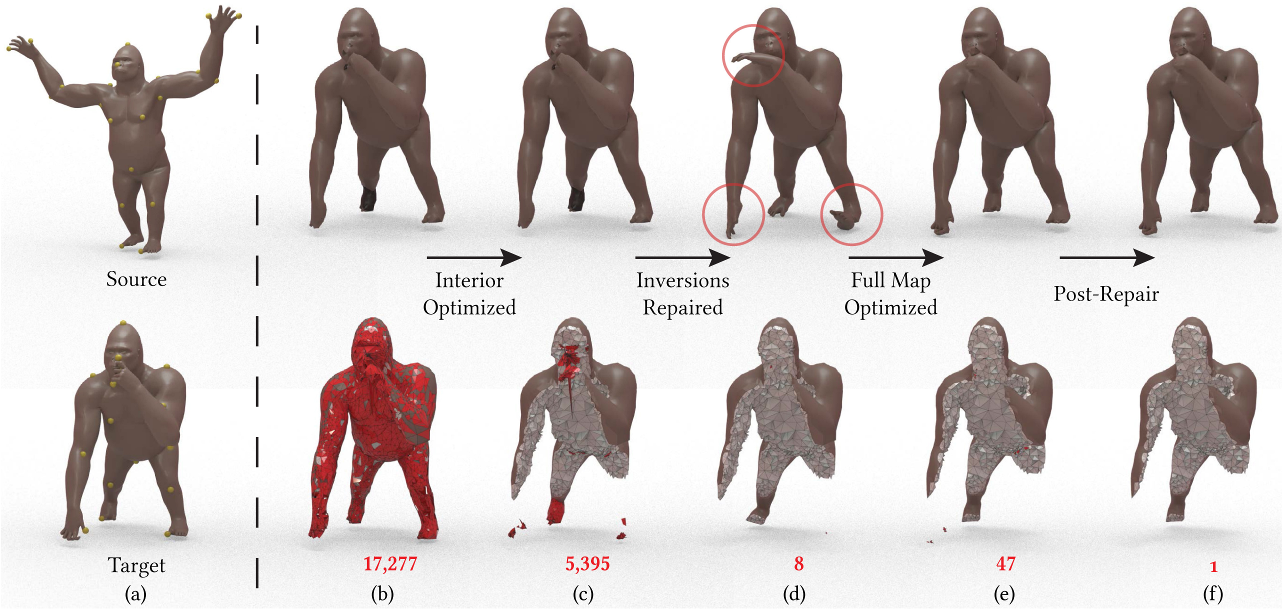


Fig. 7. Flowchart depicting each step of our method: (a) initial source and target shapes, with landmarks shown as yellow spheres; mapped shape; (b) at initialization; (c) after optimization converges while keeping the boundary fixed; (d) after tetrahedron inversion repair; (e) at convergence; and (f) after post-convergence tetrahedron repair. Top row shows the boundary of the mapped shape at every step, and the bottom row shows a cut through the interior, revealing interior tetrahedra. Inverted and collapsed tetrahedra are red. The number of inverted tetrahedra is listed under each cut-through mesh. Our initial map (b) has all interior tetrahedra collapsed to the boundary, resulting in 17,277 (46%) degenerate or flipped tetrahedra. Steps (c) and (d) optimize and repair the interior, resulting in eight flipped tetrahedra. The tetrahedron repair step restores elements of the map to match the source, as the hands and feet rotate. The final optimization followed by the post-convergence repair produces a map that closely matches the boundary with negligible inversions (one flipped tetrahedron).

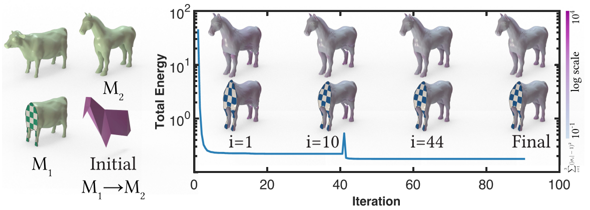


Fig. 8. Optimization of Equation (24) using a landmark initialization. Despite a coarse initialization, our algorithm approximates the target shape after one iteration. Further optimization decreases surface distortion and improves interior regularity, as visualized by the checkerboard patterns. At iteration 41, the inverted tetrahedron repair is performed, causing a jump in the projection energy E_P, from which our algorithm quickly recovers.

The gradient with respect to each vertex is found by gathering the gradients of each tetrahedron adjacent to that vertex.

Optimizing for P_{ij}. Fixing X_{12}, X_{21}, the remaining energy terms with respect to P_{ij} are of the form $\|P_{ij}A - B\|^2_{M_i}$ with $A \in \mathbb{R}^{n_j \times 6}, B \in \mathbb{R}^{n_i \times 6}$. Following Ezuz et al. [2019], this minimization can be understood as a *projection* problem solved independently for each row of P_{ij}.

In our case, we need to project the points in A to the 6-dimensional tetrahedral mesh with vertices B, whose connectivity is the same as M_j. The presence of several additional energy terms in our formulation also leads to a unique projection problem. Since the problem can be solved independently, we implement an efficient solution using CUDA programming. To enforce a hard boundary-to-boundary constraint, we map rows of A corresponding to the boundary of M_i to the boundary of the target embedding.

5.3 Inverted Tetrahedron Repair

The initial maps suggested in Section 5.1 are straightforward to compute, but they are quite distant from our desired output; indeed, the majority of tetrahedra in our initial maps have zero volume. Although alternating between the two steps above is guaranteed to decrease the objective function in each step, empirically, we find in the initial stages our algorithm can get stuck in local optima due to inverted elements. Here, we describe a heuristic strategy that empirically can improve the quality of our output.

In this tetrahedron repair step, we find all inverted tetrahedra. We then take the 1-ring neighborhood of the vertices in the inverted tetrahedra and use L-BFGS to minimize f_{ARAP} with the remaining vertices fixed.

5.4 Full Algorithm and Stopping Criteria

Overall, our optimization procedure follows four broad steps:

(1) map initialization (Section 5.1);
(2) optimization while keeping the boundary fixed (Section 5.2);
(3) inverted tetrahedron repair (Section 5.3);
(4) optimization of all vertices (Section 5.2); and
(5) post-convergence inverted tetrahedron repair (Section 5.3).

For stages 2 and 4, we set as our convergence criteria one of (i) the norm of the gradient $<10^{-6}$, (ii) the objective function

ALGORITHM 1: Coordinate descent with tetrahedra uninversion

```
Input: initial maps $P_{12}$, $P_{21}$
Output: optimized maps $X_{12}$, $X_{21}$, $P_{12}$, $P_{21}$

$\partial P_{12}^{(0)} \leftarrow P_{12}(\partial V_1, :)$ // initial boundary map
$\partial P_{21}^{(0)} \leftarrow P_{21}(\partial V_2, :)$
$X_{12} \leftarrow P_{12} V_2$ // initial vertex map
$X_{21} \leftarrow P_{21} V_1$

while !converged do // optimize boundary map
    for $(i, j) \in \{(1, 2), (2, 1)\}$ do
        $P_{ij} \leftarrow \arg\min_{P \in \mathcal{P}_{ij}} E_R[P, X_{ji}] + E_Q[P, X_{ij}]$
        $X_{ij} \leftarrow \arg\min_{X \in \mathbb{R}^{n_i \times 6}} E_{ARAP}[X_{ij}]$
                $+ E_R[X_{ij}, P_{ji}] + E_P[X_{ij}] + E_Q[X_{ij}, P_{ij}]$
        $\partial P_{ij} \leftarrow \partial P_{ij}^{(0)}$ // restore boundary

// inverted tetrahedron repair
idx $\leftarrow \det J(X_i^{T_k}) \leq 0, \forall T_k \in \mathcal{T}_i$ // find inverted tetrahedra
$X_{ij}(idx) \leftarrow \arg\min_{X \in \mathbb{R}^{n_i \times 6}} E_{ARAP}[X_{ij}(idx)]$ // 1-ring nbhd.

while !converged do // optimize full map
    for $(i, j) \in \{(1, 2), (2, 1)\}$ do
        $P_{ij} \leftarrow \arg\min_{P \in \mathcal{P}_{ij}} E_R[P, X_{ji}] + E_Q[P, X_{ij}]$
        $X_{ij} \leftarrow \arg\min_{X \in \mathbb{R}^{n_i \times 6}} E_{ARAP}[X_{ij}]$
                $+E_R[X_{ij}, P_{ji}] + E_P[X_{ij}] + E_Q[X_{ij}, P_{ij}]$
```

decreases by less than 10^{-7} between successive iterations, or (iii) run for 50 iterations; the third criterion is a fallback that rarely occurs in practice. For stage 5, we limit vertex displacement to preserve map quality by limiting to 100 steps of L-BFGS and we restrict optimization to only vertices in inverted tetrahedra.

Algorithm 1 summarizes our full procedure.

5.5 Implementation Details

Unless otherwise noted, all figures are generated using identical parameters. We use grid search to identify reasonable parameters; the results of our analysis are provided in Figure 6. We set the rigidity parameter $\alpha = 0.5$ and the boundary conformation parameter $\gamma = 25$, achieving a reasonable tradeoff between average distance to the target and maintaining per-tetrahedron map quality as measured using $\det \hat{J}$, the normalized Jacobian determinant. To find these values, we initialize $\beta = 0.25$ and increase linearly to $\beta = 5$ over 20 iterations. In practice, we found our method was insensitive to the choice of β.

We generate tetrahedral meshes using fTetWild [Hu et al. 2020]. Prior to mapping, we normalize each mesh to have volume 1. We perform one tetrahedron repair step, as we found negligible improvement after performing more.

We implement our method in MATLAB, using CUDA to optimize the projection step by extending the projection code in Li et al. [2021] to $\mathbb{R}^6$. Our code is available at https://github.com/mabulnaga/symmetric-volume-maps.

6 EXPERIMENTS

We measure map quality by assessing distortion and closeness to matching the target shapes (Section 6.1). We validate our method by mapping pairs of shapes from four datasets (Section 6.2) and report visualizations and numerical scores evaluating the result (Section 6.3). We also compare our method to several variants of a baseline mapping approach (Section 6.4). We test the robustness of our method in Section 6.5 and evaluate the choice of symmetrized energy on computing a map in Section 6.6.

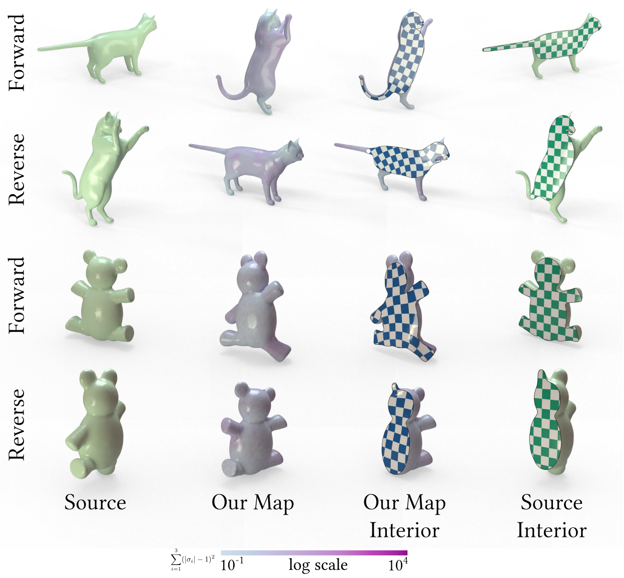


Fig. 9. Forward and reverse maps on related pairs of shapes. We observe smooth patterns of distortion on the boundary while capturing distinguishing geometric features, such as the transformation of the tail of the cat and movement of the bear's ears. Distortion is uniform throughout the interior.

6.1 Quality Metrics

We validate our method using the metrics outlined below.

Boundary matching. We measure fit to the target boundary using the Hausdorff distance $d_{\max}$ and the chamfer distance d_{avg} defined as follows:

$$d_{\max}(M_1, M_2) = \max\left\{ \sup_{\mathbf{x} \in M_1} \inf_{\mathbf{y} \in M_2} d(\mathbf{x}, \mathbf{y}), \sup_{\mathbf{y} \in M_2} \inf_{\mathbf{x} \in M_1} d(\mathbf{x}, \mathbf{y}) \right\} \tag{27}$$

$$d_{\text{avg}}(M_1, M_2) = \frac{1}{|\mathcal{V}_1| + |\mathcal{V}_2|} \left[\sum_{\mathbf{v}_i \in \mathcal{V}_1} d(\mathbf{v}_i, M_2) + \sum_{\mathbf{v}_j \in \mathcal{V}_2} d(\mathbf{v}_j, M_1) \right]. \tag{28}$$

Here, $\mathcal{V}_1$ and $\mathcal{V}_2$ denote the sets of vertices of M_1 and M_2, respectively. To make the measures above scale-independent, we normalize both quantities by the length of the diagonal of the bounding

Table 3. Map Quality Evaluation

Map (Initialization)	Time (min.)	E_R ($\times 10^{-3}$)	E_{ARAP} ($\times 10^{-3}$)	n_{inv}	$\hat{d}_{max}$ ($\times 10^{-2}$)	$\hat{d}_{avg}$ ($\times 10^{-2}$)	$\det \hat{J}$
X_{ij} (Surface)	31 ±21	1.47 ±1.9	81.7 ±78.5	7.7 ±9.1	2.5 ±1.2	0.10 ±0.046	0.98 ±0.02
P_{ij} (Surface)	31 ±21	1.29 ±1.65	134.5 ±115.4	649 ±549	1.9 ±0.78	0.072 ±0.028	0.96 ±0.04
X_{ij} (Landmark)	107 ±53	7.45 ±10.7	93.6 ±73.3	15.8 ±10.9	2.7 ±1.0	0.12 ±0.046	0.97 ±0.02
P_{ij} (Landmark)	107 ±53	6.67 ±9.7	176.6 ±145.4	723 ±515	2.6 ±1.0	0.11 ±0.038	0.94 ±0.04

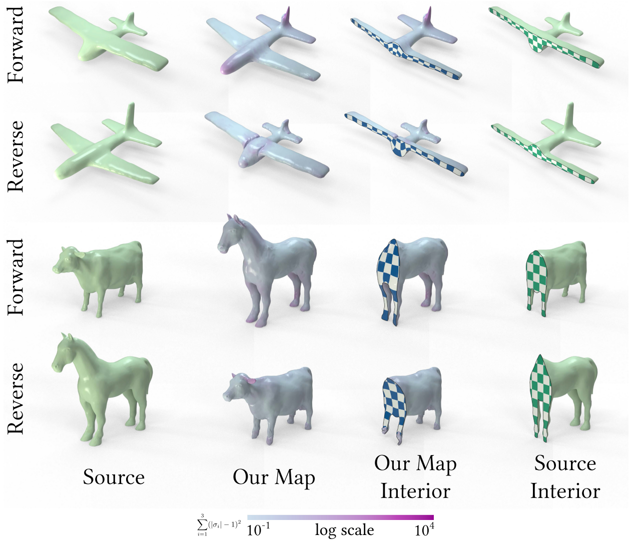


Fig. 10. Forward and reverse maps on far-from-isometric shapes. Our maps capture the extreme deformations, for example, by growing and collapsing the airplane rudder and deforming the ears of the horse and cow pair. Matching boundary features expectedly leads to high local distortion, as a large volume change is required to model these transformations. The checkerboard pattern reveals that regions with high boundary distortions also cause interior distortion (see airplane), but the computed maps are uniform and smooth elsewhere.

box enclosing the target mesh. We use hats to denote normalized quantities: $\hat{d}_{\max}$ and $\hat{d}_{\text{avg}}$.

To visualize the distortion in the interiors of tetrahedral meshes, we use a mapped checkerboard pattern. In each map visualization, using Houdini, we slice the source shape with a plane and place an extrinsic checkerboard pattern on the intersection, using rounding and modulo operations on coordinates. We push forward the planar intersection surface through our map and render the result using a custom shader that looks back to the corresponding coordinate in the source and evaluates the checkerboard function. Interpolation happens by finding the closest element (`xyzdist`) and then transferring coordinates (`primuv`).

Distortion and inversion. We measure the quality of the transformation by computing the number of inverted tetrahedra (n_{inv}) and the mean normalized Jacobian determinant det $\hat{J}$ (weighted by tetrahedron volume), where the columns of J are normalized as in Li et al. [2021]. Figures containing qualitative results depict distortion per tetrahedron using the ARAP energy $\sum_{i=1}^{3}(|\sigma_i|-1)^2$.

6.2 Datasets

We evaluate our method on 24 mesh pairs from four datasets. For datasets where only triangle meshes are available, we tessellate the interiors. We randomly select pairs of shapes distorted non-isometrically from the SHREC19 dataset [Dyke et al. 2019]. We also randomly select matching and non-matching pairs of humans and animals for nonrigid correspondence from the TOSCA dataset [Bronstein et al. 2008a]. Finally, we obtain tetrahedral meshes of models of natural objects and CAD models from Li et al. [2021] and Fu et al. [2016], from Thingi10k [Zhou and Jacobson 2016], and from Thingiverse [Japan 2022]. The resulting meshes had (mean ± standard deviation) 50,010 ± 34,663 tetrahedra. We manually choose landmarks on the boundary surfaces for every mapping example (marked on most figures); Table 5 provides the number of landmarks and number of tetrahedra for each pair.

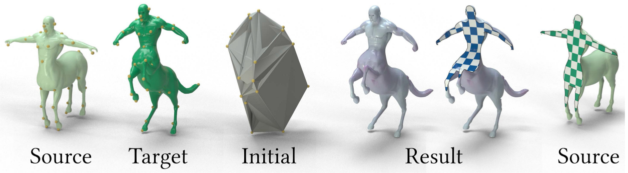


Fig. 11. Resulting map when initialized using only a sparse set of landmark points. Despite an initialization that collapses the mesh to a set of landmarks, we produce a map that captures sharp geometric features of the target including the hands and bends of the legs. The distortion is smooth and uniform throughout the boundary and interior.

6.3 Validation

In this section, we demonstrate our maps on several pairs.

Quantitative evaluation and map selection. Table 3 measures performance of both sets of maps, X_{ij} and P_{ij}, using surface map initialization and landmark initialization. Using the image of the map X_{ij}, we achieve close matchings to the target boundary with negligible tetrahedron inversions and while effectively maintaining tetrahedron quality. The landmark-based initialization achieves comparable performance, with slightly higher $\hat{d}_{max}$. These results indicate our method is robust to the choice of initialization. The constrained maps P_{ij} have significantly higher tetrahedron inversion due to the constraint $P_{ij} \in \mathcal{P}_{ij}$, which results in boundary tetrahedron foldovers. Since the boundary matching metrics are comparable for both maps, we use X_{ij} as the final map. The low number of tetrahedron inversions (n_{inv}) and small E_R indicate the resultant maps are nearly inverses of one another. Table 5 presents results for all pairs in our dataset.

Algorithm flowchart. We demonstrate each step of our algorithm in Figure 7. First, we compute an initial boundary map using the method of Ezuz et al. [2019]. This initial map is interpolated from the boundary to the interior by mapping each interior vertex to the target of its closest boundary vertex, as described in Section 5.1. This procedure results in a significant number of inverted or collapsed tetrahedra (Figure 7(b)). The interior is then improved by minimizing the map energy over the interior vertices (Figure 7(c)). Then, we repair inverted tetrahedra, dramatically reducing the number of flipped tetrahedra, as described in Section 5.3. The mapped mesh starts to restore its source pose; the hands and feet rotate (Figure 7(d)). We compute the final map by optimizing over all vertices (Figure 7(e)) and then perform post-convergence tetrahedron repair, arriving at a solution that closely conforms to the target boundary while minimizing distortion (Figure 7(f)).

Figure 8 visualizes our optimization routine initialized with landmarks. A few intermediate shapes are demonstrated. Our algorithm quickly recovers the target shape, and the optimization improves surface matching and reduces boundary and interior distortion.

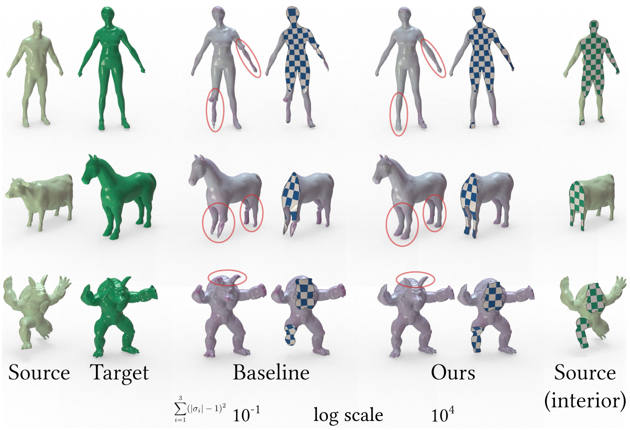


Fig. 12. Comparison of our map with the baseline approach using $K = 25$ with landmark equality constraints. Red ovals indicate distorted regions in the baseline where our method succeeds. Our approach effectively preserves geometric features and produces high-quality maps.

Map results. We demonstrate our method on several pairs. Figure 9 shows the forward and reverse maps between pairs of deformations from the same domain. In both examples, distortion is smooth throughout the boundary, and our map successfully matches geometric features, for example, the curved tail and the ears in the cat pairs. The checkerboard patterns demonstrate that our maps are smooth in the interior.

Figure 10 shows results for the more challenging problem of mapping between pairs of shapes from different domains. Distortions are mainly smooth on the boundary but are expectedly high in regions with large displacements, e.g., in the nose and rudder of the airplane in the forward direction. Here, the volume of the nose has to shrink substantially, while the rudder has to expand in height. Similarly, we see large distortion in the cow-horse pair, particularly in the ears in the reverse map and in the knees and feet in the forward map. Our boundary term yields maps that closely conform to the target at the cost of greater tetrahedral distortion.

Figure 11 demonstrates our resultant map when initialized using a sparse set of landmark points (Section 5.1, landmark-based initialization). While the initial map is unintelligible, our output matches the target shape closely. The final map has low distortion throughout the boundary and captures the narrow features of the target, including the fingers and bends in the legs. Furthermore, the checkerboard pattern reveals uniform distortion in the interior.

6.4 Baseline Comparison

We compare to the volumetric mapping approach of Kovalsky et al. [2015]. Their method inputs a surface map with optimized interior and computes a similar map that is orientation-preserving with bounded condition number K. Linear equality constraints on the vertices are used to fix parts of the map.

We compute the initial volumetric map by first computing a surface map as in Section 5.1 and then repairing degenerate tetrahedra by minimizing the Dirichlet energy while keeping the boundary fixed, as was done by Kovalsky et al. [2015]. We test four different sets of equality constraints for extracting the final volumetric maps: (1) fixing the boundary map; (2) fixing the boundary map for vertices not in inverted tetrahedra; (3) fixing landmarks; and (4) preserving center of mass. We use conformality bound $K \in \{5, 25, 50, 100\}$.

Table 4. Map Quality Comparison to the Baseline for $K = 25$.

	Constraint	n_{inv}	$\hat{d}_{max}$ ($\times10^{-2}$)	$\hat{d}_{avg}$ ($\times10^{-2}$)	det $\hat{J}$
	Ours	8 ± 13.8	**2.35 ± 1.45**	0.097 ± 0.05	**0.98 ± 0.02**
Baseline	Boundary	2,740 ± 2,210	2.84 ± 1.06	**0.085 ± 0.049**	0.82 ± 0.17
	Boundary (no flip)	11.1 ± 31.8	7.9 ± 8.5	0.33 ± 0.38	0.89 ± 0.1
	Landmark	1.8 ± 3.2	5.2 ± 3.1	0.7 ± 0.26	0.89 ± 0.11
	Center of mass	**1.7 ± 2.6**	7.2 ± 0.58	0.8 ± 1.2	0.89 ± 0.11

Table 4 compares map quality across the dataset for each equality constraint using $K = 25$. Similar behavior arose for other values of K, so they are not shown. We compare with the matching forward maps from our method. The fixed boundary map results in comparably low $\hat{d}_{\text{max}}$, $\hat{d}_{\text{avg}}$ to our method, but with a significantly greater number of flipped tetrahedra and poor map quality (det $\hat{J} = 0.82$) compared to our approach (det $\hat{J} = 0.98$). The strongest baseline uses the landmark equality constraints, resulting in improved n_{inv}, at the cost of map quality and boundary matching.

Figure 12 compares our map with the baseline using the fixed landmark constraint. Our method correctly maps features that are distorted by the baseline, such as the arm and leg of the human and hooves of the horse. The baseline approach performs well on the armadillo, a map between shapes of the same domain, but produces higher distortion. These visual and quantitative results demonstrate the strength in our free-boundary formulation, which effectively matches geometric features.

Surface map repair. Figure 13 shows how our algorithm recovers artifacts in the 2D surface map initialization procedure (Section 5.1) and compares with the baseline using the fixed landmark constraint. Starting from our landmarks, the surface map method of Ezuz et al. [2019] results in parts of the surface that are folded inside out (the backs of triangles are shown in black), as seen on the arms and legs of the human and the paws of the dog; the initial maps also have collapsed boundary features (hand of the human, tail of the dog). Both our method and the baseline target orientation-preserving maps and correct these inverted areas. Unlike the baseline, our method recovers from the inverted regions to match the target shape. Furthermore, we fill small regions such as the tail of the dog and the hands and feet of the human.

6.5 Map Robustness

We test the robustness of our method on challenging mapping cases. We first assess the ability to map from smooth, high-resolution shapes to coarse meshes with sharp features. Figure 14 demonstrates mapping to polycube shapes from Fu et al. [2016], using the P_{ij} maps. We successfully map bidirectionally between the smooth and coarse shapes, although expectedly higher distortion arises in the corner regions.

Figure 15 tests matching between nonisometric pairings. We stretch one arm and leg of the human mesh and obtain close matchings in both directions, although higher distortion arises at the ends of the stretched regions due to large changes in volume required to match to the target.

Figure 16 tests the robustness of our method to mesh quality. Figure 16 (top) maps a high-resolution horse to progressively

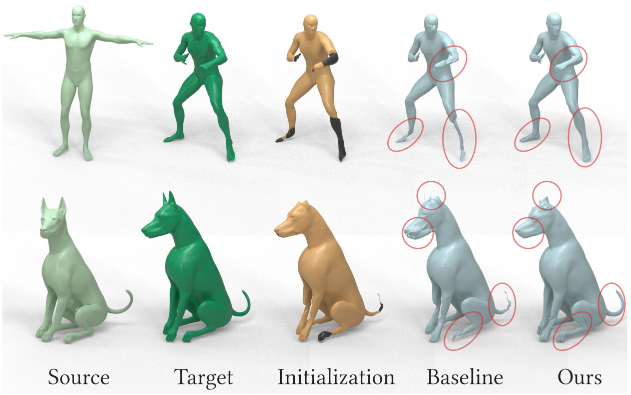


Fig. 13. Refinement of the initial boundary map using [Ezuz et al. 2019] and comparison to the baseline with landmark equality constraints. The backs of boundary triangles are shown in black. The initial map produces areas of the surface turned inside out and collapses regions such as the hands of the human and tail of the dog. Both our method and the baseline can produce orientation-preserving correspondences. Compared to the baseline, our approach restores collapsed and distorted regions and effectively matches the target shape (red ovals). This experiment also reveals that our method can recover from poor initialization.

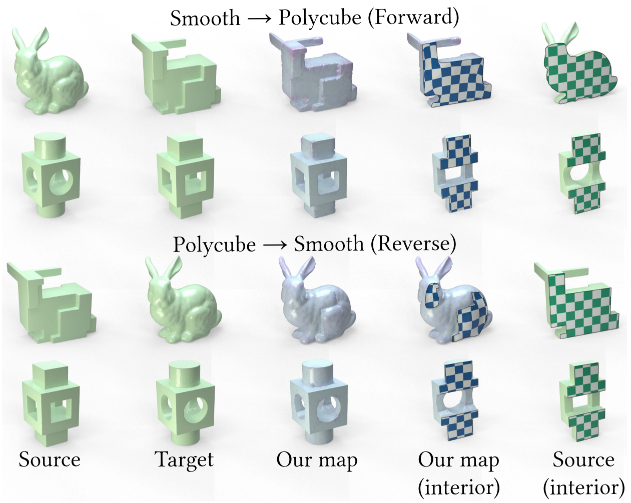


Fig. 14. Map between smooth and polycube shapes. Our method produces close matchings in both directions, though higher distortion arises in the corner regions of the polycube.

downsampled versions. Despite differences in mesh resolution, we successfully map to the target shapes with minimal inversions, although small features like the ears of the horse are distorted. This artifact is due to few tetrahedra representing these regions in the downsampled mesh. Figure 16 (bottom) assesses the sensitivity of our method to mesh quality by mapping a bird with thin, elongated tetrahedra faces to one with regular tetrahedra. We achieve a close matching, suggesting our method is robust to mesh quality.

6.6 Symmetrized Energy Choice

We experiment with the choice of symmetrized energy and its effect on producing a map. As described in Section 3.3, several

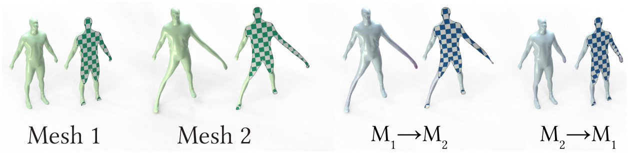


Fig. 15. Nonisometric mapping of a stretched human. Close matchings are obtained, though higher distortion arises in the stretched regions.

symmetrized energies do not favor isometry, while our choice, the sARAP energy, does. Figure 17 compares the output when optimizing using the sARAP, the symmetrized Dirichlet (sDir), and the 3rd-order symmetrized Dirichlet (sDir^3) energies. The 3rd-order Dirichlet is used, since tri-harmonic functions are used to achieve C^1 continuity in 3D [Iwaniec and Onninen 2010]. In these experiments, we remove the tetrahedron repair step, which made the artifacts worse. We compare two choices of γ and visualize the resultant maps.

Both the sDir and sDir^3 energy completely collapse the map for $\gamma = 0.1$, since the projection term has little effect at keeping the map intact. Similarly, parts of the mapped mesh degenerate with $\gamma = 25$. In both cases, the sDir^3 energy, however, maintains continuity. In contrast, the sARAP energy does not produce a collapsed map, although it starts to restore the source when $\gamma = 0.1$.

This experiment verifies our analysis in Section 3.2 and additionally shows that methods using energies that do not favor isometry can be sensitive to parameter choice.

7 EXAMPLES

Volumetric maps are useful for transporting data between domains. Below, we depict some use cases that would benefit from our low-distortion, near-diffeomorphic maps.

7.1 Internal Geometry Transfer

In contrast to pulling back functions on M_2 to M_1, we can also push forward maps into M_1 to M_2. This category of data includes point clouds, collections of curves, and arbitrary subdomains $U \subset M_1$.

As an example of how data can be easily transported using our maps, in Figure 18, we push forward integral curves of a frame field on domain M_1 through $\phi : M_1 \rightarrow M_2$. The

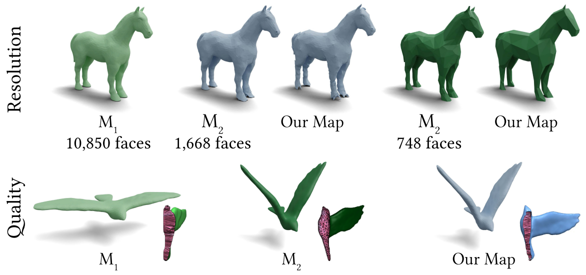


Fig. 16. Map sensitivity to mesh quality. Top: mapping a high-resolution horse mesh to progressively downsampled versions (boundary triangle faces indicated). Bottom: mapping a bird with thin, elongated tetrahedra faces to one with regular tetrahedra. In all cases, the targets are matched closely with few inversions (maximum of $n_{inv} = 2$), though in the horse examples, small geometric features, such as the ears, are lost due to limited representation.

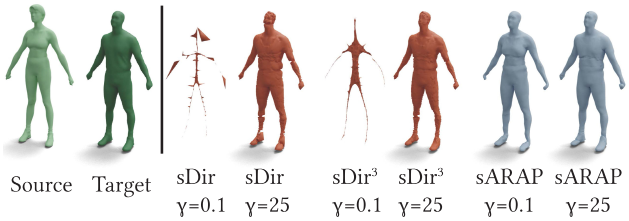


Fig. 17. Comparison of maps when optimizing with the symmetrized Dirichlet (sDir), the 3rd-order sDir^3, and the sARAP energies. sDir and sDir^3 produce collapsed maps for both values of γ, although $\gamma = 25$ keeps parts of the map intact, as it pushes vertices to the boundary. The sARAP energy does not collapse, but starts to show the source shape for $\gamma = 0.1$, as expected.

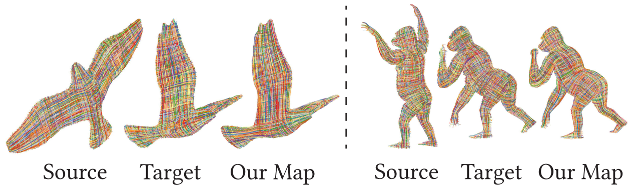


Fig. 18. When the integral curves of an octahedral frame field are pushed forward from a source domain (left) to a target domain (right), the result looks similar to the integral curves of a field computed directly on the target (center). The mapped curves remain nearly orthogonal, illustrating the low metric distortion of our map.

frame fields and their integral curves were generated using ARFF [Palmer et al. 2020]. Integral curves were pushed forward by mapping the curve vertices individually using piecewise linearity. The integral curves remain nearly orthogonal under the map, showing that it is close to isometric. The pushed-forward integral curves closely match the integral curves computed directly on M_2, also reflecting the map's degree of metric preservation.

In another example, we simulate an internal geometry transfer task. As shown in Figure 20, we place several objects representing anatomy inside of our source mesh and push these forward to our target. Despite rotation of the head and movement of the arm, structure is largely maintained. For the meshes used in this example, we credit Averin [2017], Leemhuis [2018], Medical [2013], Reininger [2015], and YEG 3D Printing [2015].

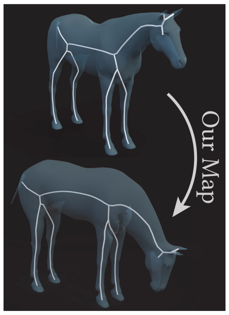


Fig. 19. Internal curve-skeleton transfer.

In a final example, we transfer a curve-skeleton of a horse mesh to our target (Figure 19). The source skeleton is generated using the approach of Cao et al. [2015]. The transferred skeleton captures the deformation of the horse, as evidenced by the curvature of the spine. Previous work has proposed skeleton transfer by finding a rigid transformation between skeletons of two surface meshes [Seylan and Sahillioğlu 2019]. In contrast, our volumetric approach facilitates internal geometry transfer and does not require computing matchings of internal shapes.

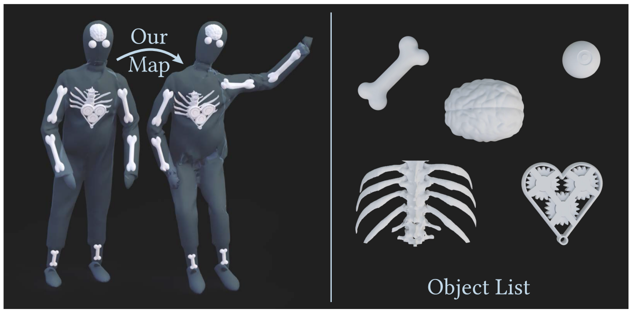


Fig. 20. Internal geometry transfer. We place several objects representing human anatomy in the interior of our source mesh and push these forward to the target using our volumetric map.

7.2 Hex Mesh Transfer

Our maps can transport other volumetric structures. Hexahedral meshing remains difficult and often requires extensive human intervention; our maps can transport expensive-to-compute hex meshes between domains. Figure 21 transports a hexahedral mesh designed using the method of Li et al. [2021] on one domain to a deformed domain. Similar to how we push forward integral curves, we transport a hex mesh by mapping its vertices individually, maintaining the combinatorial structure of the mesh. Due to the low metric distortion of the map, the distortion of most of the hexahedra remains low, as measured by the scaled Jacobian. However, the right foot of the mapped hex mesh has two toes joined together. This artifact is caused by projection to the wrong boundary target, an artifact also encountered by Li et al. [2021]; as their approach has user interaction, they suggest adding landmarks during the optimization to clarify difficult targets.

7.3 Volumetric Data Transfer

We demonstrate one example of volumetric data transfer using a dataset of placentas extracted from fetal MRI [Abulnaga et al. 2022]. The mapping is done on data from two patients. The first mapped pair contains two scans acquired where the mother is lying in two positions: supine and left lateral. The second contains two scans acquired ∼ 10 minutes apart. Figure 22 presents the results. The figure marks one important anatomical landmark, a cotyledon, which is responsible for the exchange of blood from the maternal side to the fetal side [Benirschke and Driscoll 1967]. Cotyledons appear as hyperintense circular regions in MRI. We observe close correspondence in the placental geometry. Similar patterns are seen in the mapped texture and the target. In this application, neither example has a clearly defined source or target shape. The symmetry in our method is advantageous for downstream tasks, such as statistical shape analysis or label propagation, as it prevents bias caused by arbitrarily selecting a source and a target. We leave to future work a detailed study of our method's relevance to MRI data.

8 DISCUSSION

We successfully map a collection of shapes of diverse geometry and demonstrate that our maps closely match the target boundary with low distortion throughout the volume and a negligible amount of flipped tetrahedra. Our method is robust to the choice of initialization (Figures 8, 11, and 13) and can produce a dense

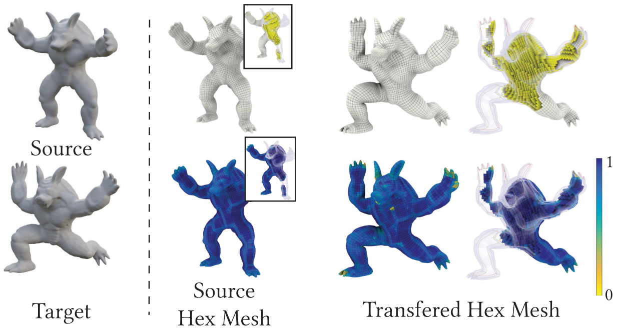


Fig. 21. Hex mesh pushed forward from one volume to another using our map. We observe low distortion, as measured by the scaled Jacobian overall, but there is some distortion in the mapped right foot. Hex meshes are visualized with HexaLab [Bracci et al. 2019], which clamps negative scaled Jacobian values to 0.

correspondence even when starting with a low-quality, many-to-one map (Figures 8 and 11). Compared to the baseline, our free boundary-based approach can recover from poor initialization (Figure 13) and produce higher-quality maps, as shown in Table 4 and Figure 12. Our examples illustrate scenarios that require a volumetric correspondence, namely, internal geometry transfer, hex mesh transfer, and volumetric data transfer.

Key to the development of our algorithm was the analysis of symmetric distortion energies in Sections 3.2–3.3. We symmetrized several common distortion energies and found that only the sARAP energy had the desirable properties of favoring isometry, preserving structure, and being nonsingular. We provide a simple way to symmetrize a distortion energy and check if it satisfies these properties. Figure 17 also shows that some choices of energy can lead to degenerate maps that are sensitive to the parameters used. The nonsingularity of the sARAP energy is favorable for computing a map given a degenerate initialization. Since volumetric correspondence has no obvious initializer, this property is key in our target applications, as we can recover from poor initialization. Future work remains in designing symmetric distortion energies that satisfy more desirable properties.

The connection between the theoretical analysis in Sections 3.2–3.3 to our algorithm design relies on $\psi = \phi^{-1}$. We use soft constraints to encourage a bijection and produce maps with low reversibility energy ($E_R = (1.47 \pm 1.9) \times 10^{-3}$) and few flipped tetrahedra (7.7 ± 9.9). In practice, we cannot guarantee $\psi = \phi^{-1}$, as our initialization is non-invertible, and the existence of an invertible map is not guaranteed. However, our experimental results demonstrate the theoretical analysis is relevant, as our computed maps favor isometries ($\det \hat{J} = 0.98 \pm 0.02$) and do not collapse (Figure 17). It remains an open problem to guarantee $\psi = \phi^{-1}$.

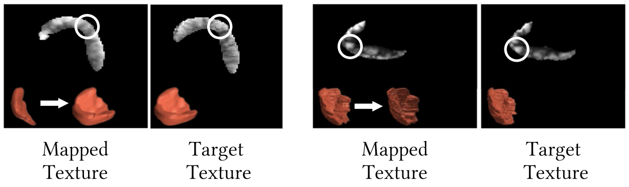


Fig. 22. Volumetric data transfer of two fetal MRI volumes visualized as cross-sections of 3D MRI. The figure shows texture transfer between two volumes in a scenario where the mother is lying in the supine and left lateral position *(left)* and in a scenario where the two volumes are approximately 10 minutes apart *(right)*. The circle marks the location of a cotyledon in the target texture.

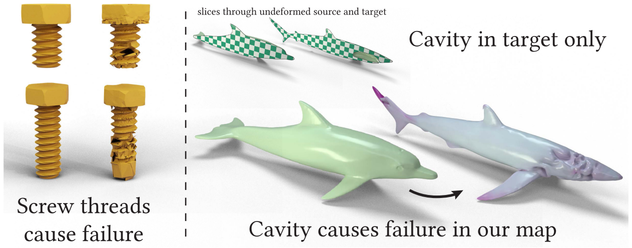

slices through undeformed source and target
Cavity in target only
Screw threads
cause failure
Cavity causes failure in our map

Fig. 23. Limitations. We were unable to map between the screw threads, as the map required removing or adding a large amount of material, leading to significant distortion. In our second example, the target shape, a shark, had a large cavity in its interior, while the source, a dolphin, did not.

8.1 Limitations

We observed a few failure cases, as can be seen in Figure 23. First, we encountered shapes where finding a volumetric map was simply infeasible. In the screw threads example, the required map would have to add or remove a large amount of volume, which would lead to substantial distortion. Furthermore, the threads on the boundary differ in number, making it impossible to match sharp features. In the second case, we were unable to map a shark with a cavity in its interior to a dolphin with a solid interior. The cavity is a large hollow area to which a volumetric approach is highly sensitive. Furthermore, our method is unable to change topology when mapping between shapes of different genus (Figure 24), and we are unable to prescribe topological constraints. Another limitation is that our method may not be suitable for partial volume matching, since we normalize input meshes to have volume 1. Last, as demonstrated in Figure 21, our method can join together small features in the boundary (e.g., armadillo toes). This artifact is caused by an incorrect boundary projection. A potential fix would be to have soft landmark constraints in the optimization.

Finally, our method takes between minutes and hours to compute the correspondences. The computational cost is problematic if desiring mapping a collection of shapes, despite our algorithm being advantageous in that we can map between shapes that are far-from-isometries, and we do not require the same connectivity between shapes. The computational bottleneck is computing the SVD for each tetrahedron many times on the CPU to approximate the gradient of the objective function. A future direction is to improve the convergence time by using a second-order method and to use the GPU for parallelization.

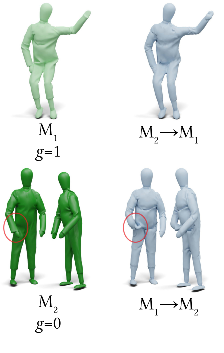


Fig. 24. Highly distorted region (red circle) when mapping from a genus-1 to a genus-0 shape.

Table 5. Quantitative Results on All Mesh Pairs in Our Dataset

Names		ℓ	Time	$\lvert\mathcal{T}_i\rvert$		$E_R(\times 10^{-3})$		$E_{ARAP}(\times 10^{-3})$		n_{inv}		$\hat{d}_{\max}(\times 10^{-2})$		$\hat{d}_{\text{avg}}(\times 10^{-2})$		$\lvert\det \hat{J}\rvert$	
scan_011	scan_019	23	29.23	36,420	43,527	2.23	1.52	55.89	64.91	3	11	1.47	7.82	0.12	0.24	0.98±0.05	0.974±0.0545
scan_011	scan_030	19	11.39	36,420	37,588	0.12	0.11	43.18	43.22	2	0	1.83	2.52	0.10	0.12	0.981±0.0372	0.984±0.0275
scan_019	scan_039	16	17.57	43,527	50,713	0.30	0.32	55.87	58.49	0	2	1.57	2	0.12	0.14	0.976±0.0503	0.979±0.0456
airplane1	airplane2	7	28	24,894	30,700	1.29	3.02	257.28	174.17	8	12	2.33	2.34	0.10	0.12	0.968±0.0773	0.954±0.0979
armadillo	deformed armadillo	21	75.89	81,114	113,794	0.11	0.11	25.40	28.22	2	3	1.15	1.39	0.05	0.06	0.99±0.0244	0.988±0.0259
bird_1	bird_2	12	19.55	30,361	90,810	0.17	0.18	45.16	44.36	0	7	1.23	1.15	0.06	0.08	0.987±0.0278	0.984±0.0318
cat0	cat1	17	9.86	17,867	22,988	0.51	0.45	40.33	51.52	0	1	1.98	1.73	0.08	0.08	0.984±0.0479	0.981±0.0467
cat4	cat5	18	14.05	25,985	22,710	0.92	1.02	52.71	56.03	5	5	3.92	2.89	0.13	0.11	0.98±0.0498	0.98±0.0496
centaur0	centaur1	37	13.70	30,357	26,954	0.42	0.38	26.06	29.56	1	3	1.04	1.42	0.05	0.07	0.993±0.0212	0.988±0.0299
dancer	dancer2	13	43.11	58,535	36,902	7.32	4.06	268.53	287.04	41	17	1.32	2.18	0.12	0.11	0.934±0.131	0.951±0.0909
dog4	dog5	27	17.04	31,469	30,160	1.58	2	61.45	53.25	7	3	2.29	2.27	0.09	0.10	0.979±0.0527	0.98±0.0538
dog6	dog7	24	33.92	26,739	43,771	3.87	4.34	137.22	121.09	3	14	7.01	2.38	0.20	0.17	0.961±0.0754	0.958±0.0863
dog7	dog8	25	71.15	81,145	85,128	0.34	0.35	22.03	22.51	4	9	2.86	4.78	0.08	0.12	0.992±0.031	0.992±0.0286
Dolphin	Shark	9	39.75	129,443	55,440	2.51	2.90	123.05	80.68	11	13	3.96	5.32	0.14	0.16	0.981±0.0576	0.981±0.0592
dragon_stand	dragonstand2	28	70.37	109,823	194,651	0.01	0.01	0.60	0.32	0	0	0.75	0.95	0.02	0.02	1±0.002	1±0.003
fish1	fish2	8	73.57	64,410	58,215	3.07	2.07	149.62	179.76	55	16	2.10	2.09	0.16	0.12	0.969±0.0872	0.969±0.0763
glass1	glass2	13	12.83	30,921	13,439	7.28	7.81	273.30	283.75	24	0	1.85	1.33	0.22	0.10	0.918±0.122	0.89±0.107
gorilla1	gorilla5	26	30.52	37,417	59,375	1.26	1.24	33.04	58.37	1	4	4.05	2.76	0.09	0.10	0.988±0.0326	0.978±0.0448
horse0	horse5	16	20.55	31,507	34,978	0.23	0.21	31.10	35.11	0	1	2.16	2.37	0.05	0.06	0.989±0.0414	0.99±0.027
Cow_t	Horse_t	21	20.09	31,694	32,515	0.58	1.16	117.74	129.02	17	18	2.25	3.43	0.13	0.23	0.978±0.0515	0.969±0.0641
human1	human2	21	42.27	56,550	82,581	0.57	0.87	58.34	60.89	0	33	1.67	2.09	0.06	0.10	0.988±0.0267	0.985±0.0428
michael0	michael7	20	19.53	19,445	30,014	0.40	0.35	21.62	27.28	1	4	1.75	1.46	0.06	0.07	0.992±0.0221	0.991±0.0269
seahorse2	seahorse4	22	8.24	13,720	15,667	0.09	0.11	16.89	17.14	1	1	1.10	1.91	0.04	0.04	0.993±0.0227	0.993±0.0213
toy1	toy2	12	31.54	75,236	62,880	0.43	0.47	47.36	53.13	5	3	4.74	3.66	0.08	0.07	0.987±0.0345	0.979±0.05

Our maps closely match the target boundaries while producing low distortion and few tetrahedron inversions. Here, ℓ denotes the number of landmarks. Time is measured in minutes.

8.2 Future Work

An exciting future direction is to develop application-specific volumetric correspondences. We provided a few examples of tasks where volumetric correspondence is useful. Our example of mapping MRI signals demonstrated that while matching geometries can improve correspondence, a method that incorporates both the geometry and signal intensities is needed. One framework could be to combine our vertex-based approach with functional maps.

We envision this work to be a starting point for dense volumetric correspondence applicable to a broad set of shapes. The nascent area of volumetric correspondence is largely unexplored, and our theoretical discussion suggests many intriguing mathematical questions and algorithmic design challenges.

ACKNOWLEDGMENTS

The authors would also like to thank David Palmer for insightful discussions and help with the frame field and hex mesh transfer experiments. We are grateful to Lingxiao Li for his help in extending the CUDA code for tetrahedron projection in $\mathbb{R}^6$. We would like to thank Shahar Kovalsky for his help in setting up the baseline code and Melinda Chen for her help in generating the curve-skeleton. Finally, we are grateful to Yu Wang, Zoë Marschner, Artem Lukoianov, and Ishita Goluguri for their help in proofreading.

REFERENCES

S. Mazdak Abulnaga, Esra Abaci Turk, Mikhail Bessmeltsev, P. Ellen Grant, Justin Solomon, and Polina Golland. 2022. Volumetric parameterization of the placenta to a flattened template. *IEEE Trans. Med. Imag.* 41, 4 (2022), 925–936.

Iman Aganj, Martin Reuter, Mert R. Sabuncu, and Bruce Fischl. 2015. Avoiding symmetry-breaking spatial non-uniformity in deformable image registration via a quasi-volume-preserving constraint. *NeuroImage* 106 (2015), 238–251.

Noam Aigerman and Yaron Lipman. 2013. Injective and bounded distortion mappings in 3D. *ACM Trans. Graph.* 32, 4 (2013), 106.

Noam Aigerman and Yaron Lipman. 2015. Orbifold Tutte embeddings. *ACM Trans. Graph.* 34, 6 (2015), 1–12.

Noam Aigerman and Yaron Lipman. 2016. Hyperbolic orbifold Tutte embeddings. *ACM Trans. Graph.* 35, 6 (2016), 217–1.

Noam Aigerman, Roi Poranne, and Yaron Lipman. 2014. Lifted bijections for low distortion surface mappings. *ACM Trans. Graph.* 33, 4 (2014), 1–12.

Noam Aigerman, Roi Poranne, and Yaron Lipman. 2015. Seamless surface mappings. *ACM Trans. Graph.* 34, 4 (2015), 1–13.

Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 1999. 3D shape histograms for similarity search and classification in spatial databases. In *International Symposium on Spatial Databases.* Springer, 207–226.

Brian B. Avants, Charles L. Epstein, Murray Grossman, and James C. Gee. 2008. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Med. Image Anal.* 12, 1 (2008), 26–41.

Eugeny Averin. 2017. Dog bone. Retrieved from https://www.thingiverse.com/thing:2090975.

M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes. 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* 61, 2 (2005), 139–157.

Kurt Benirschke and Shirley G. Driscoll. 1967. The pathology of the human placenta. In *Placenta.* Springer Berlin, 97–571.

Matteo Bracci, Marco Tarini, Nico Pietroni, Marco Livesu, and Paolo Cignoni. 2019. HexaLab.net: An online viewer for hexahedral meshes. *Comput.-aid. Des.* 110 (2019), 24–36.

Alon Bright, Edward Chien, and Ofir Weber. 2017. Harmonic global parametrization with rational holonomy. *ACM Trans. Graph.* 36, 4 (2017), 1–15.

Alexander M. Bronstein, Michael M. Bronstein, Alfred M. Bruckstein, and Ron Kimmel. 2008b. Analysis of two-dimensional non-rigid shapes. *Int. J. Comput. Vis.* 78, 1 (2008), 67–88.

Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. 2008a. *Numerical Geometry of Non-rigid Shapes.* Springer Science & Business Media.

Pascal Cachier and David Rey. 2000. Symmetrization of the non-rigid registration problem using inversion-invariant energies: Application to multiple sclerosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 472–481.

Marcel Campen, Cláudio T. Silva, and Denis Zorin. 2016. Bijective maps from simplicial foliations. *ACM Trans. Graph.* 35, 4 (2016), 1–15.

Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhixun Su. 2015. Point cloud skeletons via Laplacian-based contraction. In *IEEE Conference on Shape Modeling and Applications.*

Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4 (2010), 38:1–38:6.

Gary E. Christensen and Hans J. Johnson. 2001. Consistent image registration. *IEEE Trans. Med. Imag.* 20, 7 (2001), 568–582.

David Cohen and Mirela Ben-Chen. 2019. Generalized volumetric foliation from inverted viscous flow. *Comput. Graph.* 82 (2019), 152–162.

Anastasia Dubrovina and Ron Kimmel. 2011. Approximately isometric shape correspondence by matching pointwise spectral features and global geodesic structures. *Adv. Adapt. Data Anal.* 3 (2011), 203–228.

R. M. Dyke, C. Stride, Y.-K. Lai, P. L. Rosin, M. Aubry, A. Boyarski, A. M. Bronstein, M. M. Bronstein, D. Cremers, M. Fisher, T. Groueix, D. Guo, V. G. Kim, R. Kimmel,

Z. Lähner, K. Li, O. Litany, T. Remez, E. Rodolà, B. C. Russell, Y. Sahillioğlu, R. Slossberg, G. K. L. Tam, M. Vestner, Z. Wu, and J. Yang. 2019. Shape correspondence with isometric and non-isometric deformations. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association.

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. 2019. Reversible harmonic maps between discrete surfaces. *ACM Trans. Graph.* 38, 2 (2019), 1–12.

Michael S. Floater and Kai Hormann. 2005. Surface parameterization: A tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*. Springer Berlin, 157–186.

Nick C. Fox, Gerard R. Ridgway, and Jonathan M. Schott. 2011. Algorithms, atrophy and Alzheimer's disease: Cautionary tales for clinical trials. *Neuroimage* 57, 1 (2011), 15–18.

Xiao-Ming Fu, Chong-Yang Bai, and Yang Liu. 2016. Efficient volumetric polycube-map construction. *Comput. Graph. Forum (Paci. Graph.)* 35, 7 (2016).

Xiao-Ming Fu and Yang Liu. 2016. Computing inversion-free mappings by simplex assembly. *ACM Trans. Graph.* 35, 6 (2016).

Xiao-Ming Fu, Yang Liu, and Baining Guo. 2015. Computing locally injective mappings by advanced MIPS. *ACM Trans. Graph.* 34, 4 (2015), 71.

Xiao-Ming Fu, Jian-Ping Su, Zheng-Yu Zhao, Qing Fang, Chunyang Ye, and Ligang Liu. 2021. Inversion-free geometric mapping construction: A survey. *Computat. Visual Media* 7, 3 (2021), 289–318.

James Gain and Dominique Bechmann. 2008. A survey of spatial deformation from a user-centered perspective. *ACM Trans. Graph.* 27, 4 (2008), 1–21.

Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. 2021. Foldover-free maps in 50 lines of code. *ACM Trans. Graph.* 40, 4 (2021).

D. Geman and Chengda Yang. 1995. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Process.* 4, 7 (1995), 932–946.

Craig Gotsman, Xianfeng Gu, and Alla Sheffer. 2003. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graph.* 22, 3 (2003), 358–363.

Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro, and Michael Halle. 2000. Conformal surface parameterization for texture mapping. *IEEE Trans. Visualiz. Comput. Graph.* 6, 2 (2000), 181–189.

Joel Hass and Patrice Koehl. 2017. Comparing shapes of genus-zero surfaces. *J. Appl. Computat. Topol.* 1, 1 (2017), 57–87.

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast tetrahedral meshing in the wild. *ACM Trans. Graph.* 39, 4 (2020).

Xue Hua, Boris Gutman, Christina P. Boyle, Priya Rajagopalan, Alex D. Leow, Igor Yanovsky, Anand R. Kumar, Arthur W. Toga, Clifford R. Jack, Norbert Schuff, Gene E. Alexander, Kewei Chen, Eric M. Reiman, Michael W. Weiner, and Paul M. Thompson. 2011. Accurate measurement of brain changes in longitudinal MRI scans using tensor-based morphometry. *NeuroImage* 57, 1 (2011), 5–14.

Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J. Guibas. 2008. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1449–1457.

Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. 2004. Invertible finite elements for robust simulation of large deformation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 131–140.

Tadeusz Iwaniec and Jani Onninen. 2010. Deformations of finite conformal energy: Existence and removability of singularities. *Proc. London Math. Societ.* 100, 1 (2010), 1–23.

Varun Jain, Hao Zhang, and Oliver Van Kaick. 2007. Non-rigid spectral correspondence of triangle meshes. *Int. J. Shape Model.* 13, 1 (2007), 101–124.

Yahoo! Japan. 2022. Yahoo! Japan Thingiverse. Retrieved from https://www.thingiverse.com/yahoojapan/designs.

Sarang Joshi, Brad Davis, Matthieu Jomier, and Guido Gerig. 2004. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage* 23 (2004), S151–S160.

Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended intrinsic maps. *ACM Trans. Graph.* 30, 4 (2011), 1–12.

Stefan Klein, Marius Staring, and Josien P. W. Pluim. 2007. Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-splines. *IEEE Trans. Image Process.* 16, 12 (2007), 2879–2890.

P. Knupp. 1995. Mesh generation using vector fields. *J. Comput. Phys.* 119, 1 (1995), 142–148.

Shahar Z. Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. 2014. Controlling singular values with semidefinite programming. *ACM Trans. Graph.* 33, 4 (2014), 68–1.

Shahar Z. Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. 2015. Large-scale bounded distortion mappings. *ACM Trans. Graph.* 34, 6 (2015), 191–1.

Vladislav Kraevoy and Alla Sheffer. 2004. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graph.* 23, 3 (2004), 861–869.

Sing Chun Lee and Misha Kazhdan. 2019. Dense point-to-point correspondences between genus-zero shapes. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 27–37.

Jim Leemhuis. 2018. Programmable animatronic eyeball mechanism demo. Retrieved from https://www.thingiverse.com/thing:2857377.

Alex Leow, Sung-Cheng Huang, Alex Geng, James Becker, Simon Davis, Arthur Toga, and Paul Thompson. 2005. Inverse consistent mapping in 3D deformable image registration: Its construction and statistical properties. In *Biennial International Conference on Information Processing in Medical Imaging*. Springer, 493–503.

John P. Lewis, Matt Cordner, and Nickson Fong. 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *27th Annual Conference on Computer Graphics and Interactive Techniques*. 165–172.

Lingxiao Li, Paul Zhang, Dmitriy Smirnov, S. Mazdak Abulnaga, and Justin Solomon. 2021. Interactive all-hex meshing via cuboid decomposition. *ACM Trans. Graph.* 40, 6 (2021), 1–17.

Xin Li and S. S. Iyengar. 2014. On computing mapping of 3D objects: A survey. *ACM Comput. Surv.* 47, 2 (2014), 1–45.

Yaron Lipman and Thomas Funkhouser. 2009. Möbius voting for surface correspondence. *ACM Trans. Graph.* 28, 3 (2009), 1–12.

Roee Litman and Alexander M. Bronstein. 2013. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 1 (2013), 171–180.

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A local/global approach to mesh parameterization. *Comput. Graph. Forum* 27, 5 (2008), 1495–1504.

Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. 2017. Variance-minimizing transport plans for inter-surface mapping. *ACM Trans. Graph.* 36, 4 (2017), 1–14.

Diana Mateus, Radu Horaud, David Knossow, Fabio Cuzzolin, and Edmond Boyer. 2008. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Prevue Medical. 2013. Lower spine and ribcage from chest CT. Retrieved from https://www.thingiverse.com/thing:54669.

Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. 2002. Stable real-time deformations. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 49–54.

Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. 2006. Physically based deformable models in computer graphics. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, 809–836.

Francisco P. M. Oliveira and Joao Manuel R. S. Tavares. 2014. Medical image registration: A review. *Comput. Meth. Biomech. Biomed. Eng.* 17, 2 (2014), 73–93.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph.* 31, 4 (2012), 1–11.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. 2016. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*. 1–60.

Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. 2010. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1555–1564.

Gilles-Philippe Paillé and Pierre Poulin. 2012. As-conformal-as-possible discrete volumetric mapping. *Comput. Graph.* 36, 5 (2012), 427–433.

David Palmer, David Bommes, and Justin Solomon. 2020. Algebraic representations for volumetric frame fields. *ACM Trans. Graph.* 39, 2 (2020).

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable locally injective mappings. *ACM Trans. Graph.* 36, 4 (2017).

Urban Reininger. 2015. Geared heart. Retrieved from https://www.thingiverse.com/thing:662447.

Mert R. Sabuncu, B. T. Yeo, Koen Van Leemput, Tom Vercauteren, and Polina Golland. 2009. Asymmetric image-template registration. In *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 565–573.

Yusuf Sahillioğlu. 2020. Recent advances in shape correspondence. *Visual Comput.* 36, 8 (2020), 1705–1721.

Yusuf Sahillioğlu and Ladislav Kavan. 2015. Skuller: A volumetric shape registration algorithm for modeling skull deformities. *Med. Image Anal.* 23, 1 (2015), 15–27.

Samuele Salti, Federico Tombari, and Luigi Di Stefano. 2014. SHOT: Unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.g* 125 (2014), 251–264.

Patrick Schmidt, Janis Born, Marcel Campen, and Leif Kobbelt. 2019. Distortion-minimizing injective maps between surfaces. *ACM Trans. Graph.* 38, 6 (2019), 1–15.

John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2004. Inter-surface mapping. *ACM Trans. Graph.* 23, 3 (2004), 870–877.

M. M. Selim and R. P. Koomullil. 2016. Mesh deformation approaches—A survey. *J. Phys. Math.* 7, 2 (2016), 1–9.

Çağlar Seylan and Yusuf Sahillioğlu. 2019. 3D skeleton transfer for meshes and clouds. *Graphic. Mod.* 105 (2019), 101041.

Alla Sheffer, Emil Praun, and Kenneth Rose. 2007. Mesh parameterization methods and their applications. *Found. Trends Comput. Graph. Vis.* 2, 2 (2007), 105–171.

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z. Kovalsky, and Yaron Lipman. 2017. Geometric optimization via composite majorization. *ACM Trans. Graph.* 36, 4 (2017), 1–11.

Daniel Sieger, Stefan Menzel, and Mario Botsch. 2015. On shape deformation techniques for simulation-based design optimization. In *New Challenges in Grid Generation and Adaptivity for Scientific Computing*. Springer, 281–303.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean flesh simulation. *ACM Trans. Graph.* 37, 2 (2018), 1–15.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic eigensystems for isotropic distortion energies. *ACM Trans. Graph.* 38, 1 (2019), 1–15.

Jason Smith and Scott Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Trans. Graph.* 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. Wiley Online Library, 1617–1626.

Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. Entropic metric alignment for correspondence problems. *ACM Trans. Graph.* 35, 4 (2016), 1–13.

Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. 2013. Deformable medical image registration: A survey. *IEEE Trans. Med. Imag.* 32, 7 (2013), 1153–1190.

Oded Stein, Jiajin Li, and Justin Solomon. 2021. A splitting scheme for flip-free distortion energies. arXiv:2107.05200 [cs.GR].

Jian-Ping Su, Xiao-Ming Fu, and Ligang Liu. 2019. Practical foldover-free volumetric mapping construction. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 287–297.

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A survey on shape correspondence. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 1681–1707.

Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel et al. 2017. Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision*. IEEE, 517–526.

Max A. Viergever, J. B. Antoine Maintz, Stefan Klein, Keelin Murphy, Marius Staring, and Josien P. W. Pluim. 2016. A survey of medical image registration–under review. *Med. Image Anal.* 33 (Oct. 2016), 140–144.

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. 2008b. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Trans. Graph.* 27, 1 (2008), 1–18.

Yalin Wang, Xianfeng Gu, Shing-Tung Yau et al. 2003. Volumetric harmonic map. *Commun. Inf. Syst.* 3, 3 (2003), 191–202.

Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. 2008a. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Imag. Sci.* 1, 3 (2008), 248–272.

Jiazhi Xia, Ying He, Xiaotian Yin, Shuchu Han, and Xianfeng Gu. 2010. Direct-product volumetric parameterization of handlebodies via harmonic fields. In *Shape Modeling International Conference*. IEEE, 3–12.

YEG 3D Printing. 2015. Frankenstein's monster with removable brain. Retrieved from https://www.thingiverse.com/thing:1081535.

Mei-Heng Yueh, Tiexiang Li, Wen-Wei Lin, and Shing-Tung Yau. 2019. A novel algorithm for volume-preserving parameterizations of 3-manifolds. *SIAM J. Imag. Sci.* 12, 2 (2019), 1071–1098.

Paul A. Yushkevich, Brian B. Avants, Sandhitsu R. Das, John Pluta, Murat Altinay, Caryne Craige, Alzheimer's Disease Neuroimaging Initiative et al. 2010. Bias in estimation of hippocampal atrophy using deformation-based morphometry arises from asymmetric global normalization: An illustration in ADNI 3T MRI data. *Neuroimage* 50, 2 (2010), 434–445.

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A dataset of 10,000 3D-printing models. *arXiv preprint arXiv:1605.04797* (2016).

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.* 23, 4 (1997), 550–560.

Daniel Zoran and Yair Weiss. 2011. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*. IEEE, 479–486.

Received 13 June 2022; revised 28 September 2022; accepted 12 November 2022