



HAL
open science

Functional Characterization of Intrinsic and Extrinsic Geometry

Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, Maks Ovsjanikov

► **To cite this version:**

Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, Maks Ovsjanikov. Functional Characterization of Intrinsic and Extrinsic Geometry. ACM Transactions on Graphics, 2017, 17. hal-01741949

HAL Id: hal-01741949

<https://hal.science/hal-01741949v1>

Submitted on 23 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Functional Characterization of Intrinsic and Extrinsic Geometry

ETIENNE CORMAN*

LIX, École Polytechnique

JUSTIN SOLOMON*

Massachusetts Institute of Technology

MIRELA BEN-CHEN

Technion — Israel Institute of Technology

LEONIDAS GUIBAS

Stanford University

MAKS OVSJANIKOV

LIX, École Polytechnique

We propose a novel way to capture and characterize distortion between pairs of shapes by extending the recently proposed framework of shape differences built on functional maps. We modify the original definition of shape differences slightly and prove that, after this change, the discrete metric is fully encoded in two shape difference operators and can be recovered by solving two *linear* systems of equations. Then, we introduce an extension of the shape difference operators using offset surfaces to capture extrinsic or embedding-dependent distortion, complementing the purely intrinsic nature of the original shape differences. Finally, we demonstrate that a set of four operators is *complete*, capturing intrinsic and extrinsic structure and fully encoding a shape up to rigid motion in both discrete and continuous settings. We highlight the usefulness of our constructions by showing the complementary nature of our extrinsic shape differences in capturing distortion ignored by previous approaches. We additionally provide examples where we recover local shape structure from the shape difference operators, suggesting shape editing and analysis tools based on manipulating shape differences.

CCS Concepts: •Computing methodologies → Shape analysis;

Additional Key Words and Phrases: Shape differences, Laplacian, embedding, triangle mesh

ACM Reference Format:

Etienne Corman,* Justin Solomon,* Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. 2016. Functional characterization of intrinsic and extrinsic geometry. *ACM Trans. Graph.* VOLUME, NUMBER, Article NUMBER (MONTH 2016), 17 pages.
DOI: XXXX

1. INTRODUCTION

One classic approach to comparing surfaces separates metrics of similarity into *intrinsic* and *extrinsic* measurements. Intrinsic quantities are those that can be expressed exclusively in terms of distances

*Equal contribution

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2016 ACM. \$15.00
DOI: XXXX

along the surface, whereas extrinsic quantities are those that must be defined using surface normal vectors and/or an embedding into space. A crowning result of classical differential geometry describes local geometry in terms of two quantities: the first and second fundamental forms, which capture the intrinsic Gaussian and extrinsic mean curvatures, respectively [Bonnet 1867].

Considerable research in geometry processing has been dedicated to measuring intrinsic and extrinsic curvature in an attempt to replicate this attractive characterization of shape. From a practical standpoint, however, this task remains challenging for potentially noisy or irregular meshes considered in geometry processing. After all, surface curvature is a second-derivative quantity whose approximation on a piecewise-linear mesh requires discretization and mollification to deal with noise. Measurement of curvature aside, algorithms for recovering geometry from discrete curvatures remain difficult to formulate for many discretizations.

In this paper, we formulate an alternative characterization of surface geometry suited for analysis, comparison, and synthesis tasks in the discrete setting. Several desiderata inform our design; a suitable framework for representing shape should

- capture and distinguish intrinsic and extrinsic geometry,
- express shape properties in a multiscale fashion to distinguish noise and fine-scale detail from large-scale structure,
- come from a smooth theory of shape,
- provide insensitivity to tessellation,
- be naturally expressible on continuous surfaces and on triangle mesh discretizations, and
- admit an inverse operator for reconstructing the embedded shape.

In short, we wish to pass from pointwise embeddings to a “dual” space featuring a more democratic treatment of intrinsic and extrinsic shape properties. The themes of multiscale, dual expressions of correspondence is a central tenet of the recent *functional maps* framework [Ovsjanikov et al. 2012], which we aim to leverage in our work.

We approach this task by extending the theory of *shape differences*, introduced by Rustamov et al. [2013] for purely intrinsic comparisons of shape structure. Rather than defining a shape in isolation, their construction characterizes shape by considering the distortion or difference of the target shape from a fixed source shape given a functional map between them [Ovsjanikov et al. 2012]. Shape differences are couched in the language of functional analysis, indirectly measuring changes in angles and distances through

the effects of these changes on inner products of functions and their gradients, which allows for straightforward discretization via piecewise-linear finite elements (FEM). They also naturally enjoy a multi-scale property, via truncation to a reduced basis, such as the Laplace–Beltrami eigenfunctions. This property is particularly useful in distinguishing features at different levels of detail via simple multi-scale basis computation and projection.

We modify and extend this framework to derive a shape representation that is complete, encoding both the intrinsic and the extrinsic distortion without loss of information in both the continuous and discrete cases. To this end, we begin by reexamining the discretization of shape differences on triangle meshes. We modify the original definition of discrete area-based shape difference and prove an analog of a continuous property mentioned in [Rustamov et al. 2013] that shape differences fully capture intrinsic structure. Inspired by this fully-discrete result, we proceed to ask whether shape differences also can capture extrinsic structure. Toward this goal, we define an additional pair of shape differences on a thickened surface that captures extrinsic geometry.

We show that our full set of differences is sufficient to reconstruct a shape up to rigid motion in the discrete setting, under mild assumptions. This provides an *invertibility* property: We can take different embeddings of a mesh with fixed topology in \mathbb{R}^3 , convert those embeddings to shape differences, and convert back to embeddings without losing any information. We additionally provide convex techniques for recovering geometric information like triangle areas and edge lengths in the case that a shape difference is truncated to low-frequency components or noisy. In this case, the inverse shape-difference-to-shape-embedding problem is underconstrained, but we show empirically that our machinery recovers reasonable estimates.

To summarize, our main contributions are:

- Theoretical discussion establishing that properly modified shape difference operators from [Rustamov et al. 2013] fully encode the intrinsic metric of a triangle mesh. Notably, these operators enjoy a direct connection to smooth theory—providing some degree of tessellation invariance—as well as multiscale approximation in the Laplace–Beltrami basis.
- A novel set of shape differences aimed at capturing and characterizing *extrinsic* or embedding-dependent information, with an associated observation that generically this set of shape differences is complete and encodes shapes up to rigid motion.
- A set of approaches for recovering geometric structure and an embedding from the shape differences with theoretical guarantees of recovery in the presence of complete information, which we apply to shape editing operations based on manipulating shape differences.

We demonstrate the usefulness of these contributions on a variety of tasks, ranging from the exploration of cloth simulation data using our novel extrinsically-sensitive shape difference operators to the accurate transfer of intrinsic functions like geodesic distances, and finally to recovering shape embedding even in the presence of approximate or truncated functional correspondences.

2. RELATED WORK

Representation and manipulation of extrinsic and intrinsic structure is a vast theme pervading the geometry processing literature. We refer to [Botsch et al. 2010] for discussion of the basic questions of representation and interaction with continuous differential geometry. Here, we highlight research linked to our particular approach.

Functional maps. We study relationships between surfaces through the lens of *functional maps*, introduced in [Ovsjanikov et al. 2012]. Functional maps express maps between surfaces through linear operators transporting functions on one surface to functions on another. Beyond the technique proposed in the original paper, many algorithms exist for computing functional maps, e.g., via sparsity [Pokrass et al. 2013], joint diagonalization [Kovnatsky et al. 2013], consistency [Huang and Guibas 2013], supervised learning [Corman et al. 2014], matrix completion [Kovnatsky et al. 2015], or estimation from a point-to-point map [Corman et al. 2015].

Our goal of using functional maps to characterize local and global geometry builds upon the machinery of *shape differences* [Rustamov et al. 2013]; see §4 for a summary. Rustamov and colleagues [2013] show that in the case of smooth surfaces, shape differences fully encode intrinsic geometry. They do not, however, pursue a corresponding analysis for the discrete case. Furthermore, their work focuses solely on intrinsic geometry and hence cannot characterize extrinsic bending, critical for describing differences between nearly-isometric shapes like articulated bodies and cloth.

Shape-from-Laplacian. Recovering structure from intrinsic shape differences is closely linked to recovering structure from Laplacian operators. Both in the continuous [Rosenberg 1997] and discrete [Zeng et al. 2012] cases, the Laplace–Beltrami operator fully encodes intrinsic surface geometry, namely the Riemannian metric for smooth manifolds and edge lengths for discrete meshes. For triangle meshes, de Goes and colleagues [2014] provide convex machinery for recovering the intrinsic structure of the mesh; their encoding of intrinsic structure using only Laplacian matrices is more compact than our pair of area and conformal shape differences, at the cost of a nonlinear objective sensitive to incomplete information.

The theoretical and practical contributions proposed in this paper provide considerable insight beyond the fundamental mathematical contributions in these other works. Specifically, the convex optimizations in [Zeng et al. 2012; de Goes et al. 2014] operate in the case of *complete, noise-free information*. They cannot be used for projection-style problems, e.g. finding the closest set of edge lengths to a noisy input Laplacian approximation or to finding an intrinsic structure *consistent* with a truncated spectral approximation of the full operator. Additionally, we show how to use related machinery to encode extrinsic bending rather than only edge lengths.

Encoding extrinsic geometry. A natural question is whether intrinsic structure can be used to reconstruct a surface embedding up to a global rigid transformation. Numerous examples of isometric smooth surface pairs disprove this notion in the continuous case [Kreyszig 1959]. While exact isometries of triangle meshes are rare for surfaces without boundary with the exception of inward/outward “popping” of valence-three vertices, near-isometries can often arise and have significant differences in the embedding, making shape recovery from intrinsic data like edge lengths a numerically ill-conditioned problem; these near-isometries appear because small variations in the input edge lengths can lead to large changes in the resulting embedding. Nevertheless, Boscaini and colleagues [2015] provide an algorithm for recovering a surface embedding in \mathbb{R}^3 from shape differences or equivalent structures. They apply the SMACOF algorithm [Leeuw et al. 1977] for multidimensional scaling to generate an extrinsic embedding that replicates shape differences in a least-squares sense. As an alternative, [Panozzo et al. 2014] propose an algorithm for embedding from local approximations of the metric tensor; we will use an extension of this algorithm in §7.3. Both of these methods, how-

ever, operate using only intrinsic information and are subject to the ambiguity and instability caused by isometry invariance.

Adding extrinsic information to a shape representation allows it to be embedded in \mathbb{R}^3 up to rigid motion. In theory, the Gauss–Codazzi equations fully characterize surfaces from the first and second fundamental forms [Bonnet 1867] (see [Carmo 1976, pg. 236]). In geometry processing, [Eigensatz et al. 2008] reconstructs surfaces from prescribed principal curvatures, while [Fröhlich and Botsch 2011] use nonlinear optimization methods to recover shape from dihedral edge lengths. These methods and many subsequent techniques employ nonlinear least-squares fits with few guarantees or characterization of their behavior. Wang, Liu, and Tong [2012] propose a linear technique for embedding meshes from their edge lengths, dihedral angles, and axes of rotation across mesh faces.

In this paper, we make use of offset surfaces to introduce extrinsic information to the shape difference representation. Offset surfaces have appeared in geometry processing for some related tasks, including cage generation [Ben-Chen et al. 2009] and shape optimization for printing [Musialski et al. 2015]. While techniques like [Jung et al. 2004] are needed to generate “clean” offset surfaces for geometry editing purposes, in our case self-intersection and related artifacts are acceptable since the offset surface is not used for display but rather for geometric computation. [Cohen-Steiner and Morvan 2003; Hoffmann et al. 2014] provide curvature theories for discrete surfaces using offset geometry.

3. OVERVIEW

Our two main goals are to modify and extend the definition of the shape difference operators of Rustamov [2013] so as to capture extrinsic distortion and to facilitate shape inference, i.e. to recover the metric and potentially the embedding of a target shape, given a base shape and a collection of shape differences.

We achieve these goals in several stages. The main ingredient for constructing both smooth and discretized shape differences is the computation of inner products between functions. So, rather than working directly with shape differences, we largely focus on matrices of functional inner products, which can be constructed on a single shape rather than a pair. Hence, after reviewing the smooth construction of shape differences (§4.1), we reexamine the discretization of intrinsic inner products on triangle meshes and show how a simple modification of the area-based inner product fully encodes intrinsic geometry in an easily-inverted fashion (§4.2).

We then capture extrinsic shape structure by introducing two operators built from intrinsic inner products on offset surfaces of a base shape (§5). We accompany our construction with theoretical characterization of the new information provided by extrinsic products (§5.3) and conclude by making explicit how our constructions involving inner product matrices apply to the construction of differences *between* shapes (§6). In this section, we also consider how truncating shape differences written in the Laplace–Beltrami basis affects the linear systems we pose.

With our new definitions and analysis in place, we propose optimization procedures for recovering intrinsic and extrinsic shape structure from the shape difference operators, potentially expressed in a reduced basis (§7). While the basic machinery for recovering metric information from shape differences is purely linear, we propose the use of more general convex optimization tools that add resilience to noise and incomplete information by explicitly enforcing the triangle inequality and/or smoothness. We conclude by demonstrating the ability of our constructions to capture and characterize extrinsic distortion ignored by previous approaches (§8.1). We furthermore apply our methods to recovering the metric and

shape embedding and to facilitating novel shape editing operations via manipulating shape difference operators (§8.4).

4. STRUCTURE OF DISCRETE INNER PRODUCTS

By examining the derivation of formulas for computing shape differences, we can reveal how they are related to local surface geometry. This analysis not only elucidates the information encoded in a given shape difference but also will inform our design of algorithms for recovering shape embeddings from shape differences.

4.1 Smooth Shape Differences

In the continuous case, Rustamov et al. [2013] consider two inner products between functions $f, g \in C^\infty(M)$ defined over a two-dimensional surface $M \subset \mathbb{R}^3$:

$$\langle f, g \rangle_A^M := \int_M f(x)g(x) d\mu(x)$$

$$\langle f, g \rangle_C^M := \int_M \langle \nabla f(x), \nabla g(x) \rangle d\mu(x).$$

This pair of operators is preserved if and only if M undergoes isometric deformation, showing that they fully determine the intrinsic geometry of M . The product $\langle \cdot, \cdot \rangle_A^M$ is the “area-based inner product” on M , encapsulating its distribution of local area elements, and the product $\langle \cdot, \cdot \rangle_C^M$ is the “conformal inner product” on M , encapsulating local angle measurements.

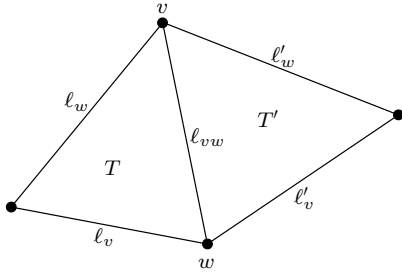
Suppose $F : C^\infty(M) \rightarrow C^\infty(N)$ is a linear *functional map* taking functions on M to functions on N . Given any inner product $\langle \cdot, \cdot \rangle^M$ on M and a corresponding inner product $\langle \cdot, \cdot \rangle^N$ on N , the Riesz Representation Theorem guarantees the existence of an operator $D_{\langle \cdot, \cdot \rangle^M} : L^2(M) \rightarrow L^2(M)$ such that $\langle F(f), F(g) \rangle^N = \langle f, D_{\langle \cdot, \cdot \rangle^M} g \rangle^M$ for all $f, g \in L^2(M)$ [Brezis 2010]. The area-based shape difference between M and N with respect to F is the operator $D_A := D_{\langle \cdot, \cdot \rangle_A^M}$ and the corresponding conformal shape difference is the operator $D_C := D_{\langle \cdot, \cdot \rangle_C^M}$.

In our discussion of discrete problems, we will first focus on discretizing and analyzing the inner product functions $\langle \cdot, \cdot \rangle_A^M$ and $\langle \cdot, \cdot \rangle_C^M$, since these can be understood without relating M to a second surface N . We then transition from inner products to shape differences in §6. In the discrete case, the latter is a consequence of basic linear algebra rather than the Riesz Representation Theorem; essentially if a matrix P defines an inner product $\langle v, w \rangle_P := (Pv)^\top (Pw)$, then we can refactor to write $\langle v, w \rangle_P = v \cdot Dw$ where $D = P^\top P$.

4.2 Discrete Inner Products

Each quantity above is straightforward to discretize in the language of finite elements over a triangle mesh; see [Brenner and Scott 2007; Sayas 2008; Strang and Fix 2008] for general introductions to this approach. To this end, suppose M is represented using a connected, orientable, and manifold triangle mesh with vertices V and triangles T . We model functions as vectors $f \in \mathbb{R}^{|V|}$ interpolated to triangle interiors in piecewise-linear fashion.

We will begin our fine-grained examination of shape differences by posing functional inner products on these meshes in terms of discrete geometry. Our ultimate goal is to show that before truncation in a low-frequency basis, the area-based and conformal inner product matrices completely encode the intrinsic structure of meshed geometry. This property is also stated in [Rustamov et al. 2013] in the

Fig. 1. Notation for the conformal product C .

continuous case; their discretization, however, does not admit such completeness due to the use of lumped area weights, as explained below.

Consider a single triangle T on M , and suppose f and g are affine functions on T ; in other words, f and g are evaluated in the interior of T via barycentric interpolation of the three scalar values f_1, f_2, f_3 and g_1, g_2, g_3 defined on the vertices of the triangle. Multiplying these functions and integrating reveals that the inner product of f and g on T is given by

$$\langle f, g \rangle_A^T = \frac{\mu(T)}{12} (f_1 \ f_2 \ f_3) \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix},$$

where $\mu(T)$ is the area of T and f_i, g_i denote the values of f, g on vertex v_i . As a sanity check, taking $f_i = g_i = 1 \ \forall i$ recovers the area of T . This is the *exact* L^2 inner product of f and g defined over the meshed surface using piecewise-linear interpolation, without mass lumping commonly introduced in finite element discretizations; this distinction is critical for our construction.

Taking inner products over all of M requires summing over triangles T . If $f, g \in \mathbb{R}^{|V|}$, then $\langle f, g \rangle_A^M$ is given by $f^T A g$, where

$$A_{vw} = \frac{1}{12} \cdot \begin{cases} 2 \sum_{T \sim v} \mu(T) & \text{when } v = w \\ \sum_{T \sim e} \mu(T) & \text{when } e = (v, w) \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

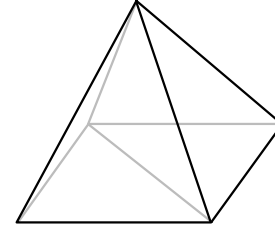
where $T \sim v$ denotes iteration over triangles adjacent to v and $T \sim (v, w)$ denotes iteration over triangles adjacent to edge (v, w) . This $|V| \times |V|$ ‘‘Galerkin mass matrix’’ A is nondiagonal but positive definite, integrating products of piecewise linear functions exactly. See e.g. [Strang and Fix 2008, Chapter 10, (32)] for an example of its appearance in finite elements.

We can think of A as a linear operator $A(\mu) : \mathbb{R}^{|T|} \rightarrow \mathbb{R}^{|V| \times |V|}$ that constructs the area-based functional inner product matrix A given a vector $\mu \in \mathbb{R}^{|T|}$ of triangle areas. We can show that $A(\cdot)$ is invertible in the following sense:

PROPOSITION 1. *Suppose M has a boundary or at least one interior vertex with odd valence. Then, $A(\mu)$ uniquely determines μ , recoverable via a linear solve.*

The proof of this proposition and others below is in the appendix. A proposition of this nature does *not* hold if masses are lumped down the diagonal of A . This observation is intuitive in that a triangle mesh has approximately two times the number of triangles as vertices.

If f and g are piecewise-linear functions on M , then their gradients are piecewise-constant and expressible using one vector per triangle. Taking dot products of these gradients and integrating over M shows that $\langle f, g \rangle_C^M = f^T C g$, where

Fig. 2. A mesh for which $C(\ell^2; \mu)$ is not invertible when $\mu = 1$.

$$C_{vw} = \frac{1}{8} \cdot \begin{cases} -\sum_{u \sim v} C_{uv} & \text{when } v = w \\ \mu(T)^{-1} (\ell_{vw}^2 - \ell_v^2 - \ell_w^2) & \text{when } v \sim w \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Notation for the $v \neq w$ case is shown in Figure 1; $e \sim v$ denotes an edge e adjacent to vertex v , and ℓ_{vw} is the length of the corresponding edge. This matrix is the familiar cotangent Laplacian matrix cast in terms of edge lengths and triangle areas; this form also appears e.g. in [Boscaini et al. 2015]. Comparing (1) and (2), scaling the edge lengths of a mesh by some factor α will correspondingly scale A by α^2 while C will be left unchanged; unless otherwise noted, we scale meshes in our experiments to have unit surface area to remove dependence on global scaling.

A crucial observation that we make here is that if the triangle areas encoded in μ are fixed then the mapping $C(\ell^2; \mu) : \mathbb{R}^{|E|} \rightarrow \mathbb{R}^{|V| \times |V|}$ taking squared edge lengths $\ell^2 \in \mathbb{R}^{|E|}$ to a conformal inner product matrix C is *linear*. Note also that C is fully determined by its values C_{vw} for $v \sim w$. Thus, if we represent the list of inner products C_{vw} as a vector c in $\mathbb{R}^{|E|}$ then for a fixed set of area weights μ , there exists a matrix B_μ such that $c = B_\mu \ell^2$. The entries of B_μ are, of course, given in eq. (2). In the pipeline that we propose below, we will first recover the triangle areas and then use those to recover edge lengths from the corresponding inner products. The following proposition shows that ‘‘generically’’ the matrix B_μ is invertible, i.e., the set of weights μ for which B_μ is singular has measure 0 in $\mathbb{R}^{|T|}$.

PROPOSITION 2. *Assume that the mesh M is manifold without boundary. Then, for almost all choices of areas μ , the map $C(\ell^2; \mu)$ uniquely determines ℓ , which is recoverable via a linear solve.*

This proposition implies that the linear map $C(\ell^2; \mu)$ is invertible for a small (possibly zero) perturbation of any set of area weights μ . Nevertheless, there exist cases in which the squared edge lengths are not recoverable via inversion of the linear map $C(\ell^2; \mu)$ for a fixed set of area weights. One example of such a shape is shown in Figure 2, consisting of two tetrahedra glued at their bases. In this case, all the triangles have equal area weights, and it can be seen that the resulting linear system is singular. We also remark that the condition of no boundary is necessary in the Prop. 2 above, as it is possible to construct meshes for which the map $C(\ell^2; \mu)$ is singular for *all* choices of μ (e.g., a pair of triangles glued along a shared edge). For all the meshes that we tried in practice (§8), we have observed that the resulting system is both invertible and typically well-conditioned. We leave the formulation of the necessary and sufficient conditions on the mesh and the weights μ for the invertibility of $C(\ell^2; \mu)$ as a question for future work.

5. ENCODING EXTRINSIC STRUCTURE

Intrinsic inner products capture the metric tensor (first fundamental form) of a surface, so to complete our representation we show how a

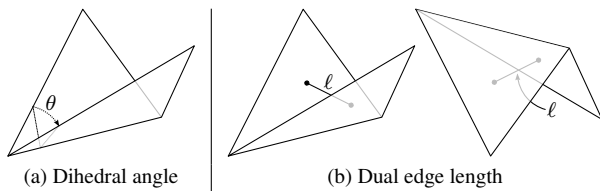


Fig. 3. Two potential ways to encode extrinsic mesh structure.

related structure can be used to encode its second fundamental form. In keeping with previous discussion, we will use additional inner product matrices to derive a multiscale representation of this missing information. While there exist many possible ways to measure extrinsic distortion, this “functional” language facilitates a connection between continuous and discrete characterizations and unifies our treatment of intrinsic and extrinsic distortion.

These added structures complement the area-based and conformal products by making our representation of a shape unique up to rigid motion. In addition to providing a lossless representation of surface geometry in the presence of complete information, we demonstrate how the new products can capture and encode geometric relationships that are not captured by purely intrinsic analysis.

5.1 Extrinsic Alternatives

In discrete language, the inner product matrices $A(\mu)$ and $C(\ell^2; \mu)$ determine the edge lengths of a triangle mesh but not its dihedral angles, illustrated in Figure 3(a). Additionally providing dihedral angles is sufficient to recover a mesh up to rigid motion. There are many expressions of extrinsic shape that potentially encode these angles; before presenting our final solution, we mention a few straightforward alternatives to explain why they are less desirable.

At the most basic level, any technique encoding one value per edge of a triangle mesh could be used to represent dihedral angles. For instance, since the angles are in a vector $\theta \in \mathbb{R}^{|E|}$, we could use an analog of Proposition 2 to store them in the matrix $C(\theta; \mu)$. That is, from a mechanical perspective Proposition 2 shows we can take one value per edge in a vector $\theta \in \mathbb{R}^{|E|}$ and generate a matrix $C(\theta; \mu)$ without losing information. One could plausibly replace edge lengths in θ —the original intention of this construction—with dihedral angles; this roughly corresponds to taking products of functional gradients under the second fundamental form h as $\int h(\nabla f, \nabla g) dA$. The resulting matrix is not positive semidefinite, however, which prevents a smooth analog from the Riesz Representation Theorem (which applies only to positive definite inner products) and causes numerical issues due to departure from the cone of semidefinite matrices. Dihedral angles also are known only up to a period of 2π , providing potential for ambiguity in the expression of the vector θ .

In an attempt to bring back the positive definiteness enjoyed by the intrinsic formulation, we might attempt to encode the edge lengths of a *dual* mesh, shown in Figure 3(b). These lengths indirectly encode dihedral angles up to sign but are unable to distinguish between inward and outward folding directions, as shown in the figure. Obvious techniques for disambiguating the inward and outward folds generally accompany edge lengths with signs, reintroducing the problems discussed in the previous paragraph.

An alternative construction might define extrinsic shape differences via the *Gauss map*, or map from a surface into the unit sphere based on normal direction; see [Meyer et al. 2003] for an example in geometry processing. While the Gauss map is used in classical differential geometry to derive extrinsic properties of surfaces, we

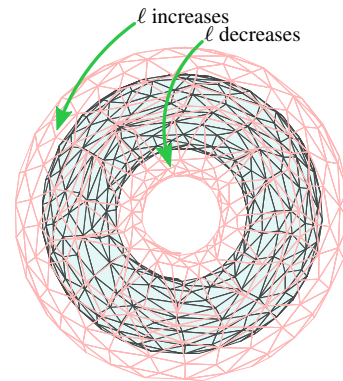


Fig. 4. Edge lengths change according to curvature of the offset surface.

find it to be unstable within the shape difference framework. In particular, the image of the Gauss map is composed of many overlapping spherical triangles that change rapidly from vertex to vertex. Projection of this information into low-frequency Laplace-Beltrami bases tends to remove the majority of the meaningful geometric signal. In a sense, however, we can view the offset surface construction proposed below as a means of smoothing out this construction.

Before proceeding, we should remark that strictly speaking it may not be necessary to provide extrinsic information at all. According to a classical result by Gluck [1975], almost all triangulated simply connected closed surfaces are rigid. Although this result might imply that triangle edge lengths are, in general, sufficient to reconstruct the mesh up to rigid motion, this is only true if the metric is known *exactly*; moreover, it is highly nontrivial to recover an embedding even if one is known to exist. When the edge-lengths are perturbed or are approximated, the corresponding embedding might either not exist or be very far from the desired shape. As we show below, the presence of explicit information about the extrinsic distortion can greatly help in both direct and inverse problems and provides complementary information to the intrinsic distortion measured by prior methods.

5.2 Offset Surfaces

Our construction of an extrinsic shape representation is an extension of the dual mesh idea from §5.1 that does not suffer from sign ambiguity. Instead, we are able to rely upon the positive definiteness of inner product matrices directly to encode both intrinsic and extrinsic information. In short, rather than encoding a metric and its derivative, we encode a metric and a slightly deformed metric, both of which admit natural positive definite representations.

The intuition for our construction is illustrated in Figure 4. Suppose we wish to recover the embedding of the blue torus. As discussed in the previous section, it may be difficult to reconstruct the torus purely from its list of edge lengths. Instead, suppose we generate an *offset surface* by displacing each vertex and face along its outward normal a fixed distance t . The operation is extrinsic, since the mesh moves through the surrounding space, modulating edge lengths ℓ based on the curvature of the surface. The edge lengths in the interior of the torus shrink while the edge lengths on the exterior expand, effectively distinguishing the bend direction.

In the continuous case, we can formalize the effect of offsetting a surface as follows:

PROPOSITION 3. *Suppose M is a compact orientable Riemannian 2-manifold without boundary. Consider a family of immersions*

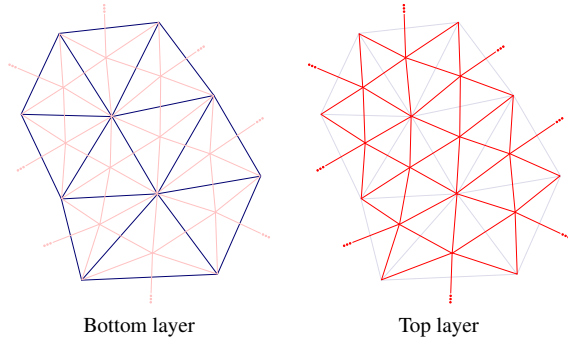


Fig. 5. Topology of offset mesh.

$F_t : M \rightarrow \mathbb{R}^3$ satisfying

$$\frac{\partial F_t}{\partial t}(p) = n(p) \quad \forall (p, t) \in M \times \mathbb{R}_+,$$

where n denotes the outward unit normal of $M_0 := F_0(M)$ when $t = 0$. If $g_{ij} := \langle \frac{\partial F_t}{\partial x_i}, \frac{\partial F_t}{\partial x_j} \rangle$ is the metric of the embedded surface and $h_{ij} := \langle \frac{\partial F_t}{\partial x_i}, \frac{\partial n}{\partial x_j} \rangle$ is its second fundamental form, then

$$\frac{\partial g_{ij}}{\partial t} \Big|_{t=0} = 2h_{ij} \Big|_{t=0} \text{ and } \frac{\partial \mu}{\partial t} \Big|_{t=0} = H\mu,$$

where $H := (g^{ij} h_{ij})_{t=0}$ is the mean curvature at $t = 0$ and $\mu := \sqrt{\det g}|_{t=0}$. Furthermore, for any smooth $\phi, \psi : M \rightarrow \mathbb{R}$,

$$\frac{\partial}{\partial t} g(\nabla \phi_t, \nabla \psi_t) = -2h(\nabla \phi_t, \nabla \psi_t),$$

where $\phi_t := \phi \circ F_t^{-1}$, $\psi_t := \psi \circ F_t^{-1}$, and gradients are along $M_t := F_t(M)$.

Results of this nature are fairly well-known for offset surfaces; see e.g. [Patrikakis and Maekawa 2009] for related discussion. More informally, the proposition shows that the second fundamental form of M is encoded through the change in metric while the surface is being offset along its normal directions.

When M is an oriented triangle mesh, there are many potential constructions of discrete offset surfaces, and several likely would suffice for the proofs in this paper. For mathematical simplicity, we choose the construction in Figure 5. On the left we show triangles of the original mesh M in blue. On the right, we define the topology of the offset mesh in red, which contains a vertex for every vertex of M and every triangle of M . For a fixed constant $t > 0$, we place the vertices distance t above M along its face/vertex normals; any reasonable definition of a unit-length vertex normal suffices. Offset vertices associated with triangles are placed directly above the barycenter of the triangle.

5.3 Recovery of Embedding

In the end, we encode the geometry accompanying a fixed triangle mesh topology using four structures: the intrinsic area-based and conformal inner product operators and the same operators for the offset surfaces with fixed normal offset distance $t > 0$. We denote the offset surface of M as M_t . In this section, we show—at least before truncation—that these four difference matrices are sufficient for fully reconstructing a shape.

The challenge of reconstructing a triangle mesh from its edge lengths arguably comes from the fact that there are many ways to

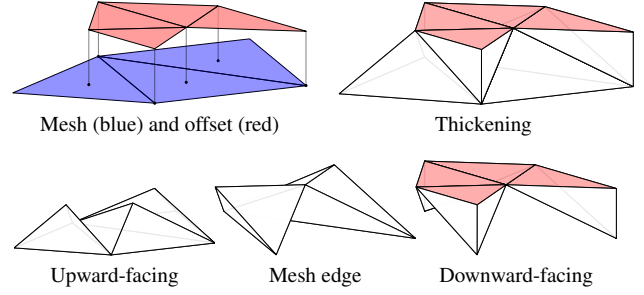


Fig. 6. Canonical thickening (top right) of a triangle mesh (top left); types of tetrahedra in the thickening (bottom).

glue together two adjacent triangles by fixing different dihedral angles. Rigidity may imply that only one such embedding exists, but it is not obvious from local relationships. Contrastingly, an oriented *tetrahedral* mesh is easy to reconstruct from its list of edge lengths simply by gluing individual tetrahedra face by face.

Hence, our intuition for why a mesh plus its offset are enough to reconstruct the mesh comes from a volumetric perspective. This intuition is confirmed by Proposition 3 and the Gauss–Codazzi equations [Bonnet 1867; Carmo 1976], since offset geometry provides extrinsic curvature information, which in turn determines an embedding. We chose the topology of the offset mesh (Figure 5) specifically to allow for a canonical “thickening” of the offset slice into a tetrahedral mesh, shown in Figure 6. Using mesh-based shape differences, we can recover the edge lengths of the bottom and top layers of the thickening. By construction of the offset mesh, we are able to recover the lengths of the interior edges of the thickening, effectively proving the following proposition:

PROPOSITION 4. *Suppose a mesh M satisfies the criteria in Propositions 1 and 2. Given the topology of M , the area-based and conformal product matrices $A(\mu)$ and $C(\ell^2; \mu)$ of M , and the area-based and conformal product matrices $A_t(\mu_t)$ and $C(\ell^2_t; \mu_t)$ of M_t , the geometry of M can (almost always) be reconstructed up to rigid motion.*

A formal argument for this proposition is in the appendix.

5.4 Discussion

We pause to summarize the theoretical development in the previous sections. We began by reconsidering the construction of inner products and shape differences from first-order finite elements. When area elements are not lumped, we showed that inner product matrices fully determine the edge lengths of a mesh and that they can be recovered by solving two linear systems of equations: one for recovering the triangle areas, and the other for recovering the edge lengths. Moreover, generically, both systems are non-singular.

In both the continuous and discrete cases, these intrinsic measurements are not enough to distinguish isometric shapes. Even worse, the space of near-isometric shapes can be very large. Hence, we propose generating an offset surface M_t from a mesh or surface M . In the continuous case, the geometry of M_t determines the *extrinsic* structure of M by encoding its second fundamental form. In the discrete case, combining edge lengths of M with edge lengths of M_t fully determines M up to rigid motion. The main development is that we can *completely* determine a shape using functional inner products via the constructions above.

Our theoretical contributions deal with the noise-free, non-truncated case. Roughly, they show that *if* intrinsic/extrinsic shape differences were computed from an embedded mesh M with fixed

topology, then the embedding of M almost always can be covered from those differences up to rigid motion. We evaluate sensitivity to noise and the possibility of recovering geometry from truncated shape differences empirically in §8.

6. FROM INNER PRODUCTS TO SHAPE DIFFERENCES

With the goal of working with quantities that exist when meshes are not in vertex-for-vertex correspondence, we shift from working with matrices of inner products to shape differences. This shift is needed to propose algorithms in §7 for estimating the dense structure of a target mesh given a source mesh and an approximate relationship between the source and the target, represented as a functional map computed e.g. using assorted correspondence techniques.

6.1 Discrete Shape Differences

We begin by considering two meshes M and N in vertex-for-vertex correspondence, with areas $\mu_M, \mu_N \in \mathbb{R}^{|T|}$ and squared edge lengths $\ell_M^2, \ell_N^2 \in \mathbb{R}^{|E|}$. Based on the continuous definitions in §4.1, the “full” area-based and conformal shape difference between meshes M and N are [Rustamov et al. 2013, §5, “option 1”]

$$\begin{aligned} D_A &= A(\mu_M)^{-1} A(\mu_N) \\ D_C &= C(\ell_M^2; \mu_M)^{-1} C(\ell_N^2; \mu_N). \end{aligned} \quad (3)$$

A straightforward corollary of the discussion in §4.2 is that these two differences completely determine the edge lengths and triangle areas of N given the geometry of M . Notice the first relationship is still linear in μ_N and the second in ℓ_N^2 , so reconstruction still amounts to solving a linear system.

Similarly, the extrinsic differences are simply the shape differences between the offset surfaces:

$$\begin{aligned} D_A^E &= A(\mu_{M_t})^{-1} A(\mu_{N_t}) \\ D_C^E &= C(\ell_{M_t}^2; \mu_{M_t})^{-1} C(\ell_{N_t}^2; \mu_{N_t}). \end{aligned} \quad (4)$$

The discussion in §5.3 implies that the tuple (D_A, D_C, D_A^E, D_C^E) is sufficient to reconstruct N up to rigid motion given M .

6.2 Source-Truncated Correspondence

More commonly, suppose $\Phi \in \mathbb{R}^{|V| \times k}$ contains the orthonormal Laplace–Beltrami basis of M , truncated to k functions. Assuming M and N are still in vertex-for-vertex correspondence, we can write “reduced” shape differences as

$$\begin{aligned} D_A^\Phi &= \Phi^\top A(\mu_N) \Phi \\ D_C^\Phi &= \text{diag}(-\{\lambda_i^M\}) + \Phi^\top C(\ell_N^2; \mu_N) \Phi, \end{aligned} \quad (5)$$

where the eigenvalues of the Laplacian on M are λ_i^M . These differences no longer determine angles and edge lengths exactly but still encode a multiscale notion of geometry that is valuable for understanding the relationships between M and N ; extrinsic differences can be defined analogously from the offset surface. We can still define linear systems for computing μ_N and ℓ_N^2 from D_A, D_C, μ_M , and ℓ_M^2 using these relationships, although they are unlikely to be full-rank for small k ; we provide regularizers in the next section.

These truncated differences essentially correspond to removing rows and/or columns from the full shape differences after writing them in the Laplace–Beltrami eigenbasis. Such a computation can be useful for multiscale analysis of surface deformations, in which vertex-for-vertex correspondence is known but high-frequency changes may not be useful to analyze. What remains,

however, is to consider the case when M and N are not in vertex-for-vertex correspondence and both have incomplete bases.

6.3 Source- and Target-Truncated Correspondence

Suppose we are given truncated bases $\Phi^M \in \mathbb{R}^{|V_M| \times k_M}$ and $\Phi^N \in \mathbb{R}^{|V_N| \times k_N}$ for the eigenspaces of M and N , respectively, and a functional map matrix $F \in \mathbb{R}^{|V_N| \times |V_M|}$ taking functions written in the Φ^M basis on M to functions in the Φ^N basis on N . Following [Rustamov et al. 2013, §5], we define shape differences in this case as

$$\begin{aligned} D_A^{\Phi^M, \Phi^N} &= F^\top F \\ D_C^{\Phi^M, \Phi^N} &= \text{diag}(-\{\lambda_i^M\}) + F^\top \text{diag}(-\{\lambda_i^N\}) F. \end{aligned} \quad (6)$$

Whereas the truncated shape differences in (5) contain a limited window of values from the full shape difference matrix, in this final case the non-truncated entries of the shape difference matrices also undergo some change. This is because even if a function on M is in the column space of Φ^M , it will not be transported fully to N by the functional map F due to removal of high frequencies.

These shape differences are discretizations of analogous linear operators in the smooth setting. For this reason, even though the differences in (6) no longer satisfy exact equality relationships like those in (5) for recovering areas μ_N and squared edge lengths ℓ_N^2 from shape differences and the geometry of M , we will pose *approximate* relationships

$$\begin{aligned} D_A^{\Phi^M, \Phi^N} &\approx (\Phi^M)^\top A(\mu_{M \leftarrow N}) \Phi^M \\ D_C^{\Phi^M, \Phi^N} &\approx \text{diag}(-\{\lambda_i^M\}) + \Phi^{M \top} C(\ell_{M \leftarrow N}^2; \mu_{M \leftarrow N}) \Phi^M. \end{aligned} \quad (7)$$

The unknown variables $\mu_{M \leftarrow N}$ and $\ell_{M \leftarrow N}^2$ can be thought of as pullbacks of the metric of N to that of M , in the sense that they attempt to assign areas and edge lengths to the topology of M to mimic inner products on N . The first condition is linear in $\mu_{M \leftarrow N} \in \mathbb{R}^{|T_M|}$ and the second in $\ell_{M \leftarrow N}^2 \in \mathbb{R}^{|E_M|}$.

Since the shape differences in (6) are the most realistic test cases, we will assume in our experiments that truncated shape differences are computed in this fashion unless noted otherwise. That is, we will assume that we are given a source- and target-*truncated* shape difference. The experiments in §8.2 verify that this approximation is reasonable as long as k_M and k_N are sufficiently large.

7. RECOVERY OF INTRINSIC AND EXTRINSIC STRUCTURE

Having established theoretical aspects of intrinsic and extrinsic shape differences, we now provide algorithms for recovering a shape N given a base shape M and shape differences to N and its offset. First, we recover triangle areas from the base and offset surfaces from corresponding area-based shape differences. With these areas fixed, we then recover edge lengths, which were shown in §5.3 to completely determine the surface.

Both steps can be carried out using linear solves when shape differences are not truncated. When dealing with truncated or inexact functional maps, we augment the optimization with constraints ruling out unreasonable structures. We also show how to apply existing techniques for recovery of an embedding from edge lengths of the surface and its offset.

7.1 Triangle Area Computation

We first show how to recover areas of triangles given an area-based shape difference. Our approach is an extension of the basic linear

technique outlined in the proof of Proposition 1, extended to deal with truncation and noise.

Following §6, suppose D_A is the area-based shape difference between M and N in the Laplace–Beltrami basis Φ_M . Recall that our goal is to pull the geometry of N back to the mesh of M . Hence, the area-based difference from M to the reconstructed target shape N^* should satisfy $D_A^* = \Phi_M^\top A(\mu_N) \Phi_M$. If the reduced basis Φ_M on M has k functions, this linear system for μ_N has k^2 equations and $|T|$ unknowns. So, we need at least $k \sim \sqrt{|T|}$ to have a well-posed system.

The quality of the solution found by solving this system without regularization depends on two factors: the quality of D_A and the conditioning of the resulting linear problem. We find that both limitations are improved considerably by introducing a nonnegativity constraint, leading to the following optimization problem for μ_N :

$$\min_{\mu_N} \|\Phi_M^\top A(\mu_N) \Phi_M - D_A\|_{\text{Fro}}^2 \quad (8)$$

s.t. $\mu_N(T) \geq 0 \forall \text{ triangles } T.$

We solve this and other convex programs using the Mosek toolbox [MOSEK ApS 2015]. We additionally add a regularizing viscosity term $\varepsilon \|\mu_N - \mu_M\|_2^2$ for small $\varepsilon > 0$, under the assumption that triangle areas should change minimally unless there is evidence to do otherwise; we set $\varepsilon = 10^{-4}$ in all the experiments in this paper. This regularizer allows the problem to be solved when D_A is truncated, which would otherwise make it underdetermined, and regardless improves stability when D_A is noisy.

7.2 Edge Length Computation

Now that we can compute triangle areas, we can recover edge lengths. As in the last section, we start from Proposition 2 to propose a basic linear system for squared edge lengths and then provide regularization techniques for dealing with inexact or truncated differences.

The conformal shape difference encodes the transformation of the cotangent Laplacian through the deformation. Again borrowing from §6, the geometry of N can be pulled back to M via the following linear condition on squared edge lengths ℓ_N^2 given fixed areas μ_N :

$$\text{diag}(-\lambda_i^M) D_C = \Phi_M^\top C(\ell_N^2; \mu_N) \Phi_M. \quad (9)$$

Solving this linear system of equations for ℓ_N^2 depends critically on the approximated areas μ_N ; numerical or discretization error from the method in §7.1 invalidates this step, regardless of the quality of D_C . To provide resilience to this issue and to noise in D_C , we add constraints to this system ruling out unrealistic edge lengths ℓ_N^2 .

To define a triangulation, the squared edge lengths ℓ_N^2 must be nonnegative; furthermore, $\sqrt{\ell_N^2(T)}$ must respect the triangle inequality in each mesh triangle T . We enforce the latter constraint via the following proposition:

PROPOSITION 5. *The symmetric matrix \mathbf{E} defined by*

$$\mathbf{E} = \frac{1}{2} \begin{pmatrix} 2x_1 & x_3 - x_1 - x_2 & x_2 - x_1 - x_3 \\ x_3 - x_1 - x_2 & 2x_2 & x_1 - x_2 - x_3 \\ x_2 - x_1 - x_3 & x_1 - x_2 - x_3 & 2x_3 \end{pmatrix}$$

is positive semidefinite if and only if x_1, x_2, x_3 are nonnegative and their square roots satisfy the triangle inequality.

We also can link squared edge lengths to the computed triangle areas $\mu_N(T)$. This link is provided by the submatrices \mathbf{E}_k defined

as

$$\mathbf{E}_k = \frac{1}{2} \begin{pmatrix} 2x_i & x_k - x_i - x_j \\ x_k - x_i - x_j & 2x_j \end{pmatrix},$$

where $\{i, j, k\} = \{1, 2, 3\}$. With this definition in place, we leverage the following proposition:

PROPOSITION 6. *\mathbf{E} is positive semidefinite if and only if $x_k \geq 0$ for all $k \in \{1, 2, 3\}$ and $\det(\mathbf{E}_3) \geq 0$. Moreover, if $\mathbf{E} \succeq 0$, then $\det(\mathbf{E}_k) = 4\mu_N(T)^2$.*

Enforcing constraints derived from these relationships in the computation of edge lengths from a shape difference leads to the following optimization problem:

$$\min_{\ell_N^2} \|\Phi_M^\top C(\ell_N^2; \mu_N) \Phi_M - \text{diag}(-\lambda_i^M) D_C\|_{\text{Fro}}^2$$

s.t. $\ell_N^2 \geq 0$
 $\det(\mathbf{E}_3(T)) = 4\mu_N(T)^2 \forall \text{ triangles } T.$

This problem, however, is large and non-convex due to the determinant constraint. A convex relaxation is possible by noticing that the cone of symmetric positive semidefinite matrices with determinant ≥ 1 is convex; this observation derives from the convexity of the function $A \mapsto -\log(\det A)$ [Boyd and Vandenberghe 2004]. So, the former problem can be relaxed to a convex problem:

$$\min_{\ell_N^2} \|\Phi_M^\top C(\ell_N^2; \mu_N) \Phi_M - \text{diag}(-\lambda_i^M) D_C\|_{\text{Fro}}^2$$

s.t. $\ell_N^2 \geq 0$
 $\det(\mathbf{E}_3(T)) \geq 4\mu_N(T)^2 \forall \text{ triangles } T.$ (10)

The determinant constraint is handled using the rotated quadratic cone optimization in the Mosek toolbox [MOSEK ApS 2015]. While (10) contains a relaxation of the full set of constraints, we find empirically that this relaxation generally is tight; we leave it to future work to prove conditions for “exact recovery” akin to those in [Dym and Lipman 2016] for mesh alignment problems.

As in §7.1, we can additionally regularize by adding $\varepsilon' \|\ell_N^2 - \ell_M^2\|_2^2$ to the objective; our experiments use $\varepsilon' = 10^{-4}$.

7.3 Global Extrinsic Reconstruction

At this point, we have presented algorithms for recovering edge lengths for the entire canonical thickening defined in §5.3. As suggested in the proof of Proposition 4, if these edge lengths are computed without error, the thickening can be reconstructed greedily; then, the embedding of N from M is the inner envelope of this thickening.

In reality, the squared edge lengths in ℓ_N^2 likely exhibit numerical error. For this reason, we employ the algorithm in [Panozzo et al. 2014] for reconstructing a triangle mesh given its edge lengths. We adapt their approach to take into account the tetrahedra defined by the offset surface, by using the same ARAP-style deformation energy, defined on each triangle facet of each tetrahedron, and using the same alternating optimization strategy. We note, in particular, that this approach does not require embedded surfaces to be manifold, and can easily incorporate edges shared by more than two triangles, which only changes the computation of the gradient of the energy. Hence this allows us to reconstruct the entire set of triangles in the canonical thickening rather than the inner or outer surfaces only. We provide the thickening of M as a starting point for their alternating optimization algorithm. Whereas their method is subject to isometric ambiguity when embedding manifold meshes, reconstructing the entire thickened structure reduces ambiguity and more reliably provides an extrinsically correct embedding.

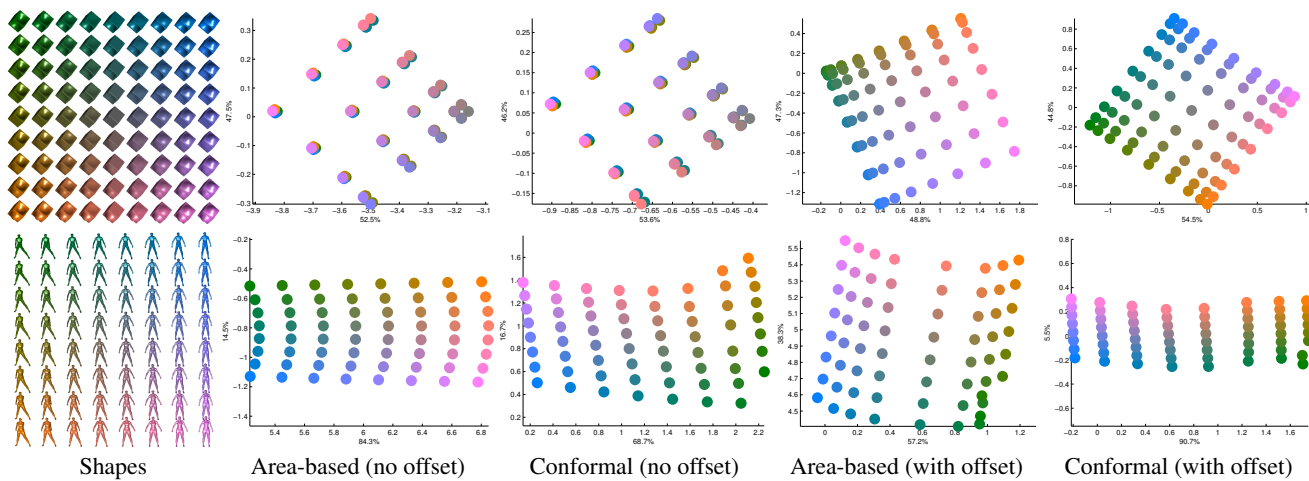


Fig. 7. PCA on collections of shape differences reveals the axes of variability within a collection; each shape on the left is colored the same as its corresponding points in the plots. The area-based and conformal differences are unable to distinguish the inward and outward bumps in the top example, leading to clusters of four points.



Fig. 8. Human models from Figure 7 sorted by the first PCA dimension for area-based shape differences (top) and area-based differences including an offset surface (bottom). The differences without offsets distinguish body type, while the differences with offsets distinguish pose.

8. EXPERIMENTS

In this section we illustrate the utility of the constructions presented above in a variety of practical application scenarios. We start by showing how the extrinsic shape differences can be useful for shape exploration and analysis, by complementing the information provided by the intrinsic differences of Rustamov et al. [2013]. We then show how our metric and shape recovery methods can be used to both infer shape structure and ultimately recover the embedding from approximate, truncated shape differences.

8.1 Shape Space

An example application of shape differences that does *not* rely on exact reconstruction of local geometry involves the extraction of variability within a collection of related shapes. Suppose we choose an arbitrary base shape and compute its shape difference matrices with the remaining shapes in a collection. Then, a simple low-dimensional description of shape variability is to do PCA on the collection of matrices, resulting in the embedding of each shape as a point in PCA space.

We use PCA experiments to illustrate the power of our proposed extrinsic differences. For instance, Figure 7 illustrates embeddings of two-parameter shape collections into the plane using the procedure above ($k_M = 50, k_N = 100$). The top row illustrates the need for extrinsic differences most clearly. Here, we generate cubes with smooth bumps, smoothly varying from an inward bump to an outward bump. Intrinsic shape differences are identical for inward and outward bumps, leading to PCA embeddings that cluster sets of four shapes together. Adding extrinsic information disambiguates the embedding problem, separating the clustered points. Similarly, the

extrinsic area-based shape difference best separates the parametric human models evenly among the two axes (57.2% variability along the principal axis, 38.3% along the secondary axis); interestingly, conformal shape differences among offset surfaces do not exhibit much variability for this particular class of surfaces.

Figure 8 highlights how intrinsic and extrinsic shape differences can measure different properties of shape. We sort the collection of human models by the one-dimensional embeddings (x -axis) of intrinsic (top) and extrinsic (bottom) area-based shape differences ($k_M = 50, k_N = 100$). The intrinsic shape differences distinguish the body type of the model and are invariant to the pose of the arms and legs; this ordering reflects the property that articulated deformations of humans are nearly isometric. Complementing this embedding, the extrinsic differences distinguish pose and are less sensitive to body type. This property is also visible in Figure 7 since the area-based embeddings without and with the offset are transposed from one another.

Figure 9 shows a similar experiment applied to shapes from individual frames of animation sequences. Both intrinsic and extrinsic shape differences are able to recover the cyclic structure of a galloping horse animation; this indicates that the galloping motion contains both intrinsic and extrinsic deformation modes. Contrastingly, the intrinsic differences severely underperform in recovering an animated sequence of deforming cloth. The physics of cloth naturally avoids intrinsic stretching and shearing, maintaining the initial developable structure. Thus, intrinsic shape differences provide little-to-no information, while the extrinsic differences capture the evolution of the animation.

From a wider perspective, the experiments in this section reveal the value of explicitly representing both intrinsic and extrinsic de-

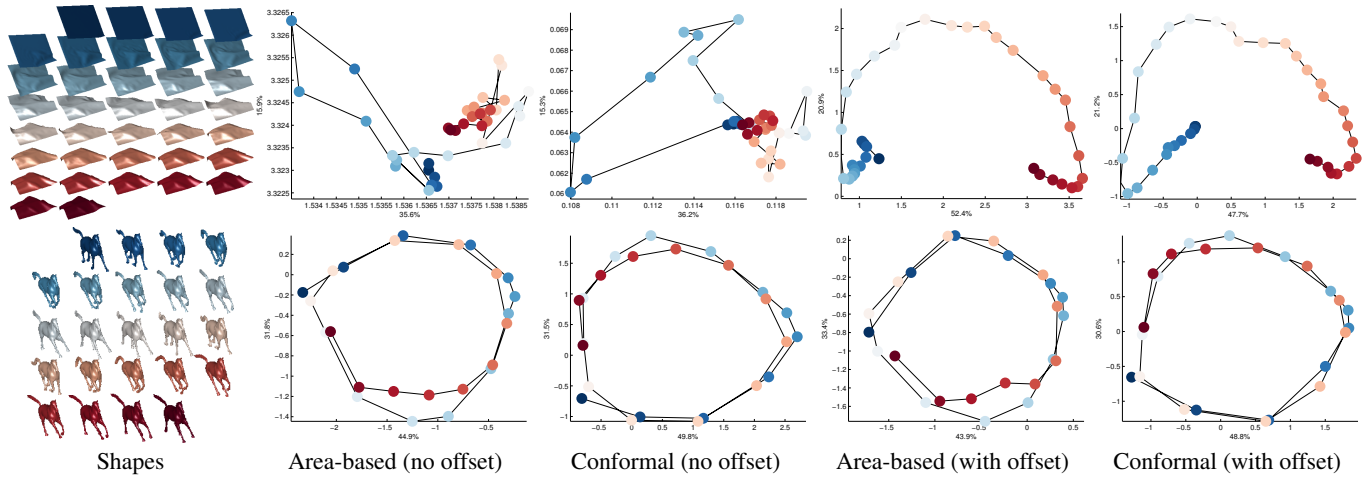


Fig. 9. PCA on shape differences applied to recovering the sequence of animated frames for a simulated piece of cloth (top) and galloping horse (bottom).

formation in navigating datasets of 3D surfaces. A sizable fraction of geometry processing algorithms, including the original work on shape differences, focuses on shape exploration based exclusively on intrinsic structure. Yet, motions like the deformation of a piece of cloth cannot be captured by this representation. While cloth deformation may be an extreme example, based on these results we advocate inclusion of both intrinsic and extrinsic structures in shape analysis rather than discarding the extrinsic information.

8.2 Effects of Truncation

The propositions in this paper show that discrete shape differences completely encode geometric structure when they are written in a full basis. For many applications, however, we approximate shape differences in a truncated low-frequency spectrum. While the effects of this truncation are difficult to characterize mathematically, in this section we evaluate the effects of this approximation numerically.

There are two potential sources of truncation error in the twice-truncated differences discussed in §6.3: The choice of k_M and the choice of k_N . As mentioned in §6.3 these two parameters have slightly different effects; decreasing k_M corresponds to removing rows or columns of the shape difference matrices, while decreasing k_N can affect the values of the entries.

Figure 10 illustrates the results of an experiment varying k_M and k_N for intrinsic shape differences and using the pipeline described in §7 to recover areas and edge lengths; recall that this technique extracts areas and edge lengths on N using calculations on M . We choose a pair of meshes with a ground-truth map to avoid additional error due to map approximation ($|V_M| = |V_N| = 1000$).

Each color plot shows the relative error of assorted quantities extrapolated from the truncated shape differences: face areas (μ), edge lengths (ℓ), truncated eigenvalues ($\{\lambda\}_{i=1}^{k_N}$), full eigenvalues ($\{\lambda\}_{i=1}^{|V|}$), and entries of the Laplacian (Δ). We assume $k_N \geq k_M$, providing the upper-triangular structure of the plots; the vertical axis represents k_M (range: $k_M \in [60, 500]$) and the horizontal axis represents k_N (range: $k_N \in [60, 500]$). We choose ε so that the viscosity regularizer contributes $< 10\%$ of the optimal objective.

These plots show that even truncated shape differences can be used to extract per-face and per-edge information about the mesh using our pipeline. Even with 15% of the Laplacian eigenvectors, we can relatively reliably extract the face areas and edge lengths

of the target mesh. Even on challenging tasks like recovering the *full* spectrum of the target mesh—beyond the eigenvalues used to compute the shape difference—our algorithm has some success.

The choice of k_N is particularly important. Intuitively, this phenomenon might be explained by the fact that modulating k_N changes the values in the shape difference matrices rather than just their size. The top row of the matrix also exemplifies a pattern we observed across our experiments; below a certain value for k_M , there is not enough information to get a meaningful indication of local geometry from the shape difference matrices.

8.3 Intrinsic Recovery

The experiments in §8.2 illustrate a remarkable observation, that we are able to recover local information about the target of a shape difference from a truncated shape difference. That is, the nonnegativity and semidefinite constraints proposed in §7 paired with regularization are sufficient to avoid the null space of the truncated linear systems for recovering areas and edge lengths.

As the cotangent Laplacian of a triangle mesh (with or without area weights) is an intrinsic structure, we can use our computed vectors μ and ℓ to *pull back* the Laplacian operator from N to M .

This technique is illustrated in Figure 11. In this experiment, we compute shape differences from a 100×200 functional map to compute μ and ℓ on the source surface; we then use (2) to construct a new Laplacian operator on M using μ and ℓ pulled back from N and show eigenfunctions of the resulting operator. Not only do the eigenvalues of the pulled-back Laplacian better approximate the Laplace–Beltrami eigenvalues of N , but qualitatively the eigenfunctions of the pulled-back Laplacian exhibit more structure in common with the eigenfunctions of N . Boxed examples in Figure 11 show particularly striking differences between the source and reconstructed eigenfunctions.

Figure 12 illustrates an application of recovering edge lengths from truncated shape differences. Without constructing an embedding, we use pulled-back edge lengths to compute two commonly-used intrinsic functions: single-source geodesic distances and the wave kernel signature [Aubry et al. 2011]. Our edge lengths enable computation of these functions on the source mesh using the metric of the target, given a functional map between them. As a baseline, computing these functions on the target and pulling them back to the source using the functional map (right column) is less accurate;

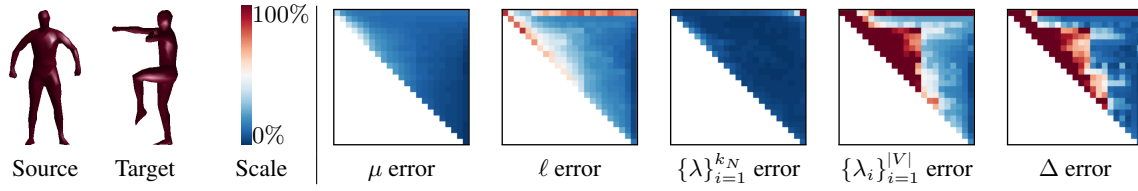


Fig. 10. Effects of truncation on computation of mesh structure. See §8.2 for discussion.

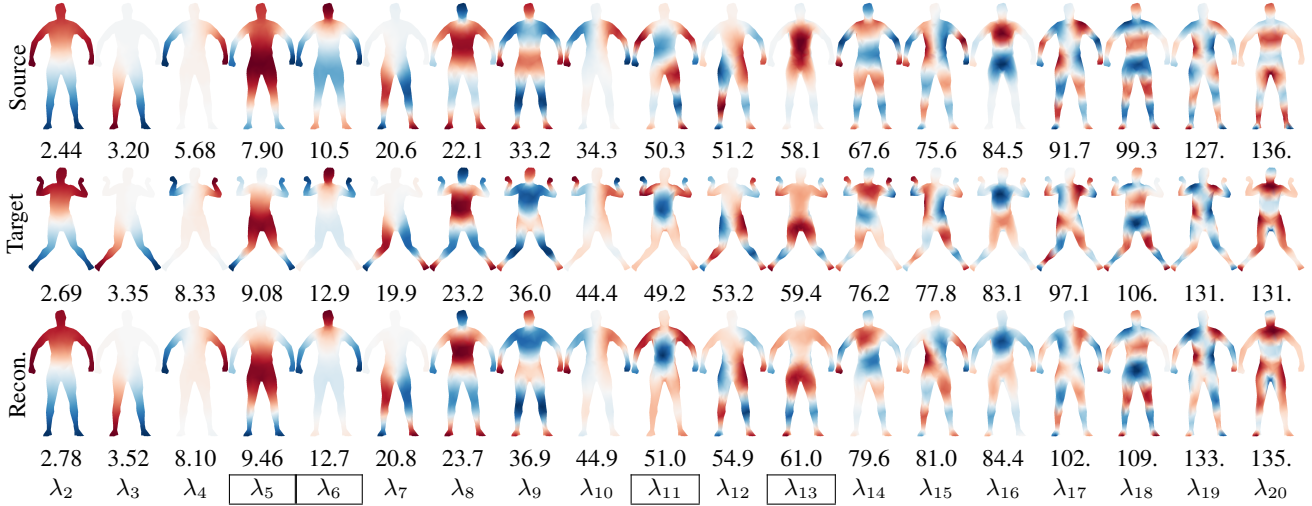


Fig. 11. Our machinery can be used to pull back Laplacian operators from a target mesh to a source. Here, we use a truncated functional map (100 Laplace–Beltrami basis functions on source, 200 on target) to compute revised edge lengths on the source mesh. Eigenfunctions of the Laplacian before and after edge length adjustment are shown with eigenvalues; boxed columns provide examples where the eigenfunction changes structure significantly.

this is due to truncation of the functional map, which removes high frequencies e.g. at the center point of the geodesic function.

8.4 Reconstruction

Figures 13, 14 and 15 illustrate experiments in which geometry is reconstructed after estimating local structure from shape differences. To highlight our method’s effectiveness on extrinsic motion, we show behavior on human shapes and cloth simulation data.

Figure 13 applies our method to reconstructing models of humans from shape differences. From a coarse human base mesh ($|V_M| = |V_N| = 502$; $k_M = 100$, $k_N = 200$ in truncated experiments), we recover various poses. We compare reconstructions using only the intrinsic shape difference (right of each pair) to reconstructions using intrinsic and extrinsic differences together (left of each pair); we also compare using a truncated basis for shape differences (second column) to using a full basis (third column). As a baseline, we compare to [Boscaini et al. 2015], which uses only intrinsic geometry (rightmost column). Reconstruction from intrinsic information shows considerable artifacts due to the non-uniqueness of the solution of the embedding problem. Our provably complete intrinsic/extrinsic description is much more stable and close to the solution. The truncation of the basis, discussed in §8.2, tends to smooth out the sharp creases as they are represented as high frequency features.

In Figure 14, we interpolate between frames of an animation sequence ($|V_M| = |V_N| = 1089$, $k_M = 100$, $k_N = 200$). After running a cloth simulation with coarse time steps, we compute the shape difference between subsequent frames ($t \in [0, 1]$). We

then use the method in §7.3 to construct plausible motion between the frames by interpolating linearly between the computed shape differences ($t = 0.5$). We further extrapolate the motion beyond the $t \in [0, 1]$ range to $t = 1.5$, effectively exaggerating the deformation between the frames. As expected, the extrinsic shape differences allow for reconstruction of largely isometric cloth motion.

Figure 15 illustrates a more challenging experiment ($|V_M| = 669$, $|V_N| = 1089$, $k_M = 60$, $k_N = 180$). In this case, we reconstruct the same cloth simulation sequence but vary the topology of the source and target meshes. Now it is impossible to pull back the deformation exactly to the new mesh topology, but we still reconstruct plausible motion, with the notable exception of artifacts near the boundary of the patch.

8.5 Timings

Figure 16 shows timings by stage for our pipeline, applied to meshes of various sizes and topologies. We employ a simplistic single-threaded implementation in MATLAB, using the Mosek toolbox [MOSEK ApS 2015] in the CVX library for convex optimization [Grant and Boyd 2014]; for this reason, the timings should be viewed as relatively pessimistic upper bounds. The “offset metric” meshes have more elements than their intrinsic counterparts thanks to the construction illustrated in Figure 6. The meshes in our experiments have relatively few vertices; an obvious and important next step for research will be to develop optimization techniques that can scale to larger models.

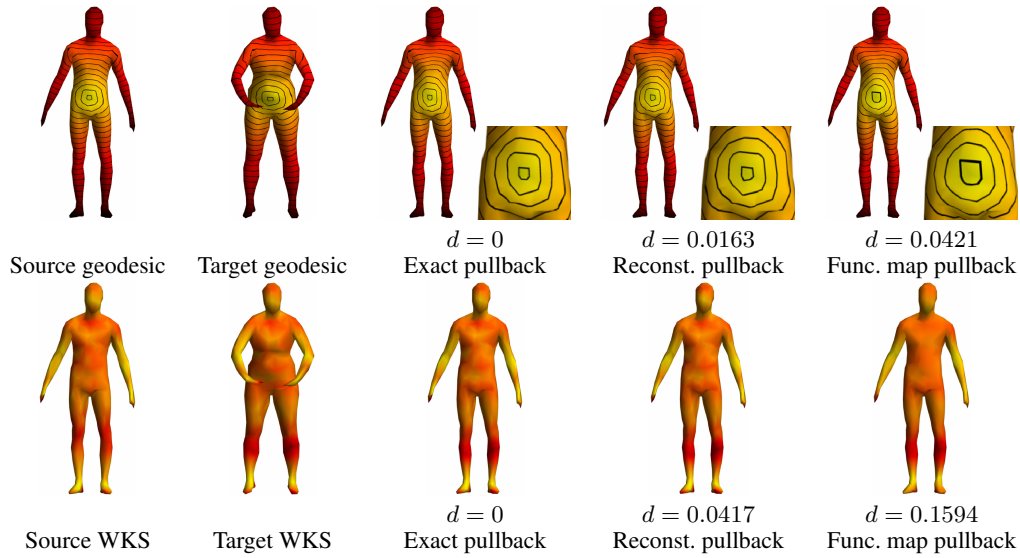


Fig. 12. Our technique can be used to recover the pullback metric and therefore compute geodesic distances without direct access to the target mesh. We compare three geodesic pullbacks on the source mesh: the exact pullback using point-to-point correspondence, the geodesic computed by reconstruction of the metric from the shape differences, and the pullback of the geodesic function using a functional map. For each pullback we compute the L^2 distance d to the exact version. Our method achieves better reconstructions than the direct usage of a functional map.

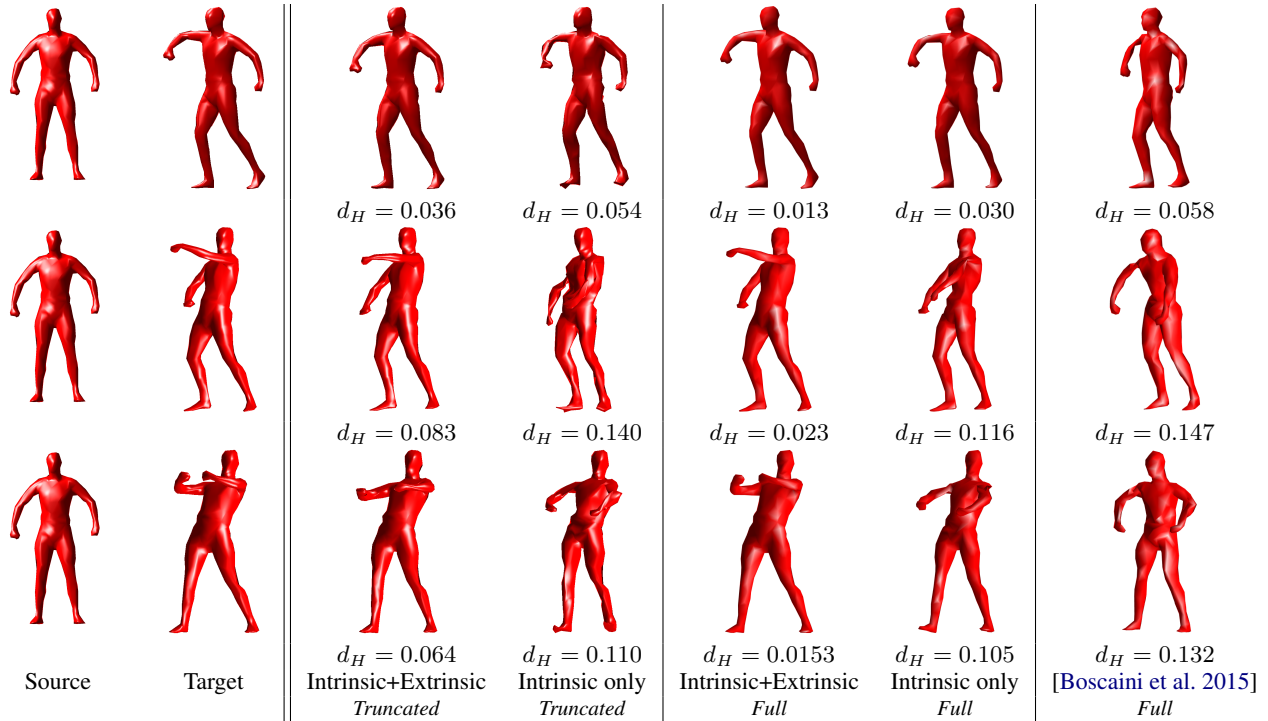


Fig. 13. Mesh recovery from a source mesh and shape differences, with (left) and without (middle) the extrinsic shape difference. Intrinsic mesh recovery using a concurrent method (right). The distance to the target d_H is measure by the Hausdorff distance on the prealigned point cloud.

9. DISCUSSION & CONCLUSION

In this paper, we introduced a new way to express intrinsic and extrinsic shape information through functional shape differences. Not only do we prove that discrete shape differences can be used to

recover shape, but we also extend to characterizing shapes up to rigid motion rather than isometry. Our four shape differences together—two intrinsic and two extrinsic—comprise a powerful description of shape that applies to a wide range of variability, including not only non-isometric shapes but also models obtained from physical

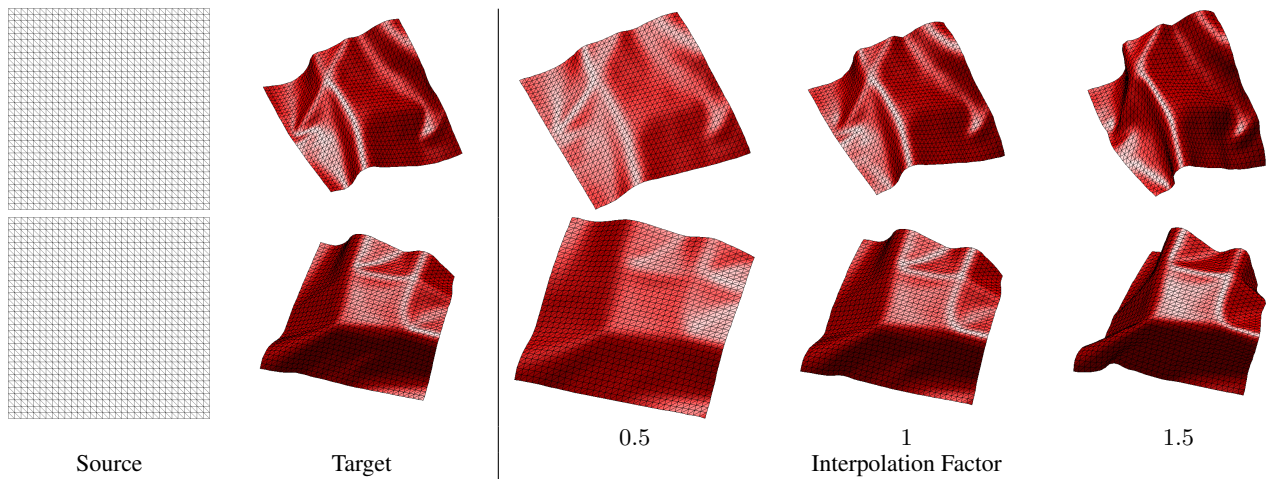


Fig. 14. Mesh recovery and interpolation from a source mesh and the intrinsic/extrinsic shape differences. The target meshes come from a cloth simulation sequence.

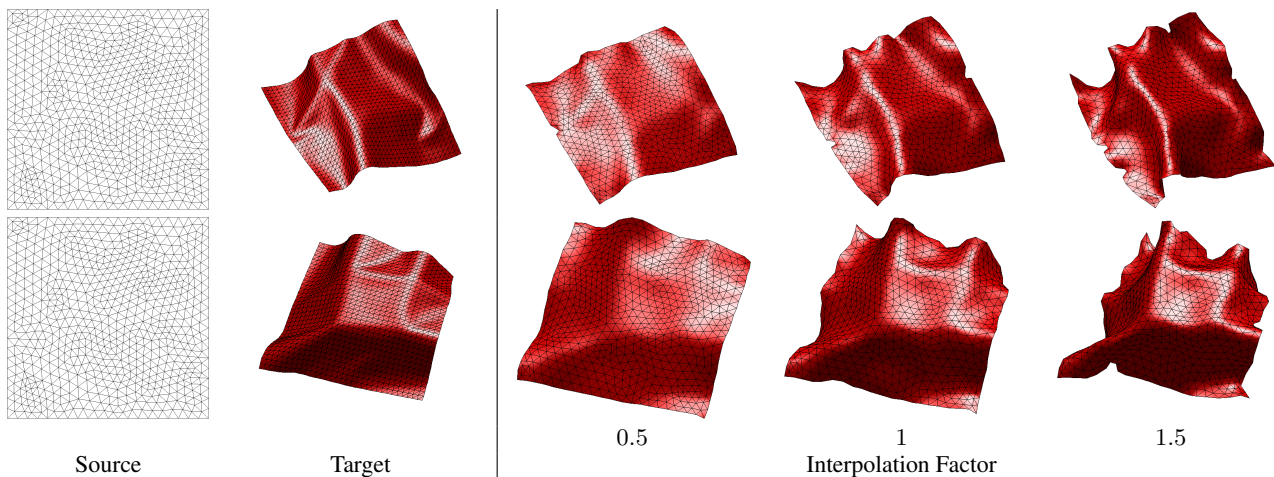


Fig. 15. Mesh recovery and interpolation. The source mesh has different connectivity than the target. The target meshes come from a cloth simulation sequence.

simulation and animation. We also show that the inverse problem of recovering shape structure from shape differences can be meaningful even in the under-determined truncated case.

While this work offers the possibility of direct application in pipelines for shape search, embeddings of shape space, and approximate reconstruction, it also suggests myriad avenues for future research. On the theoretical side, a better understanding of the effect of Laplace–Beltrami eigenfunction truncation may provide better guidance for the minimal-sized shape differences needed to reconstruct a shape; spectral truncation is a common part of the geometry processing pipeline, so any relevant theory would have the potential to affect understanding of many existing algorithms.

On the practical side, the primary limitation of our proposed reconstruction methods is the introduction of semidefinite constraints in computing the squared edge lengths ℓ ; multi-scale or lighter-weight optimization methods would enable application to larger-scale meshes. Furthermore, the regularization proposed for recovery of μ and ℓ in §7 is very generic and can be ineffective for noisy or highly truncated shape differences. Application of machine learning techniques may allow for the characterization of edge length

and triangle area distributions specific to a given class of shapes, considerably reducing the search space for our recovery algorithms.

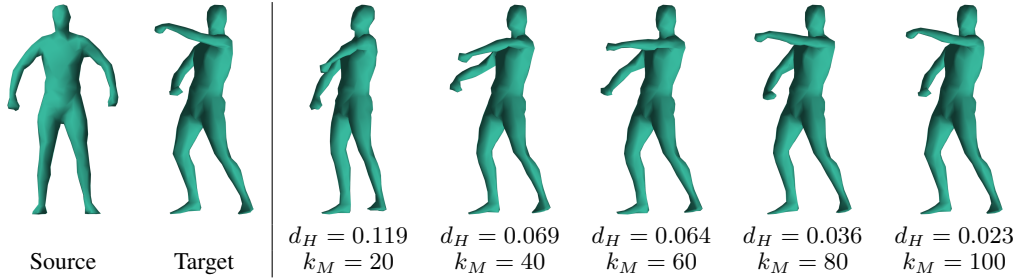
Figures 17 and 18 show examples illustrating these potential avenues for improving our pipeline. Figure 17 shows how reconstruction can fail when shape differences are over-truncated; stronger regularizers might fill in missing information when truncated shape differences are insufficient to recover edge lengths to high precision. Figure 18 shows results of shape reconstruction in the presence of noise. Here, we add noise directly to the shape difference matrix so that it no longer corresponds to an embedded surface. At some point, increasing k_N does not improve the reconstruction result, because noise in the entries dominates added high-frequency shape information.

ACKNOWLEDGMENTS

Omri Azencot participated in many discussions related to this work and contributed experiments and theory to early stages of our exploration; the authors express their gratitude for his insights. The anonymous reviewers of this paper provided extremely detailed

	# Eigenfunc.	Intrinsic metric					Offset metric					Reconst Tet
		V	F	E	Area	Edge	V	F	E	Area	Edge	
Human	100	502	1000	4500	4.7s	9.0s	1502	3000	4500	16.5s	310.9s	195.6s
Cloth HD	200	1089	2048	3136	20.6s	79.8s	3135	6016	9150	430.5s	1833.0s	912.6s
Cloth LD	60	669	1256	1924	6.0s	3.8s	1925	3688	5612	31.5s	73.2s	412.7s
Faces	180	588	1097	1687	8.5s	18.8s	1682	3208	4892	177.3s	709.6s	204.2s
Horse	160	752	1500	2250	10.1s	80.1s	2252	4500	6750	293.1s	666.6s	447.5s

Fig. 16. Performance measured on a 2015 iMac 3.3GHz.

Fig. 17. Example of failure in mesh recovery from a source mesh and shape differences. As the size of the shape difference increases more details are added to the reconstructed deformation. At $k_M = 100$ and above we achieve a high-quality reconstruction.

feedback, which helped refine and improve our exposition and experiments. The authors also wish to thank Fernando de Goes, Adrian Butscher, Davide Boscaini, Michael Bronstein, and Tom Funkhouser for discussions during the course of the project.

J. Solomon acknowledges the support of the NSF Mathematical Sciences Postdoctoral Research Fellowship (award number 1502435). E. Corman thanks the Direction Générale de l'Armement (DGA) for financial support. M. Ben-Chen received support from ISF grant 699/12, ISF equipment grant, Marie Curie CIG 303511. L. Guibas acknowledges NSF grants IIS 1528025 and 1546206, a Google Focused Research Award and a gift from the Adobe Corporation. M. Ovsjanikov acknowledges Marie-Curie CIG-334283, a CNRS chaire d'excellence, chaire Jean Marjoulet from École Polytechnique, FUI project TANDEM 2 and a Google Focused Research Award.

APPENDIX

A. PROOFS OF PROPOSITIONS

Proof of Proposition 1

Equation (1) gives A as a linear function of $\mu(\cdot)$. Hence, we must show that this formula is invertible.

First we show how to recover the area of a single triangle on M . By the second row of (1), given A we have the sum of triangle areas adjacent to any edge of M . If M has a boundary, we then know the areas of the boundary triangles. Otherwise, take v with odd valence, and enumerate its adjacent triangles as T_1, \dots, T_k for odd k . Since we know the sums of adjacent areas, we have a linear system to recover $\mu(T_1), \dots, \mu(T_k)$:

$$\begin{pmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & 1 & 1 \\ 1 & & & & & & 1 \end{pmatrix} \begin{pmatrix} \mu(T_1) \\ \mu(T_2) \\ \vdots \\ \mu(T_k) \end{pmatrix} = \begin{pmatrix} A_{12} \\ A_{23} \\ \vdots \\ A_{k1} \end{pmatrix}.$$

Consider carrying out forward substitution on the matrix. In each iteration, only the bottom row changes, from $(1, 0, \dots, 0, 1)$ to $(0, -1, 0, \dots, 0, 1)$, then to $(0, 0, 1, 0, \dots, 0, 1)$ and so on with alternating sign. When k is odd, in the last step the 1 is augmented to a 2, making the final row $(0, \dots, 0, 2)$. In other words, the matrix reduces to an upper triangular matrix with nonzero diagonal, which is invertible.

Hence, in either case we can recover $\mu(T)$ for at least one T . The remaining areas can be computed by flood filling outward from T ; given the area on one side of an edge and the sum of the adjacent areas, the adjacent area is recovered by subtraction.

Proof of Proposition 2

By construction $C(\ell; \mu)$ takes squared edge lengths ℓ and outputs the matrix C . Extracting elements of C corresponding to edges on M yields a linear operator $B: \mathbb{R}^{|E|} \rightarrow \mathbb{R}^{|E|}$ with matrix

$$B_{ij} = \frac{1}{8} \begin{cases} \mu(T_i)^{-1} + \mu(T'_i)^{-1} & \text{if } i = j \\ -\mu(T)^{-1} & \text{if } i, j \text{ are edges of } T \\ 0 & \text{otherwise.} \end{cases}$$

Here, indices i, j refer to edges on M ; for a given edge i , we label its adjacent triangles T_i and T'_i . Remark that B can be written as a weighted sum: $B = \sum_k \frac{1}{8} \mu(T_k)^{-1} B^k$, where each B^k is a matrix such that:

$$B^k_{ij} = \begin{cases} 1 & \text{when } i = j, \text{ and } i \text{ belongs to triangle } k. \\ -1 & \text{when } i, j \text{ are edges of triangle } k. \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that the intersection of the kernels of all B^k is empty, since B^k is non-singular when restricted to the values on edges of triangle k . Moreover, by considering the determinant of B as a multivariate polynomial with real coefficients, we conclude that B is either singular for any choice of values of $\mu(T_k)^{-1}$, or for a set of coefficients which has measure zero.

To complete the proof we note that if B is singular for any choice of values of $\mu(T_k)^{-1}$, then the matrix pencil $B = \sum_k a_k B^k$ is singular (i.e., B is singular for any choice of coefficients a_k). Using

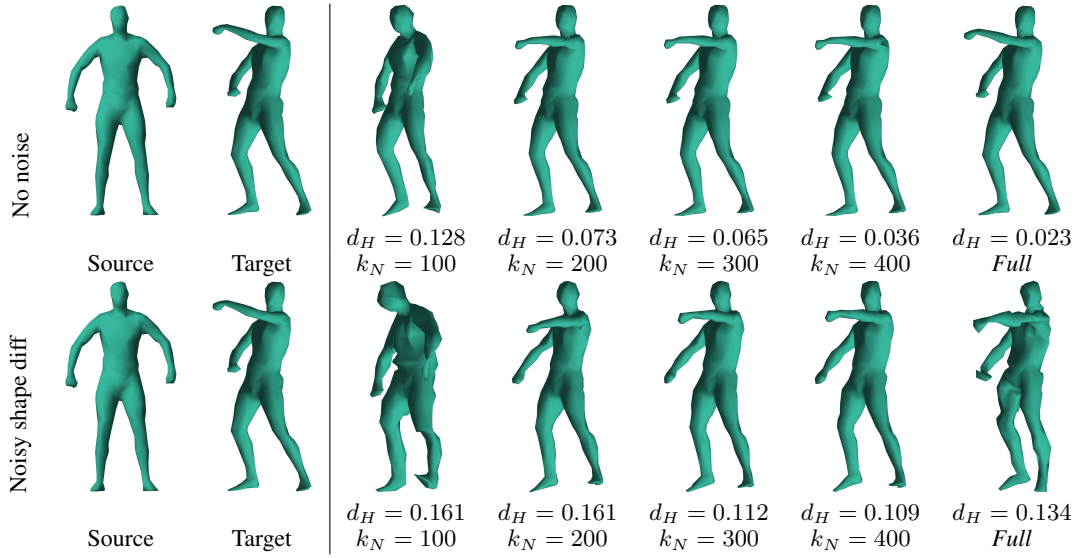


Fig. 18. Impact of the basis truncation on from a source mesh and *noisy* shape differences. In this experiment, we fix the number of basis functions on the source shape to $k_M = 100$ and reconstruct the embedding for various k_N . Top row: With no additional noise, the quality of the embedding increases with k_N . Bottom row: With added noise, larger k_N —which normally yields better transfer of high frequency deformation—does not increase the quality of the reconstruction. The noisy shape differences do not correspond to an actual embedding.

Lemma 3.4 from [Muhic and Plestenjak 2009] and the fact that B^k are symmetric, we see that in that case for every choice of a_k , there must exist a vector x such that $x^T B^k x = 0$ for every B^k , and $Bx = 0$. Now, given the values of x on some triangle, this means that its values on the adjacent triangle are either uniquely determined by the corresponding two equations (one linear, one quadratic), or these equations cannot be satisfied. By inspecting the resulting equalities, it is easy to see that at least two of the values on every triangle must be equal, and by considering any closed loop of triangles, these equations cannot be consistent for every choice of weights a_k . Thus, B cannot be a singular matrix pencil, and therefore B is invertible for almost any choice of values $\mu(T_k)^{-1}$.

Proof of Proposition 3

The results come from direct computation (assume the following expressions are evaluated at $t = 0$):

$$\begin{aligned}
 \frac{\partial g_{ij}}{\partial t} &= \frac{\partial}{\partial t} \left\langle \frac{\partial F_t}{\partial x_i}, \frac{\partial F_t}{\partial x_j} \right\rangle \text{ by definition of } g_{ij} \\
 &= \left\langle \frac{\partial n}{\partial x_i}, \frac{\partial F_t}{\partial x_j} \right\rangle + \left\langle \frac{\partial F_t}{\partial x_i}, \frac{\partial n}{\partial x_j} \right\rangle \text{ by definition of } F_t \\
 &= 2h_{ij} \text{ by definition (and symmetry) of } h_{ij}. \\
 \frac{\partial \mu}{\partial t} &= \frac{\partial \sqrt{\det g}}{\partial t} \text{ by definition of } \mu \\
 &= \frac{\det g}{2\sqrt{\det g}} g^{ij} \frac{\partial g_{ij}}{\partial t} \text{ by the chain rule} \\
 &= \sqrt{\det g} g^{ij} h_{ij} \text{ from the previous result} \\
 &= H\mu \text{ by definition of } H, \mu. \\
 \frac{\partial}{\partial t} g(\nabla \phi_t, \nabla \psi_t) &= \frac{\partial}{\partial t} \left[g_{ij} g^{ik} g^{j\ell} \frac{\partial \phi_t}{\partial x_k} \frac{\partial \psi_t}{\partial x_\ell} \right] \text{ by definition of } \nabla
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{\partial}{\partial t} \left[g^{ij} \frac{\partial \phi_t}{\partial x_i} \frac{\partial \psi_t}{\partial x_j} \right] \text{ by multiplying } g \text{ and } g^{-1} \\
 &= \frac{\partial g^{ij}}{\partial t} \frac{\partial \phi_t}{\partial x_i} \frac{\partial \psi_t}{\partial x_j} \text{ since the rest is constant} \\
 &= -g^{ik} \frac{\partial g_{k\ell}}{\partial t} g^{\ell j} \frac{\partial \phi_t}{\partial x_i} \frac{\partial \psi_t}{\partial x_j} \text{ by differentiating } g^{-1} \\
 &= -2g^{ik} h_{k\ell} g^{\ell j} \frac{\partial \phi_t}{\partial x_i} \frac{\partial \psi_t}{\partial x_j} \text{ by the first result} \\
 &= -2h(\nabla \phi_t, \nabla \psi_t) \text{ by definition of } \nabla.
 \end{aligned}$$

Proof of Proposition 4

The previous propositions show that μ , ν , μ_t , and ν_t are (almost always) sufficient to recover the edge lengths of the base and offset surfaces. The remaining edges of the canonical thickening are between the inner and outer layers and are recoverable essentially by convention. Specifically:

- The edges along surface normals are length t by definition.
- The bottom edge lengths of the “upward-facing” tetrahedra (Figure 6) are known because they are on the base surfaces. The remaining edges of these tetrahedra can be computed because the upward-facing tetrahedron is generated via normal offset from the barycenter of the base triangle by a distance t .
- “Mesh edge” tetrahedra are adjacent to “upward-facing” tetrahedra and outer faces of the thickening and hence have edge lengths fixed by their neighbors’ construction.
- Similarly, “downward-facing” tetrahedra have one normal edge of length t , and the remaining edges are on the outer surface or adjacent to an “upward-facing” tetrahedron.

The embedding of a single oriented tetrahedron is fixed up to rigid motion given its edge lengths, so the proposition follows by gluing

the tetrahedra of the canonical thickening according to the topology of the construction.

Proof of Proposition 5

We denote (e_1, e_2, e_3) the canonical basis and the indices $\{i, j, k\} \in \{1, 2, 3\}$.

If $\sqrt{x_k} \leq \sqrt{x_i} + \sqrt{x_j}$, then there exist three points (v_1, v_2, v_3) which define an embedding of a triangle. Let E be the matrix with columns $v_3 - v_2, v_1 - v_3$ and $v_2 - v_1$ then $\mathbf{E} = E^T E$. The matrix \mathbf{E} is therefore positive semidefinite.

Since \mathbf{E} is symmetric positive semidefinite, the Cauchy-Schwartz inequality holds. Expanding the expression $(e_i + e_j)^T \mathbf{E}(e_i + e_j)$ yields

$$\begin{aligned} (e_i + e_j)^T \mathbf{E}(e_i + e_j) &= x_i + x_j + 2e_i^T \mathbf{E}e_j \\ &\leq x_i + x_j + 2\sqrt{x_i x_j} \\ &\leq (\sqrt{x_i} + \sqrt{x_j})^2. \end{aligned}$$

At the same time, a direct computation shows $(e_i + e_j)^T \mathbf{E}(e_i + e_j) = x_k$ which implies that $\sqrt{x_k} \leq \sqrt{x_i} + \sqrt{x_j}$.

Proof of Proposition 6

Starting with the second statement, direct computation of the determinant shows

$$4 \det(\mathbf{E}_k) = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

When the x_i 's are squared triangle edge lengths, this is a formulation of Heron's area formula.

The first statement is proved using a well-known theorem on positive block matrices (property of Schur complements) [Boyd and Vandenberghe 2004]: $\mathbf{E} \succeq 0$ if and only if $\mathbf{E}_3 \succeq 0$ and

$$2x_3 - \begin{pmatrix} x_2 - x_1 - x_3 \\ x_1 - x_2 - x_3 \end{pmatrix}^T \mathbf{E}_3^{-1} \begin{pmatrix} x_2 - x_1 - x_3 \\ x_1 - x_2 - x_3 \end{pmatrix} \geq 0.$$

Notice that

$$\mathbf{E}_k \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -\frac{1}{2} \begin{pmatrix} x_j - x_i - x_k \\ x_i - x_j - x_k \end{pmatrix},$$

and hence the first condition is met whenever $x_3 \geq 0$. Moreover, \mathbf{E}_3 is a 2×2 matrix and therefore is positive semidefinite if and only if $x_1 \geq 0, x_2 \geq 0$ and $\det(\mathbf{E}_3) \geq 0$.

REFERENCES

- Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV Workshops*. 1626–1633.
- Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. 2009. Spatial Deformation Transfer. In *Proc. SCA*. 67–74.
- O. Bonnet. 1867. Mémoire sur la theorie des surfaces applicables sur une surface donnée. *Journal de l'École Polytechnique* 25 (1867), 313–151.
- Davide Boscaini, Davide Eynard, Drosos Kourounis, and Michael M. Bronstein. 2015. Shape-from-Operator: Recovering Shapes from Intrinsic Operators. *Computer Graphics Forum* (2015).
- M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. 2010. *Polygon Mesh Processing*. AK Peters.
- S.P. Boyd and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.
- Susanne Brenner and Ridgway Scott. 2007. *The mathematical theory of finite element methods*. Vol. 15. Springer.
- Haim Brezis. 2010. *Functional analysis, Sobolev spaces and partial differential equations*. Springer.
- Manfredo P. Do Carmo. 1976. *Differential Geometry of Curves and Surfaces*. Pearson.
- David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted Delaunay triangulations and normal cycle. In *Proc. SoCG*. 312–321.
- Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. 2014. Supervised Descriptor Learning for Non-Rigid Shape Matching. In *ECCV 2014 Workshops*. Lecture Notes in Computer Science, Vol. 8928. 283–298.
- Etienne Corman, Maks Ovsjanikov, and Antonin Chambolle. 2015. Continuous Matching via Vector Field Flow. *Computer Graphics Forum* 34, 5 (2015), 129–139.
- Fernando de Goes, Pooran Memari, Patrick Mullen, and Mathieu Desbrun. 2014. Weighted Triangulations for Geometry Processing. *ACM Trans. Graph.* 33, 3 (June 2014), 28:1–28:13.
- Nadav Dym and Yaron Lipman. 2016. Exact Recovery with Symmetries for Procrustes Matching. *arXiv:1606.01548* (2016).
- Michael Eigensatz, Robert W. Sumner, and Mark Pauly. 2008. Curvature-Domain Shape Processing. *Computer Graphics Forum* 27, 2 (2008), 241–250.
- Stefan Fröhlich and Mario Botsch. 2011. Example-Driven Deformations Based on Discrete Shells. *Computer Graphics Forum* 30, 8 (2011), 2246–2257.
- Herman Gluck. 1975. Almost all simply connected closed surfaces are rigid. In *Geometric Topology*. Lecture Notes in Mathematics, Vol. 438. 225–239.
- Michael Grant and Stephen Boyd. 2014. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. <http://cvxr.com/cvx>. (March 2014).
- T. Hoffmann, A. O. Sageman-Furnas, and M. Wardetzky. 2014. A discrete parametrized surface theory in \mathbb{R}^3 . *ArXiv e-prints* (Dec. 2014).
- Qi-Xing Huang and Leonidas Guibas. 2013. Consistent Shape Maps via Semidefinite Programming. *Computer Graphics Forum* 32, 5 (2013), 177–186.
- Wonhyung Jung, Hayong Shin, and Byoung K Choi. 2004. Self-intersection removal in triangular mesh offsetting. *Computer-Aided Design and Applications* 1, 1–4 (2004), 477–484.
- Artiom Kovnatsky, Michael M. Bronstein, Xavier Bresson, and Pierre Vandergheynst. 2015. Functional Correspondence by Matrix Completion. In *Proc. CVPR*.
- A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. 2013. Coupled quasi-harmonic bases. *Computer Graphics Forum* 32, 2.4 (2013), 439–448.
- E. Kreyszig. 1959. *Differential Geometry*. Dover.
- Jan De Leeuw, In J. R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem. 1977. Applications of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*. 133–146.
- Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. 35–57.
- MOSEK ApS. 2015. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28)*. <http://docs.mosek.com/7.1/toolbox/index.html>
- Andrej Muhic and Bor Plestenjak. 2009. On the singular two-parameter eigenvalue problem. *Electron. J. Linear Algebra* 18 (2009), 42.
- Przemyslaw Musialski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2015. Reduced-order Shape Optimization Using Offset Surfaces. *ACM Trans. Graph.* 34, 4 (July 2015), 102:1–102:9.

- 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 (July 2012), 30:1–30:11.
- Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. 2014. Frame Fields: Anisotropic and Non-orthogonal Cross Fields. *ACM Trans. Graph.* 33, 4 (July 2014), 134:1–134:11.
- N.M. Patrikalakis and T. Maekawa. 2009. *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer.
- Jonathan Pokrass, Alexander M Bronstein, Michael M Bronstein, Pablo Sprechmann, and Guillermo Sapiro. 2013. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 459–468.
- S. Rosenberg. 1997. *Laplacian on a Riemannian manifold*. Cambridge University Press.
- Raif M. Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. 2013. Map-based Exploration of Intrinsic Shape Differences and Variability. *ACM Trans. Graph.* 32, 4 (July 2013), 72:1–72:12.
- Francisco-Javier Sayas. 2008. A gentle introduction to the Finite Element Method. *Lecture notes, University of Delaware* (2008).
- Gilbert Strang and George J Fix. 2008. *An analysis of the finite element method* (2 ed.). Vol. 212. Wellesley-Cambridge.
- Y. Wang, B. Liu, and Y. Tong. 2012. Linear Surface Reconstruction from Discrete Fundamental Forms on Triangle Meshes. *Comput. Graph. Forum* 31, 8 (Dec. 2012), 2277–2287.
- Wei Zeng, Ren Guo, Feng Luo, and Xianfeng Gu. 2012. Discrete Heat Kernel Determines Discrete Riemannian Metric. *Graph. Models* 74, 4 (July 2012), 121–129.

Received October 2015; accepted December 2016