Eurographics Symposium on Geometry Processing 2019
D. Bommes and H. Huang
(Guest Editors)

Volume 38 (2019), Number 5

Limit Shapes – A Tool for Understanding Shape Differences and Variability in 3D Model Collections

Ruqi Huang[1], Panos Achlioptas[2], Leonidas Guibas[2], Maks Ovsjanikov[1]

[1]LIX, École Polytechnique, CNRS [2]Stanford University

Abstract

We propose a novel construction for extracting a central or limit shape in a shape collection, connected via a functional map network. Our approach is based on enriching the latent space induced by a functional map network with an additional natural metric structure. We call this shape-like dual object the limit shape and show that its construction avoids many of the biases introduced by selecting a fixed base shape or template. We also show that shape differences *between real shapes and the limit shape can be computed and characterize the unique properties of each shape in a collection – leading to a compact and rich shape representation. We demonstrate the utility of this representation in a range of shape analysis tasks, including improving functional maps in difficult situations through the mediation of limit shapes, understanding and visualizing the variability within and across different shape classes, and several others. In this way, our analysis sheds light on the missing geometric structure in previously used latent functional spaces, demonstrates how these can be addressed and finally enables a compact and meaningful shape representation useful in a variety of practical applications.*

CCS Concepts
• ***Computing methodologies*** → ***Spectral methods; Shape analysis;***

1. Introduction

Detecting, quantifying and analyzing variability in shape collections is a fundamental task in computer graphics and geometry processing, with applications across multiple domains, including in statistical shape analysis [ASK*05, BRLB14, HSS*09], shape exploration [KLM*12, ROA*13, KvKSHCO15], shape correspondences [HWG14] and co-segmentation [WAvK*12]. A key question that arises in all techniques for extracting variability is the choice of right *shape representation*, which can reveal the structure of each shape in the context of the collection while also being compact and easy to manipulate, enabling efficient shape analysis and processing. The majority of existing techniques dedicated to extracting variability in a collection are based on first selecting a template (or base) shape and considering the changes on all other shapes with respect to this template – this is the standard practice in medical domains where the reference shape is often referred to as an "atlas" (e.g., in brain anatomy) [GM98]. In computer graphics this approach is common both in shape reconstruction and in statistical shape analysis [ASK*05, BRLB14, HSS*09, KSKL13], but also in shape exploration (e.g., [KLM*12, OLGM11, KLM*13, ROA*13] among many others) where the template is often constructed by either simplifying some fixed base shape or by using shape abstractions derived from collections of parts and their relations.

Although easy and intuitive, template-based shape exploration

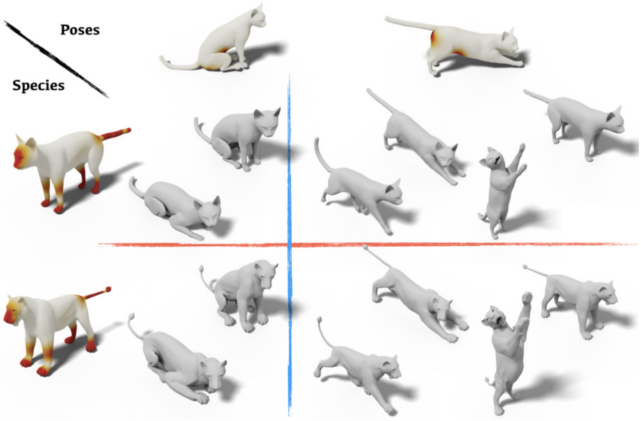


Figure 1: *Cross-collection variability between a set of cats and lions, as detected by our algorithm. Different distinctive regions are detected for different clusterings: when animals are clustered regarding their species, the paws, snouts and tails are highlighted (left column); when the animals are clustered according to their poses, the hips and abdomens are highlighted (top row).*

and analysis has obvious limitations when shape variability is large and no single prototype adequately models all given shapes. But even in settings of more modest variation, there are significant issues: first, the choice of the template can significantly affect the results in terms of the types of variability that is detected and highlighted. Second, considering the variability with respect to a fixed base shape can make it difficult to reveal cross-class variability that becomes apparent only when comparing all pairs of shapes in the collection. Third, even when the base shape is given, the exact choice of encoding for the variability remains crucial. For example, while simple techniques based on the displacement of each template vertex might be relevant for reconstruction and statistical shape analysis, their use is very limited in the context of learning, as they are not invariant to even basic rigid motions.

In this paper we present a novel approach to encoding shapes in the context of a collection that helps overcome many of the above limitations. Specifically, starting from a collection of shapes with some soft (functional) maps between them, we show how consistent latent spaces that have previously been used for improving map quality can also be exploited to reveal the geometric variability in the collection, without relying on a base or template shape (Figure 1 gives an example of how our approach highlights the regions that are distinctive between the cats and lions), or assuming a particular (e.g., star-shaped) topology of the functional map network. The proposed approach is based on a novel analysis of latent spaces, which demonstrates that after proper regularization they can be endowed with natural, unbiased geometric structure. This constructs a dual object that need not correspond to a real shape in 3D, but it has shape-like properties.

We call this dual latent shape the *limit shape* of the collection, both because it is the limit of a diffusion process implicit in Theorem 4.1 below and because it is defined in the spirit of the notion of limits and co-limits in homological algebra and category theory. Namely, limits are are categorical constructions that factorize morphisms (maps) from an object to each member of a class, by passing through a universal intermediate limit object that is then itself mapped to all individual objects in the class. In co-limits, morphism directions are reversed and maps emanating from the class objects to another object are factorized through the co-limit [Rot08, BW90]. Furthermore, limit shapes can be used together with the notion of shape differences introduced in [ROA*13] to construct a representation for each shape in the collection. We call the shape difference between a real shape and the limit shape the *characteristic difference* of the shape, as it captures what is unique and special about that particular shape in the context of the collection. Moreover, we show how the *algebraic nature* of this limit shape construction can be exploited to detect characteristic features not only for individual shapes but also to highlight features that differentiate two shape collections.

Contributions. To summarize, our main contributions are:

- We describe how latent functional spaces can be endowed with natural geometric (metric and measure) structure, giving rise, for the first time, to a well-defined notion of a "limit shape" that abstracts an entire shape collection.
- We define characteristic shape differences between real and limit shapes and show how such differences capture the individuality of each shape and provide a shape representation that can be used for detailed shape analysis, without assuming a particular topology of the map network.
- We provide tools for a nuanced understanding of shape viability, including the separation of the different types of variability present within and across shape sub-collections.
- We demonstrate how limit shapes can mediate the construction of improved functional maps between non-isometric shapes.

2. Related Work

Template-based Shape Analysis and Exploration. Analyzing shape collections by variability around a template shape has a rich and vast history going back to D'Arcy Thompson's classic "On growth and form" [T*17], which has inspired Kendall's shape space theory [Ken89]. In computer graphics, apart from those mentioned in the introduction, shape spaces based on template variation are ubiquitous in statistical shape analysis, e.g., for defining 3D morphable models [BV99, ACP03]. Although in most cases the presence of a shape template is assumed to be given *a priori* [GSDP*18], simultaneous template construction and fitting techniques have been used for both reconstruction [WJH*07, WAO*09, TZL*12] and exploration [KLM*13], among many others. While pervasive, template-based methods also have a well-known limitation in that the choice of the template model can introduce bias in the kinds of variability that are revealed. Common selection techniques include using a particular (median) shape in a collection that is as close as possible to a centroid, or constructing a new template shape by point-wise averaging (e.g., [JDJG04]).

Our approach avoids the construction of an explicit template shape, and replaces it with an implicit template obtained via the analysis of latent functional space, which removes the bias in the template shape selection and avoids the expensive geometric (embedded 3D shape) template construction that can require non-linear optimizations in Riemannian manifolds, such as Kärcher means.

Shape Analysis with Functional Maps. Our approach takes as input a collection of shapes with soft (functional) maps between them. In this, we follow the recent line of work on shape analysis with soft maps, similar to [SNB*12, KLM*12, ROA*13]. Namely, we use the formalism of *functional maps* introduced originally in [OBCS*12] and extended significantly in follow-up works, including [KBB*13, HWG14] among others (see [OCB*17] for a recent overview).

Although originally proposed as a computational tool for shape matching, follow-up works have also shown its utility in shape analysis and exploration, starting with map visualization [OBCCG13], detection and encoding of *shape differences* [ROA*13], and co-segmentation and co-analysis [HWG14] among others. The advantage of these techniques is that they only require approximate functional maps, which are much easier to compute than precise (point-to-point) correspondences. Nevertheless, existing methods such as [OBCCG13, ROA*13] also follow the spirit of template-based techniques and assume the presence of a single *base* shape with respect to which variability is captured. A recent method introduced in [HO17] has tried to lift this assumption but is still restricted to revealing global variability within a single collection.

We extend these techniques first by proposing a template-free analysis and exploration framework using functional maps and second by proposing techniques for detecting and highlighting *cross*-collection variability.

Latent Functional Spaces. A key building block in our approach is the use of so-called latent functional spaces, which are closely related to map synchronization [WS13] and which have been used for computing *consistent functional maps* in shape and image collections [HWG14, WHG13, WHOG14]. On the other hand, a recent approach [GSDG18] also takes the idea of constructing latent objects in the functional space, which is proposed for transporting descriptors across shapes in collections. One of our key contributions is to show that in addition to providing a powerful computational method for map inference, latent functional spaces also allow to reveal variability in shape collections and also to define a compact and informative shape representation.

3. Preliminaries, Notation and Problem Setup

Throughout our work, we assume to be given a collection of related 3D shapes and a set of functional maps among some shape pairs. Our main goal is to develop a theoretical foundation for a novel representation for the shapes in the collection, and to show how this representation can be effectively used in practical applications.

Specifically, we assume as input a set of shapes $\mathcal{S} = \{S_i\}_{i=1}^n$, typically represented as triangle meshes. To each shape S_i we associate a Laplacian matrix $\mathcal{L}_i$ discretized using the standard cotangent weight scheme [PP93, MDSB03], s.t. $\mathcal{L}_i = M_i^{-1} L_i$, where L_i is the cotangent weight (also called a stiffness) matrix and M_i is the diagonal lumped area (mass) matrix. These can be seen to encode the metric and measure structure of each shape respectively. We denote by Λ_i the diagonal matrix of the first k eigenvalues and by Φ_i the matrix of eigenfunctions of $\mathcal{L}_i$, such that $L_i\Phi_i = M_i\Phi_i\Lambda_i$. Typically, k is fairly small (60 in most of our experiments), giving rise to compact functional representations of shapes as detailed below.

We also assume to be given a set of functional maps C_{ij} between some pairs of shapes S_i, S_j. In the simplest case, if we are given a point-to-point map between e.g. points on S_j and S_i, we can set $C_{ij} = \Phi_j^T M_j \Pi_{ij} \Phi_i$, where Π_{ij} is the binary matrix encoding the map T using: $\Pi_{ij}(p,q) = 1$ if $T(p) = q$ and 0 otherwise. Alternatively the functional maps C_{ij} can be estimated automatically using existing mapping techniques (see, e.g. [OCB*17] for an overview). The key feature of functional maps C_{ij} is that they allow to transfer real-valued functions defined on different shapes, by "translating" coefficients expressed in different bases. Thus if f is a vector representing the coefficients of a function on shape i in the basis Φ_i, then the coefficients of the corresponding function on shape j in the basis Φ_j are given simply as $C_{ij}f$.

Using the input maps C_{ij} we build a functional map network (FMN) on the collection $\mathcal{S}$, by constructing a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the i^{th} vertex in $\mathcal{V}$ corresponds to the functional space on S_i, and the edge $(i,j) \in \mathcal{E}$ if we are given a functional map C_{ij}. We assume that this network is symmetric ($(i,j) \in \mathcal{E}$ if and only if $(j,i) \in \mathcal{E}$) and is *connected* so that there exists at least one path consisting of the edges in $\mathcal{E}$ between any pair of vertices in $\mathcal{V}$.

Shape Differences Our shape representation is based on the shape differences introduced in [ROA*13], which characterize shape deformations by encoding the changes in inner products of functions. Namely, given shapes S_i, S_j and a functional map C_{ij} in the reduced basis, the authors introduce the area-based and the conformal shape differences D^A, D^C, which using our notation can be expressed as:

$$D_{ij}^A = C_{ij}^T C_{ij}, \tag{1}$$

$$D_{ij}^C = \Lambda_i^+ C_{ij}^T \Lambda_j C_{ij}, \tag{2}$$

where $^+$ is the Moore-Penrose pseudo-inverse. Intuitively D_{ij} is a linear operator, which once again, can be represented as a matrix of size $k \times k$, and which encodes the difference or distortion induced by a map C_{ij} (see Figure 2 and Eq.(4) in [ROA*13]).

The key limitation of shape difference operators for shape collection analysis, is that they require a choice of a base shape S_i and consider only *directional* changes, from shape S_i to other shapes, making it impossible to use them given an arbitrary (non star-shaped) FMN. Thus, one of our goals is to extend this construction to the case of shape collections without assuming a fixed base shape. We achieve this by exploiting the formalism of latent functional spaces [WHG13], which has been proposed for improving the consistency of functional maps.

Latent Spaces In [WHG13], the consistency of a FMN $\mathcal{G}$ is defined as the deviation of the compositions of the functional maps along cycles from the identity: ideally one would expect $C_{i_1,i_k} C_{i_k,i_{k-1}} \cdots C_{i_2,i_1} = I$, where $(i_1, i_2 \cdots i_k, i_1)$ is any cycle in $\mathcal{G}$. The authors further propose to extract a set of *consistent latent bases* Y_i on S_i such that $C_{ij}Y_i \approx Y_j, \forall\, i,j$ (since exact consistency is rare in the reduced basis), and use them to refine the quality (consistency) of functional maps. The latent bases Y_i can be thought of either as functions on S_i, or as functional maps from some *latent shape* to each shape S_i. Then, a map from S_i to S_j can be factored into a map from i to the latent shape and then to j via: $C_{ij} \approx Y_j Y_i^{-1}$. The exact factorizations then implies the exact consistency. While useful as a tool for improving functional maps, the exact structure of latent shapes is still not fully understood, and they have so far not been used for representing and analyzing shapes in a collection.We refer readers to Appendix B for more details.

4. Latent Representation

In this section, we first show how latent shapes can be endowed with geometric structure, and be made more stable, through an extra regularization, and then define a latent space shape representation.

4.1. Canonical Latent Basis and the Limit Shape

Our first key observation is that the latent shape plays the role of an "average shape" in analyzing shape collections – a shape-like object that represents the entire collection, and which can be endowed with a natural geometric structure. Crucially, unlike existing approaches, for example in computational anatomy [You10] that consider building templates or average shapes, we characterize the latent shape *directly in the functional domain*, without attempting to embed it in the ambient space.

The following theorem establishes the connection between the

ALGORITHM 1: Computing the Limit Shape by Constructing a Canonical Consistent Latent Basis (CCLB)

```
input  : A set of consistent latent basis $\{Y_i\}_{i=1}^n$ obtained
         from a shape collection $S$ and associated FMN $\mathcal{G}$.
         The eigenbasis $\Phi_i$, eigenvalues $\Lambda_i$ on each $S_i$.
output: A set of canonical consistent latent basis $\{\tilde{Y}_i\}_{i=1}^n$,
         the eigenbasis $\Phi_0$ and the spectrum $\Lambda_0$, for the
         limit shape.
(1) Compute the eigen-decomposition of $E = \sum_i Y_i^T \Lambda_i Y_i$ so that
    $EU = U\Lambda$ and let $\tilde{Y}_i = Y_i U$.
(2) Let $\Phi_0 = \Phi_i \tilde{Y}_i$ for an arbitrary $i$, and $\Lambda_0 = \Lambda$ from the
    previous step.
```

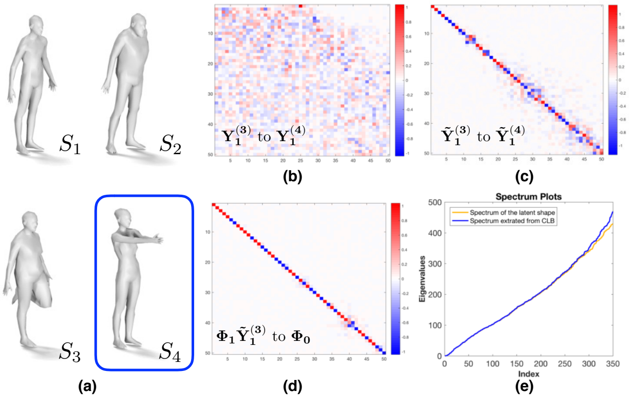


Figure 2: *(a) Input shapes, where S_4 is an additional shape; (b) the transformation matrix between the standard latent basis S_1 computed with and without S_4; (c) the same transformation matrix but between canonical latent bases. (d) The transformation matrix between the computed latent basis and the theoretical ground-truth stated in Theorem 4.1 when expressing functional maps in a reduced basis (e) the computed spectrum and the theoretical ground-truth (see the text for details).*

consistent latent basis (CLB) and the geometry of the latent shape, while at the same time highlighting the limitations of the previously used approaches for constructing latent bases:

Theorem 4.1 *Given a collection of discrete 3D shapes in* 1-1 *vertex correspondence and sharing the same mesh connectivity, and a consistent FMN $\mathcal{G}$, in which the functional maps are represented in the eigenbasis Φ_i on each S_i. Let Y_i be the consistent latent basis satisfying the conditions: $C_{ij}Y_i = Y_j, \forall (i,j) \in \mathcal{G}, s.t. \sum_i Y_i^T Y_i = I$, and $\sum_i Y_i^T \Lambda_i Y_i = \Lambda$, where Λ is a diagonal matrix. Then, the eigenbasis Φ_0 of the latent shape whose metric and measure are given by $L = \frac{1}{n}\sum_i L_i, M = \frac{1}{n}\sum_i M_i$, i.e. $L\Phi_0 = M\Phi_0\Lambda_0$, can be recovered as $\Phi_0 = \Phi_i Y_i$ for any i.*

This theorem (proved in Appendix A) suggests that the CLB carries information about the "average" geometry in the collection, given, in the full basis, by the average metric and measure matrices. We stress that Theorem 4.1 is primarily of theoretical interest: the assumption of the identical triangulations and of the exact consistency presented in the full basis are for the explicit characterization (i.e., L, M) of the theoretical averaging shape.

Role of Proper Regularization. As detailed in Appendix B, [WHG13] proposes to compute CLB by solving the optimization problem $\min_Y \|C_{ij}Y_i - Y_j\| s.t. \sum_i Y_i^T Y_i = I$. Geometrically, and in light of Theorem 4.1, this corresponds to only averaging *the measure* of the shapes, which leads to metric ambiguity. This can result in significant instabilities in the extracted latent basis. We demonstrate this effect in Figure 2. Namely, we compute the CLB on S_1 using two shape collections: $Y_1^{(3)}$ computed from $\{S_i\}_{i=1}^3$, and $Y_1^{(4)}$ computed from $\{S_i\}_{i=1}^4$. Figure 2(b) shows the change of basis matrix between these two settings, which has noisy off-diagonal entries, suggesting significant perturbations of the latent shape.

To overcome such instability, we propose to construct a *canonical* consistent latent basis (CCLB) by introducing an extra normalization which forces $\sum_i Y_i^T \Lambda_i Y_i$ to be a diagonal matrix, and which corresponds in Theorem 4.1 to averaging the *metric* on the latent shape. With this additional normalization, the change of basis matrix between latent bases with and without shape S_4 shown in Figure 2(c) is much closer to a diagonal one than in Figure 2(b). The details of this construction are given in Algorithm 1. The extra normalization incorporates the metric information, yielding what we call the *limit shape* of the collection.

Applicability and approximation of CCLB in practice We argue that Algorithm 1 is well-defined: first, as reviewed in Appendix B the CLB computed from [WHG13] is completely determined by the input FMN $\mathcal{G}$; secondly, it follows from the computation of Y_i that $\sum_i Y_i^T Y_i = I$, and since U computed in Step (1) is orthonormal, $\sum_i \tilde{Y}_i^T \tilde{Y}_i = I$ is guaranteed; lastly, the well-definedness of Φ_0 comes from the consistency, namely, $\Phi_i \tilde{Y}_i \approx \Phi_j C_{ij} \tilde{Y}_i \approx \Phi_j \tilde{Y}_j$.

It is worth noting that, though the theoretical assumptions of Theorem 4.1 are typically not satisfied in practice, Algorithm 1 can be still implemented without assuming uniform triangulation nor consistency in full basis. In Figure 2 (d), we also show the proximity between the eigenbasis/spectrum of the limit shape recovered from functional maps in the reduced basis and the theoretical ground truth (computed using the identical triangulation shared by the 4 shapes in Figure 2(a)) . Namely, Figure 2(d) shows the transformation matrix between the first 50 computed eigenbasis, when functional maps are represented in a reduced basis of size 400 and the theoretical ground-truth, given by the exact averaging of the metric and measure. At the same time, Figure 2(e) shows the eigenvalues in the two cases. Moreover, we show in Figure 12 in Appendix B, that the eigenvalues and eigenfunctions are both well approximated even in the case where the shapes do not share the same triangulation.

Hereafter, we always use the *canonical consistent latent basis* in all the following formulations and applications, and denote it by Y_i, to simplify notation. This is our primary representation of the limit shape of the collection. Though the limit shape itself may not be embeddable (even under the ideal assumptions of Theorem 4.1), its basis can be transported onto any individual shape S_i of the collection via Y_i, which can be thought of as functional maps from the

```
ALGORITHM 2: Construction of the Area and Conformal
Characteristic Shape Difference Operators
input : Shape collection $S$ and associated FMN $\mathcal{G}$. The
        eigenbasis $\Phi_i$ and spectrum $\Lambda_i$ on each $S_i$.
output: A pair of latent shape differences for each shape $i$:
        area-based $\{D_i^A\}_{i=1}^n$ and conformal $\{D_i^C\}_{i=1}^n$.
(1) Compute the CLB $\{Y_i\}_{i=1}^n$ with respect to $S$ and $\mathcal{G}$ via the
    framework of [WHG13].
(2) Compute the canonical CLB $\{\tilde{Y}_i\}_{i=1}^n$ and diagonal matrix with
    the spectrum of the limit shape, $\Lambda_0$, using Algorithm 1.
(3) Construct $D_i^A = \tilde{Y}_i^T \tilde{Y}_i$, and respectively $D_i^C = \Lambda_0^+ \tilde{Y}_i^T \Lambda_i \tilde{Y}_i$.
```

limit shape to the S_i. We exploit this interpretation in Section 5. Moreover, given a new, previously unseen shape S_k with a functional map C_{ik} from S_i to S_k, where S_i is from the collection used to estimate the latent shape, the latent basis can be pushed to S_k simply via $Y_k = C_{ik} Y_i$. We use this property to improve the scalability of latent basis construction.

4.2. Shapes as Characteristic Shape Differences

Although the CCLB used in the construction of the limit shape reduces the instability present in CLB, it still can not directly be used to represent each shape S_i in the collection. The main reason is that each Y_i is expressed in the eigenbasis Φ_i of shape S_i, and therefore, one can not simply compare, for example Y_i with Y_j, which is fundamental in both shape analysis and learning applications.

Instead, we build our shape representation by defining the *characteristic shape differences* (sometimes shortened to *characteristic differences* or abbreviated as *CSD*), which are linear operators acting on the function space of the limit shape, and which, as such, are independent of the basis on each shape.

Let us denote by Λ_0 the spectrum of the limit shape, arising from step 2. of the procedure described in Algorithm 1. Then, following the formulation of [ROA*13], we define the area-based and conformal characteristic shape differences as:

$$D_i^A = Y_i^T Y_i, \tag{3}$$

$$D_i^C = \Lambda_0^+ Y_i^T \Lambda_i Y_i. \tag{4}$$

The final procedure for extracting these operators from a given collection is summarized in Algorithm 2.

Properties of CSD Given a shape collection with an associated FMN, the characteristic shape differences (CSDs) provide a representation of each shape as a pair of matrices of size $k \times k$, where k is the size of the latent basis, independent of the number of points on the shape. In this work, we argue that CSDs inherit some of the most attractive properties of shape differences, such as their compactness and informativeness, while avoiding their shortcomings. Below we summarize the main properties of this representation.

Invariance: CSDs provide a representation that is invariant to rigid (and more generally isometric) shape transformations.

Flexibility: computing CSDs only requires the knowledge of functional maps and places no restriction on the shape discretization. For example, they can accommodate collections of shapes with different number of vertices, or even with different modalities such as point-clouds and meshes.

Informativeness: CSDs fully encode the intrinsic geometry of each shape in the collection in a compact way. Indeed, it follows from Theorem 4.1 that in the presence of full information, given the FMN of a collection of shapes $\mathcal{S}$, the spectrum Λ_0 of the limit shape, and D_i^A, D_i^C for each shape in $\mathcal{S}$, one can recover the intrinsic geometry for each S_i, i.e., the area and stiffness matrices M_i, L_i, which, in turn, fully determines the edge lengths [ZGLG12].

Functoriality: if we interpret each Y_i as the functional map associating the limit shape to S_i, it follows from the functoriality property in [ROA*13] that

$$D_{ij} = Y_i D_i^{-1} D_j Y_i^{-1},$$

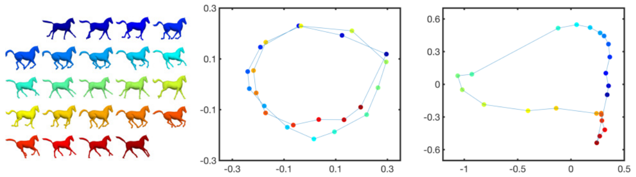
where D_{ij} is the shape difference between S_i and S_j. Thus, CSDs not only encode the difference of each shape to the limit shape but also allow to factor the difference between each pair of shapes.

Algebraic nature: CSDs are linear functional operators on the limit shape. As such, they can be represented as small matrices and manipulated using standard numerical linear algebraic tools, in practice. Moreover, they provide detailed (localized) information about the shape geometry. As we show in Section 6, this allows us to extract *partial* information to compare shape parts, in contrast to purely global shape descriptors.

Independence of base shape and of FMN with particular topology: The original shape difference construction [ROA*13] relies on the choice of a specific base shape, which in turn restricts the input FMN to be of star-shape topology, i.e., it requires to relate the base shape to *each* of the shapes in the collection. The base shape dependence can lead to biased results, especially in the practical scenarios where functional maps are represented in a reduced basis and are not perfectly consistent. On the other hand, as observed in previous works [HWG14], functional maps between similar shapes are typically much easier to compute. Meanwhile, establishing functional maps from a fixed base shape to *all* other shapes in the collection can lead to significant errors.

In contrast, our formulation overcomes both of the limitations above – we can construct the limit shape from an FMN of arbitrary topology, as long as it is connected.

To illustrate this, we consider a sequence of 23 frames of galloping horses shown in Figure 3(a), and assume that only functional maps between consecutive frames are given, resulting in a sparse FMN with chain topology. Figure 3(b) demonstrates that, even when extracted from the sparse FMN, the CSDs recover the cyclical structure of the collection, while using the shape differences from the base shape, S_{12}, computed by composing the given functional maps, leads to an erroneous embedding, as shown in Figure 3(c). Furthermore, We provide another example of the original shape differences leading to biased result in Appendix C.



(a) input shape collection

(b) PCA on latent shape differences

(c) PCA on shape differences to a fixed base shape

Figure 3: *(a) 23 frames of a galloping horse. With a given FMN of chain topology, we computed the characteristic and original original shape differences as signatures of the shapes: (b) PCA layout of the latent shape differences; (c) PCA layout of the original shape differences.*

5. Limit Shapes for Matching

As mentioned in the previous section, a key defining characteristic of our limit shape construction is that it equips the latent shape with metric and measure, which can thus be treated as a 'real' shape from the point-view of spectral techniques. This allows for example to match limit shapes to existing shapes or even match across different limit shapes. We demonstrate the utility of this construction in the following problem: given two collections of heterogeneous shapes $\mathcal{S}^A, \mathcal{S}^B$, where the maps within collection are known, our goal is to *merge* $\mathcal{S}^A$ and $\mathcal{S}^B$, i.e., to find correspondences for *all* cross-collection shape pairs.

This problem of cross-collection merging has been considered in previous works, including [WHG13, HWG14] using constructions based on the latent basis. The key difficulty is that since the latent shapes arising from the original latent basis construction lack geometric structure and cannot be matched directly, previous approaches relied on estimating maps between shapes across subcollections, and using those maps to build connections between the latent shapes via cycle consistency. This can be both error-prone when matching shapes with significant variability, and moreover requires estimating multiple correspondences between real shapes to compute a reliable map between latent shapes (e.g. [HWG14] used 60 shape pairs to merge two shape collections of size 30, while building a hierarchical latent space representation).

Instead, our limit shape construction allows to address this problem by using the metric information of the latent shapes to compute directly a *single* map across limit shapes. Since maps from the limit shapes are known within each collection, this map then provides a direct way to compute cross-collection correspondences between any pair of real shapes. To apply this idea, we pick a real shape, say $S_i \in \mathcal{S}^A$, and then simply replace the standard eigenbasis on S_i, (Φ_i, Λ_i), by $(\Phi_i Y_i, \Lambda_0)$, where Y_i, Λ_0 are computed with Algorithm 1. Effectively we transfer the spectral information from the limit shape to S_i. It is worth noting that since the maps within the collection are known and are consistent, the eigenbasis of the limit shape $\Phi_0 = \Phi_i Y_i$ is stable with respect to different choice of S_i. In the end, we pick two arbitrary shapes from each collection, transfer the spectral information of the respective limit shapes to them, and finally match them directly using existing spectral matching techniques.

Once the functional maps between two limit shapes are estimated, given any two shapes $S_{i'} \in \mathcal{S}^A, S_{j'} \in \mathcal{S}^B$, again we can transfer the eigenbasis and eigenvalues of the limit shapes to them respectively. Since the functional maps between the limit shapes are invariant to the choice of $S_{i'}, S_{j'}$, we can directly recover point-wise maps between the real shape pair using the single pair of estimated functional maps.

6. Shape Collection Comparison

Several approaches have been proposed for detecting geometric variability that exists within a given collection of shapes connected by functional maps, e.g., [ROA*13, HO17]. In this section, we show how our characteristic shape differences can be used for detecting and analyzing and distinciveness *across* different shape collections, or two subsets of a larger collection. Namely, given a set of shapes $\mathcal{S}$, an FMN $\mathcal{G}$ and a partition $\mathcal{S} = \mathcal{S}^A \bigcup \mathcal{S}^B$, we aim to capture the difference between $\mathcal{S}^A$ and $\mathcal{S}^B$, while not being sensitive to the global variability that exists within $\mathcal{S}$. This problem arises especially when trying to detect the detailed geometric properties that are responsible for the differences between shape classes (e.g., healthy vs unhealthy organs), while factoring out the "normal" or "common" variability within the collection.

To give a quantitative measure of variability that manifests within or across shape collections, we introduce first the projected characteristic differences as follows.

Projected Characteristic Shape Differences Let $\{D_i\}_{i=1}^n$ be the CSDs computed using Algorithm 2 with respect to each $S_i \in \mathcal{S}$. Now let us be given a set of functions $F = [\alpha_1, \alpha_2, \cdots, \alpha_k]$, where the $\alpha_i's$ are k orthonormal basis functions on the latent shape, i.e., $F^T F = I_k$. We construct a *projected characteristic shape difference* using F and D_i as follows:

$$P_i(F) = D_i(I - FF^T) + FF^T. \tag{5}$$

It is easy to verify that $P_i(F)\alpha = D_i\alpha$ if α is orthogonal to the subspace spanned by functions in F, and $P_i(F)\alpha = \alpha$ if α is spanned by the functions in F. Intuitively, if F contains the full basis on the latent shape, then $P_i(F) = I$, which forces the CSD to correspond to an area-preserving or conformal map, depending on the type of $D_i's$. We refer readers to Appendix B for an illustration of the projected shape difference.

Global Variability With the projected CSDs, we first formulate an energy functional for measuring how a given function is relevant to the global variability within a shape collection. Our observation is that, suppressing global variability should lead to projected CSDs that are indistinguishable from each other. Namely, we would like to find a basis F such that the latent shape differences projected onto F are as close as possible. For this, we introduce a term that measures the difference of the norms between the original and projected latent shape differences:

$$\delta(S_i, S_j, F) = \|D_i - D_j\|_{\text{Fro}}^2 - \|P_i(F) - P_j(F)\|_{\text{Fro}}^2 \tag{6}$$

According to the following lemma, the change is always nonnegative and can be written as a quadratic form.

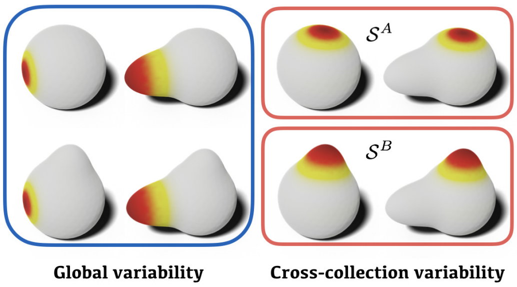


Figure 4: *The global variability of four deformed spheres (in the blue box) and the cross-collection variability regarding the partition $\mathcal{S}^A$ and $\mathcal{S}^B$ (in the red boxes) detected by our algorithms. Note that the horizontal bump (global variability) is of twice the magnitude of the vertical one (cross-collection variability).*

Lemma 6.1 *If $F^T F = Id$, then*

$$\delta(S_i, S_j, F) = Trace(F^T (D_i - D_j)^2 F) \geq 0.$$

It is natural to optimize for a function α^*_{global}, i.e., F consists of a single function, which maximizes the global change of norms within the collection, i.e.,

$$\alpha^*_{\text{global}} = \arg\max_{\alpha} \sum_{i,j} \delta(S_i, S_j, \alpha), s.t. \alpha^T \alpha = 1. \quad (7)$$

In other words, after suppressing the functional deformation related to α^*, the shapes are *maximally* brought together. According to Lemma 6.1, α^* is given by the eigenfunction associated with the largest eigenvalues of $\sum_{i,j}(D_i - D_j)^2$.

Cross-collection Variability Following the same idea above, we formulate the cross-collection variability to be such that after suppressing it, the clusters $\mathcal{S}^A$ and $\mathcal{S}^B$ should become closer to each other, while maintaining their inner structure. In other words, we aim to simultaneously maximize the changes of distances across shapes in different clusters, and minimize those within the same cluster.

Putting these two goals together, we construct:

$$\alpha^*_{\text{cross-collection}} = \arg\max_{\alpha^T\alpha=1} \sum_{(i,j) \text{ across clusters}} \delta(S_i, S_j, \alpha) - \sum_{(i,j) \text{ within same cluster}} \delta(S_i, S_j, \alpha). \quad (8)$$

As an illustration, in Figure 4, we demonstrate the optimizers $\alpha^*_{\text{global}}, \alpha^*_{\text{cross-collection}}$ respectively. Since the horizontal bump is of twice the size of the vertical one, to maximally reduce the global variability, one should suppress the former deformation. Meanwhile, it is intuitive that cluster A and B are distinguished by magnitudes of the vertical bumps, which should be detected as cross-collection variability.

7. Main Experimental Results

7.1. Merging Shape Collections

In this section, we demonstrate the utility of limit shape formulation in merging shape collections. In particular, we compare the quality of the results when merging the collection using maps between either real or latent shapes.

We assume that two input collections, $\mathcal{S}^A, \mathcal{S}^B$, where the maps within the same collection are given, and our goal is to map the pairs across collections by using a single seed cross-collection pair. Namely, given a seed pair $S^A_i \in \mathcal{S}^A, S^B_j \in \mathcal{S}^B$, we can either match S^A_i, S^B_j directly or match the respective limit shapes by transferring the eigenbasis and eigenvalues to S^A_i, S^B_j, as described in Section 5. In both cases and all the experiments, we use the standard functional map estimation pipeline [OCB*17] and refine the initial maps with a recent spectral method [MRR*19]. A small number of landmarks (3 for the first two examples, and 4 for the last one) are added to help the initialization.

Once the limit shapes are matched, we can efficiently compute the maps between any pair of real shapes across collections, without extra estimation, as mentioned in Section 5. On the other hand, in order to merge the collections with the estimated maps between the real shapes, we follow the procedure proposed in [WHG13] – note that the two FMNs within the collections are connected by the computed functional maps between the seed pair, thus we can compute the CLB with the connected FMN on $\mathcal{S}^A \bigcup \mathcal{S}^B$, and use it to estimate the functional maps between all the other pairs.

We also compare our framework to the recent work of Ganapathi-Subramanian et al. [GSDG18], in which the authors construct a modular latent space for transporting functions across shapes (thus, indirectly putting shapes in correspondence), given a set of corresponding descriptors on each shape. Since their work constructs a single modular latent space for the shapes of interest, for this baseline we estimate all cross-collection maps without a merging step. For a fair comparison, we apply the same refinement on the results.

In Figure 5, we merge a set of cats and lions. In order to show the influence of the seed pair selection, we run the above merging procedures independently five times: each time we select the pair in one of the columns in Figure 5(a). The plots in Figure 5(b) demonstrate the accumulated geodesic errors, which shows the fraction (y-axis) of points that are mapped within a given geodesic error (x-axis) to the ground-truth. with respect to matching the real shapes and the limit shapes over the five runs, and that of directly matching shapes across collections with [GSDG18]. Figure 5(c) shows further the error distribution of the merging process with respect to 5 different seed pairs. Here, each column corresponds to a particular choice of seed pair and the points in the column show the accuracy of all 25 cross-collection maps after computing the map between the seed pair. The plots in Figure 5(b) suggest that matching limit shapes is on average easier than matching real shapes and as a result the cross-collection maps are more accurate and, as shown in Figure 5(c) more stable, across different choices of seed pairs. On the other hand, since the main goal of [GSDG18] is to transfer smooth (diffused) descriptors across shapes, it can only produce suboptimal results in estimating point-wise maps.

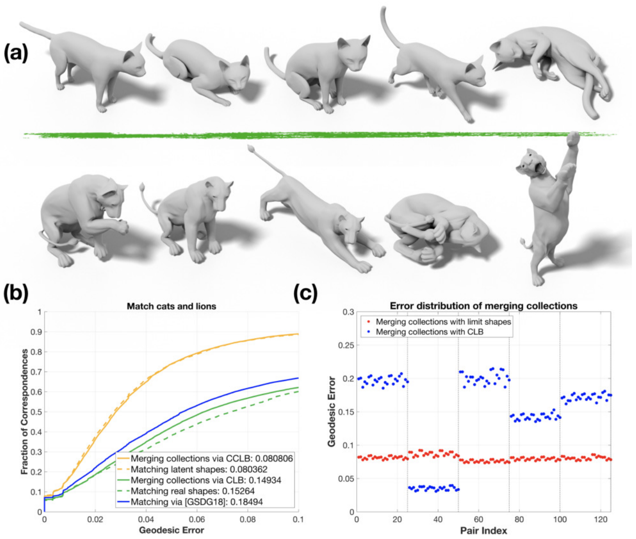


Figure 5: *(a) Heterogeneous shape collections, each pair of shapes in the same column are matched as initialization; (b) The quantitative results of matching the selected pairs and of merging collections via the single pairs using our approach and CLB, and of merging collections via [GSDG18]; (c) Error distributions of merging pairs across collections.*

Another similar example is to merge a set of men and women. Similar trend takes place in this case, especially, as shown in Figure 6(c), when the initialized maps are less accurate, the variance of error with respect to merging collections with CLB is significantly higher than that of our approach. A final example is to merge a set of camels and horses, as shown in Figure 7. Note that for some specific seed real shape pairs, merging the collections can give a satisfactory result. However, these are rare and unknown a-priori, while merging using other real shape pairs can lead to very significant errors. On the other hand, our approach is stable and produces accurate results to a large extent regardless of the shape pair used. In these two cases, the modular latent space approach also behaves similarly to that in Figure 5.

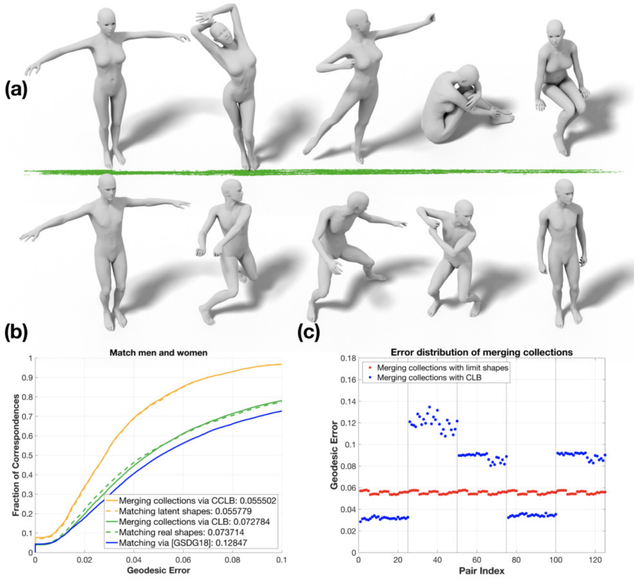


Figure 6: *(a) Heterogeneous shape collections, each pair of shapes in the same column are matched as initialization; (b) The quantitative results of matching the selected pairs and of merging collections via the single pairs using our approach and CLB, and of merging collections via [GSDG18]; (c) Error distributions of merging pairs across collections.*

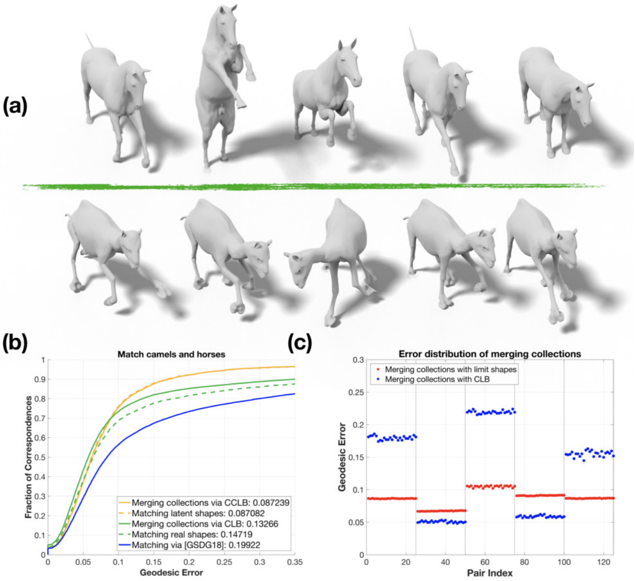


Figure 7: *(a) Heterogeneous shape collections, each pair of shapes in the same column are matched as initialization; (b) The quantitative results of matching the selected pairs and of merging collections via the single pairs using our approach and CLB, and of merging collections via [GSDG18]; (c) Error distributions of merging pairs across collections.*

Timing The experiments are conducted in a machine with 3.3 GHz CPU and 32 GB memory. Across all three sets of experiments, it takes on average 11.0 seconds to estimate functional maps between each seed pair in both the real shape and the limit shape setting. For propagating the maps between the seed pair to whole collection, another 3.2 seconds is consumed for estimating the functional maps between each pair of real shapes using CLB, while our approach is free of estimating any functional maps in this stage. Finally, for [GSDG18], estimating a pair of maps takes 11.2 seconds on average.

7.2. Geometric Exploration of Shape Collections

In the following experiments we demonstrate the utility of our method for capturing cross-collection variability in shape collections, as suggested in Section 6. In particular, we demonstrate that our method can be applied to real-world data (Figure 10), and also demonstrate that our method can extract informative signals in a semi-supervised classification task (Figure 9). We stress that in all applications, we have used *computed* functional maps between shapes that are in most cases not in 1-1 correspondence, estimated using the state-of-the-art functional map estimation framework [RPWO18], unless stated otherwise.

Throughout the results below we use the area-based CSDs, which are extracted from input functional maps of size 60 × 60. To construct the FMN, we first compute distances among shapes (using the shape-DNA descriptors [RWP06]), and form a minimum spanning tree network using these distances. When considering two clusters – we first form a spanning tree on each, and connect shapes across clusters using nearest neighbour search. The methods described in Section 6 optimize for functions α^* on the limit shape, which we map to functions on the actual shapes in the collection, resulting in a consistent and informative visualization.

Since the energy functional defined in Eq. (8) is maximized at the eigenvector associated with the largest eigenvalue of a matrix related to the CSDs, we can further compute the eigenvectors of the leading eigenvalues and obtain a set of functions on the limit shape, each of which indicates some distinctive region. For compact visualizations, we weight the α with the corresponding eigenvalues, and plot the squared-value of resulting weighted sum on the shapes. In all of our results, the color ranges from grey (zero) to red (large) values, and the regions with high values are highlighted.

Verification of Results To start with, we provide both qualitative and quantitative evaluations of our method in data where the distinctive regions are controlled by certain a priori information. In particular, we compare our method to the distinctive region (DR) detection technique presented in [SF07]. In this method, given a set of shape clusters, and an individual shape, the goal is to detect the regions that provide evidence or "justify" this shape as belonging to its cluster.

We generate the data using the PCA model proposed in [HSS*09], which provides shape deformations controlled by semantic parameters such as girth of hip, waist size via PCA-based model. To generate the data, we first sample 4 shapes of different characters and different poses, and then deforme them uniformly by: (1) enlarging the girth of hips by 10%; (2) shrinking the girth of the calves by 20% on top of (1); (3) enlarging the girth of the hips by 30%. As shown in Figure 8, the man in the rest pose is deformed respectively to the three shapes shown in the column marked as GT. The other shapes are deformed in the same way, giving rise to 3 deformations of the four shapes. We treat each deformation as a separate cluster and compare it to the cluster containing the original, undeformed shapes, which we call cluster zero.

Our goal then is to compare each deformation cluster with cluster zero and highlight the distinctive regions that give rise to that particular cluster. In order to obtain a quantitative evaluation of the highlighted area, since the shapes are in one-to-one correspondence, we compute the absolute changes of edge length (metric) between every pair of original and deformed shapes, and then accumulate this per-edge function to a per-vertex function. The resulting "ground truth" functions and the respective deformed shapes are shown in column marked GT of Figure 8.

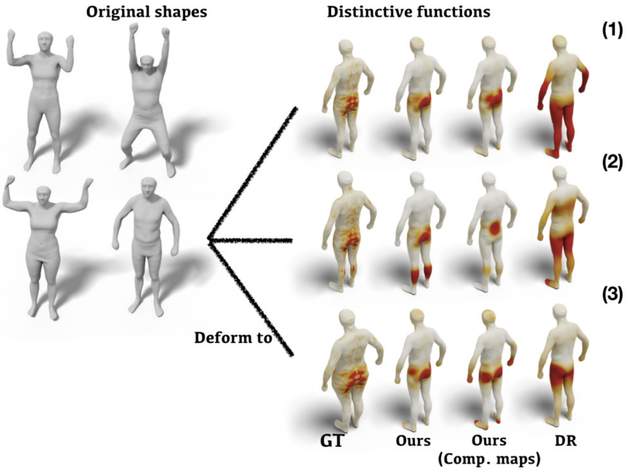


Figure 8: *Left panel: Sampled shapes to be deformed; Right panel: from left to right column: functions representing the ground-truth deformed area regarding the three test cases; the highlighted functions from our method; the highlighted functions from our method with computed maps; the highlighted functions from [SF07].*

We show the regions detected by our approach and that by [SF07] (DR) in Figure 8. From these qualitative results, it is evident that our results are cleaner and more coherent to the ground-truth in all 3 test sets, while the results from [SF07] are more diffused. In addition, we also measure the correlation between the highlighted functions and the ground-truth, i.e., the normalized inner product between the ground truth and the highlighted functions. As shown in Table 1, our results are significantly higher than those of [SF07] across all test sets. Moreover, we test our algorithm with computed functional maps and report both qualitative and quantitative results, which both achieve similar results to the ones from ground-truth maps.

methods \datasets	Set(1)	Set(2)	Set(3)
DR,scale=1	0.3637	0.3678	0.4254
DR,scale=2	0.3643	0.3663	0.4251
DR,scale=3	0.3645	0.3691	0.4141
DR,scale=4	0.3657	0.3691	0.4157
Ours	**0.6308**	**0.5340**	**0.6219**
Ours (computed maps)	0.6305	0.5224	0.5204

Table 1: *Average correlation between the highlighted functions and the ground-truth on each of the shapes of interest in the respective test cases (DR is a multi-scale approach, thus we report the results regarding all scales used in the pipeline).*

Heterogeneous Shape Collection Comparison In Figure 1, we show the computed distinctive functions highlighting the difference between a set of cats (each consists of 7207 vertices) and a set of lions (each consists of 5000 vertices), where the cross-collection maps were estimated using the original functional maps approach [OBCS*12] given a sparse set of landmarks. Note that our method correctly highlights the meaningful distinctive regions according to different clustering criteria: when the animals are grouped according to their species, the highlighted function shown on the left-most row captures the snouts, the four paws and the tails, distinctive to each class, despite the presence of the global poses variability in the collection, while when the animals are grouped by the poses, the highlighted functions on the top row detect the hips and the abdomens.

Clustering with Visual Evidence We analyze two clusters of shapes displayed in the two rows of Figure 9(a) that represent different characters in two distinct poses. It is worth noting that the shapes are from a remeshed version of the FAUST dataset, which do *not* share the same triangulation, and that we estimate maps using the state-of-the-art method [RPWO18]. In particular, we plot the first 3 significant distinctive functions in the left-most 6 columns, which successively capture the bending knees and the shoulders, intuitively distinguishing the two poses. The final weighted functions are plotted from column 7 to 9. Interestingly, we also observe that our formulation can be extended to detecting extrinsic cross-collection variability, by defining a CSD using an inner product capturing extrinsic informations (see details in Appendix C). The corresponding distinctive functions are plotted in the right-most column of Figure 9(a), where the arms and right legs are highlighted, due to the obvious changes in embeddings. Finally, it is worth noting that our method allows errors in map estimation, we refer readers to Figure 15 for an example of inaccurate estimated maps.

We argue that the distinctive function helps to encode the shapes in a way that the clusters are easier to separate. In Figure 9(b, c), we respectively show the PCA plots of shapes represented by $\{D_i\}$ and by $\{X_i\}$, where $X_i = D_i\alpha^*$ and α^* is the most significant distinctive function. Note that the clusters are more pronounced in (c) than in (b), suggesting that α^* indicates the direction to which the underlying variability is more prominent.

To quantify the separability of the representations with respect to the given clustering, we run K-means algorithm for binary clustering on the signals (e.g., the D_i or X_i above) independently 100 times, and compute the average error of separating the shape collection with respect to the given clusters. Intuitively, the lower the error is, the better the two clusters are separated.

We note that, once the CSDs are constructed, the distinctive functions can be optimized for even with a partial set of labeled data, which leads to a semi-supervised learning problem. In particular, we sample uniformly k shapes (k ranges from 1 to 10) from each of the clusters as the labeled data, compute the most distinctive function α_k^* with respect to the $2k$ shapes, and run the quantitative evaluation above on $D_i\alpha_k^*$ to compute the average error rate of clustering the *whole* 20 shapes. We report the average error in Figure 9(d), where the green dashed line corresponds to the error rate of randomly bisecting the shape collection (0.420), and the 0 labeled shape point corresponds to clustering with D_i. This plot indicates that the distinctive function is inherently relevant to the cross-cluster difference – with more and more labeled data, the clustering error with respect to $X_i's$ decreases. Remarkably, when all the labeled shapes are considered, the error rate is 0, which also verifies our observations about the PCA plots above.

Practical Application in Anatomy In this example, we compare two sets of bones of two different populations of wild boars acquired using 3D scanning techniques. In particular, as input we consider 12 bone scans with 24 consistent handcrafted landmarks and 260 sliding landmarks on each of shape [GM13]. We then estimate the FMN using only 6 out of the 24 handcrafted landmarks.

The corresponding shapes and distinctive functions are shown in Figure 10(a). The domain experts verified our result and linked these regions to a behavioural interpretation of the differences between animals that result in specific localized bone modifications.

Note that the clustering information is given by the geographic locations where the bones were collected, and the differences are not visually evident. We run the separability evaluation as in Figure 9(d), and achieve the plot in Figure 10(b). Obviously in this case, the cross-collection variability is more subtle than the one in Figure 9. However, as shown in Figure 10(b), the clustering error decreases with more and more labeled data, suggesting that the relevance of the signals our method captures and the nuance cross-collection variability.

7.3. Learning with CSDs

Our formulation of CSD provides a compact representation for each shape in a collection as a matrix of size $k \times k$, where k is the size of the latent basis. The matrix form of the CSD, in particular, allows us to use a basic learning pipeline to learn properties of shapes from the characteristic shape representation.

In this section, we use the area-based CSDs as input to a neural-network that learns to predict parameters of a generative model of 3D shapes and compare the result to another widely used 3D representation: point-clouds extracted from a 3D surface.

The dataset of interest consists of 400 human shape bodies generated with the open-source implementation [WLS*16] of the SCAPE method [ASK*05]. In particular, each shape is generated with a 12-dimensional vector of parameters, which is our regression target (see details about the data generalization in Appendix D).

In the experiment, we estimate the input functional maps (of size 50×50) within the collection using a recent unsupervised learning approach for shape matching [RO18]. On top of that, we test CSDs that are extracted from FMN of varying topology (e.g., the 10-NN and 20-NN graph with respect to the spectral distances), and truncated at different dimensions (we also report the results on ground-truth maps in Appendix D). Besides, we consider two types of architectures when the input is a CSD: Multi-Linear Perceptrons (MLPs) and Convolutional Neural Networks (CNNs).

For the point-based learning baselines, we construct the input 3D point-clouds by sampling 1024 points uniformly area-wise from the surface of each mesh. In particular, we evaluate three PointNet-like networks [QSMG16] that use encoding/decoding schemes similar to those used in [ADMG18].

Figure 11 shows the MSE between the predicted vectors and the ground-truth for the test shapes in a variety of conditions. And the

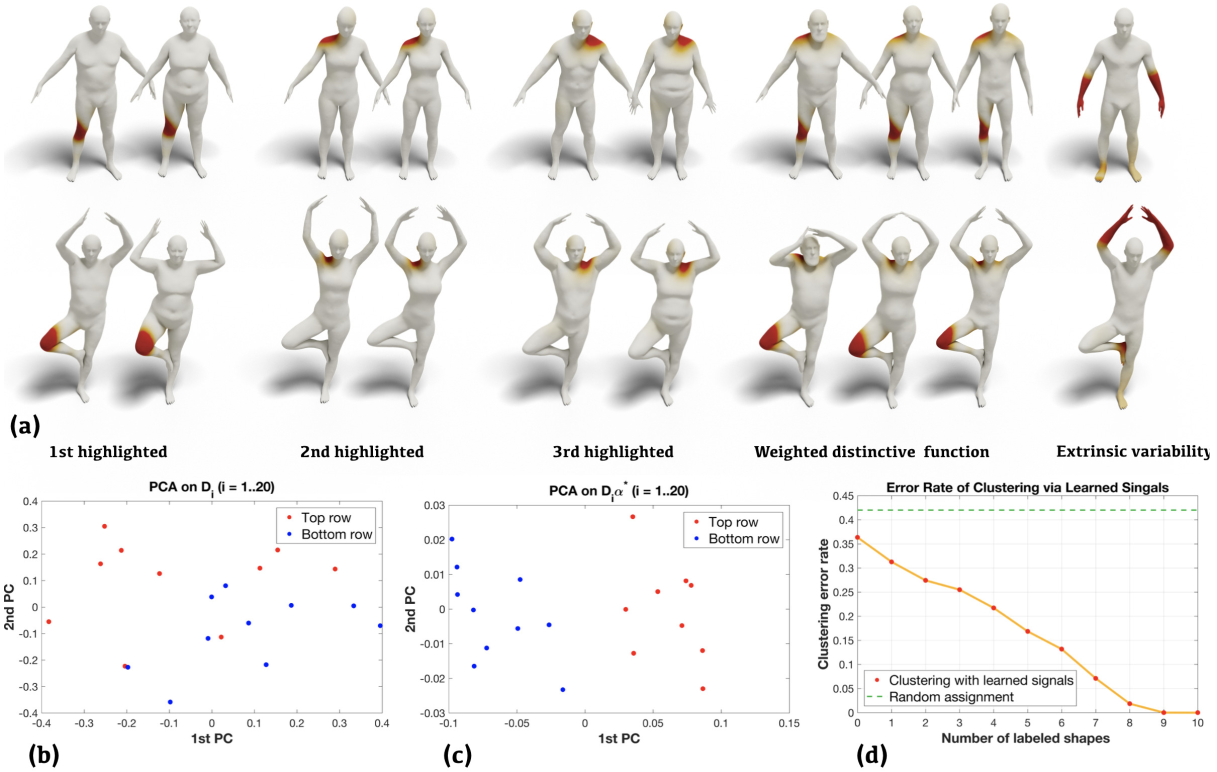


Figure 9: *(a) two sets of humans to be compared, each set (row) corresponding to a distinct pose; (b) PCA on the CSDs, D_i's, where the given clusters are not evident; (c) PCA on $D_i\alpha^{*'}$s, where the given clusters are clear; (d) error rates of clustering via the distinctive signals computed with different number of labeled data in a semi-supervised learning setting. See text for details.*

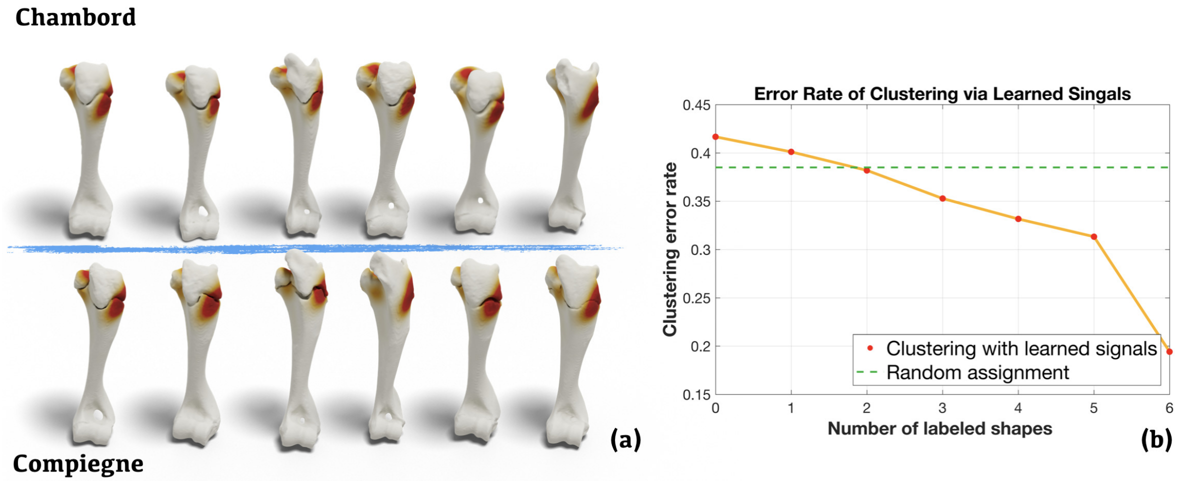


Figure 10: *(a) comparing two sets of bones corresponding to different populations of wild boars using computed functional maps; (b) error rates of clustering via the distinctive signals computed with different number of labeled data in a semi-supervised learning setting.See text for details.*

dashed-line shows the performance of the best over-all point-based architectures. Several interesting trends are revealed by Figure 11: First, CNNs operating with CSDs perform better than MLPs and both perform significantly better than PC-based nets for a wide variety of different configurations. Second, there seems to exist a optimal range between 20 and 30 CSD dimensions, which consistently produces better results across different network topologies. Third, denser topologies give rise to better results. In Figure 17

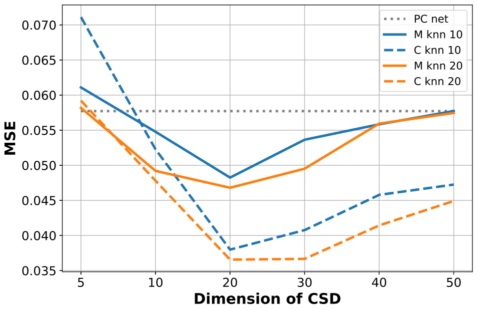


Figure 11: *Regression-based comparison of different input modalities and neural-nets. The y-axis depicts the Mean-Square-Error (MSE) on the* test *split. The x-axis corresponds to the number of CSD dimensions each shape is associated with. The dashed line is the point-based baseline (best of* 15 *models). Models starting with M (solid lines) are MLPs and starting with C (dashed lines) CNNs. knn 10 / knn 20: sparse 10/20-nearest-neighbor FMN topologies. The results are averages of 5 random seeds.*

(Appendix) we show that when using *ground truth* functional maps, similar qualitative behaviors occur e.g. CNNs are consistently better than MLPs, and the lowest MSE is achieved with a CNN operating with CSDs extracted from the densest-possible FMN topology: a clique, with MSE reaching ~ 0.01.

Indeed, CNNs rely and take advantage of the spatial proximities found in regular-grid data such as images. Analogously, according to our formulation for CSD, we have, e.g. in Eq. (3) , $D_i(k,l) = y_k^T y_l$, where y_k is the k^{th} CCLB w.r.t. S_i. Thus, instead of spatial proximity, the neighboring entries in our CSD encode and provide interactions of function pairs that are close in the “spectral” domain.

8. Conclusions

We have presented a novel approach for representing and analyzing 3D shapes in a context of one or multiple collections. Our construction of the limit shape and the characteristic shape differences is based on functional maps network connecting the shapes and a novel analysis that demonstrates that previously used latent functional spaces can both be endowed with a natural geometric structure and provide a basis for representing and comparing shapes in an unbiased way. As a collection summary, the limit shape, though not necessarily emebeddable, is shape-like in many ways and we have demonstrated its use in computing better functional maps in challenging situations. Our characteristic shape differences represent each shape in the collection as a pair of functional operators, stored as small-sized matrices in practice. This representation has many appealing properties, including invariance to rigid motions as well as full intrinsic informativeness that permits reconstruction. We have demonstrated their use in extracting and highlighting variability of interest in a set of shapes, while also suppressing other variability that we regard as nuisance (and which may in fact manifest in larger geometric deformations). We believe that this highly nuanced understanding of shape distortions and variability is important for many applications in engineering, biology, and medicine.

We also note that the matrix form of characteristic shape differences makes it an appealing representation for 3D deep learning applications. Beyond that, the difference matrices also reflect the internal structure of shapes, which renders its potential in building 3D generative model. We plan to explore this in the future work.

Acknowledgement The authors thank the anonymous reviewers for their valuable comments. This work is supported by the KAUST OSR Award No. CRG-2017-3426, a gift from the NVIDIA Corporation, the ERC Starting Grant No. 758800 (EXPROTEA), NSF grants IIS-1528025 and DMS-1546206, a Vannevar Bush Faculty Fellowship, and a gift from the Autodesk Corporation.

References

[ACP03] Allen B., Curless B., Popović Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *ACM transactions on graphics (TOG)* (2003), vol. 22, ACM, pp. 587–594. 2

[ADMG18] Achlioptas P., Diamanti O., Mitliagkas I., Guibas L. J.: Learning representations and generative models for 3d point clouds. *Proceedings of the 35th International Conference on Machine Learning* (2018). 10

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: SCAPE: Shape Completion and Animation of People. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 408–416. 1, 10

[BRLB14] Bogo F., Romero J., Loper M., Black M. J.: FAUST: Dataset and Evaluation for 3d Mesh Registration. In *Proc. CVPR* (2014), pp. 3794–3801. 1

[BV99] Blanz V., Vetter T.: A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194. 2

[BW90] Barr M., Wells C.: *Category theory for computing science*, vol. 49. Prentice Hall New York, 1990. 2

[GM98] Grenander U., Miller M. I.: Computational anatomy: An emerging discipline. *Quarterly of applied mathematics 56*, 4 (1998), 617–694. 1

[GM13] Gunz P., Mitteroecker P.: Semilandmarks: a method for quantifying curves and surfaces. *Hystrix, the Italian Journal of Mammalogy 24*, 1 (2013), 103–109. 10

[GSDG18] Ganapathi-Subramanian V., Diamanti O., Guibas L. J.: Modular latent spaces for shape correspondences. *Computer Graphics Forum 37*, 5 (2018), 199–210. 3, 7, 8

[GSDP*18] Ganapathi-Subramanian V., Diamanti O., Pirk S., Tang C., Niessner M., Guibas L.: Parsing geometry using structure-aware shape templates. In *2018 International Conference on 3D Vision (3DV)* (2018), IEEE, pp. 672–681. 2

[HO17] Huang R., Ovsjanikov M.: Adjoint map representation for shape analysis and matching. In *Proc. SGP* (2017), vol. 36. 2, 6

[HSS*09] Hasler N., Stoll C., Sunkel M., Rosenhahn B., Seidel H.-P.: A Statistical Model of Human Pose and Body Shape. In *Computer Graphics Forum* (2009), vol. 28, pp. 337–346. 1, 9

[HWG14] Huang Q., Wang F., Guibas L.: Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 36. 1, 2, 3, 5, 6

[JDJG04] Joshi S., Davis B., Jomier M., Gerig G.: Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage 23* (2004), S151–S160. 2

[KB14] Kingma D. P., Ba J.: Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014). arXiv:1412.6980. 16

[KBB*13] Kovnatsky A., Bronstein M. M., Bronstein A. M., Glashoff K., Kimmel R.: Coupled quasi-harmonic bases. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 439–448. 2

[Ken89] Kendall D. G.: A survey of the statistical theory of shape. *Statistical Science* (1989), 87–99. 2

[KLM*12] Kim V. G., Li W., Mitra N. J., DiVerdi S., Funkhouser T.: Exploring collections of 3d models using fuzzy correspondences. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 54. 1, 2

[KLM*13] Kim V. G., Li W., Mitra N. J., Chaudhuri S., DiVerdi S., Funkhouser T.: Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 70. 1, 2

[KSKL13] Kurtek S. N., Srivastava A., Klassen E., Laga H.: Landmark-guided elastic shape analysis of spherically-parameterized surfaces. *Comput. Graph. Forum 32* (2013), 429–438. 1

[KvKSHCO15] Kleiman Y., van Kaick O., Sorkine-Hornung O., Cohen-Or D.: Shed: shape edit distance for fine-grained shape similarity. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 235. 1

[MDSB03] Meyer M., Desbrun M., Schröder P., Barr A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, 2003, pp. 35–57. 3

[MRR*19] Melzi S., Ren J., Rodolà E., Wonka P., Ovsjanikov M.: Zoomout: Spectral upsampling for efficient shape correspondence. *CoRR abs/1904.07865* (2019). 7

[OBCCG13] Ovsjanikov M., Ben-Chen M., Chazal F., Guibas L.: Analysis and visualization of maps between shapes. *Computer Graphics Forum 32*, 6 (2013), 135–145. 2

[OBCS*12] Ovsjanikov M., Ben-Chen M., Solomon J., Butscher A., Guibas L.: Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 30. 2, 10

[OCB*17] Ovsjanikov M., Corman E., Bronstein M., Rodolà E., Ben-Chen M., Guibas L., Chazal F., Bronstein A.: Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses* (2017), ACM, p. 5. 2, 3, 7

[OLGM11] Ovsjanikov M., Li W., Guibas L., Mitra N. J.: Exploration of continuous variability in collections of 3d shapes. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 33. 1

[PP93] Pinkall U., Polthier K.: Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics 2*, 1 (1993), 15–36. 3

[QSMG16] Qi C. R., Su H., Mo K., Guibas L. J.: Pointnet: deep learning on point sets for 3d classification and segmentation. *CoRR abs/1612.00593* (2016). 10, 15

[RO18] Roufosse J., Ovsjanikov M.: Unsupervised deep learning for structured shape matching. *CoRR abs/1812.03794* (2018). `arXiv:1812.03794`. 10

[ROA*13] Rustamov R. M., Ovsjanikov M., Azencot O., Ben-Chen M., Chazal F., Guibas L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics 32*, 4 (2013), 1. 1, 2, 3, 5, 6, 14

[Rot08] Rotman J. J.: *An introduction to homological algebra*. Springer Science & Business Media, 2008. 2

[RPWO18] Ren J., Poulenard A., Wonka P., Ovsjanikov M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph. 37*, 6 (Dec. 2018), 248:1–248:16. 9, 10

[RWP06] Reuter M., Wolter F.-E., Peinecke N.: Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Comput. Aided Des. 38*, 4 (Apr. 2006), 342–366. 9

[SF07] Shilane P., Funkhouser T.: Distinctive regions of 3D surfaces. *ACM Transactions on Graphics 26*, 2 (June 2007), Article 7. 9

[SNB*12] Solomon J., Nguyen A., Butscher A., Ben-Chen M., Guibas L.: Soft maps between surfaces. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1617–1626. 2

[T*17] Thompson D. W., et al.: On growth and form. *On growth and form.* (1917). 2

[TZL*12] Tong J., Zhou J., Liu L., Pan Z., Yan H.: Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics 18*, 4 (2012), 643–650. 2

[WAO*09] Wand M., Adams B., Ovsjanikov M., Berner A., Bokeloh M., Jenke P., Guibas L., Seidel H.-P., Schilling A.: Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Transactions on Graphics (TOG) 28*, 2 (2009), 15. 2

[WAvK*12] Wang Y., Asafi S., van Kaick O., Zhang H., Cohen-Or D., Chen B.: Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 165. 1

[WHG13] Wang F., Huang Q., Guibas L. J.: Image co-segmentation via consistent functional maps. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 849–856. 3, 4, 5, 6, 7, 14

[WHOG14] Wang F., Huang Q., Ovsjanikov M., Guibas L. J.: Unsupervised multi-class joint image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 3142–3149. 3

[WJH*07] Wand M., Jenke P., Huang Q.-X., Bokeloh M., Guibas L., Schilling A.: Reconstruction of deforming geometry from time-varying point clouds. In *Proc. SGP* (2007), pp. 49–58. 2

[WLS*16] Wang H., Li Y., Su H., Wang Z., Tu C., Lischinski D., Cohen-Or D., Chen. B.: Synthesizing Training Images for Boosting Human 3d Pose Estimation. In *3D Vision* (2016), IEEE, pp. 479–488. 10, 14

[WS13] Wang L., Singer A.: Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA 2*, 2 (2013), 145–193. 3

[You10] Younes L.: *Shapes and diffeomorphisms*, vol. 171. Springer Science & Business Media, 2010. 3

[ZGLG12] Zeng W., Guo R., Luo F., Gu X.: Discrete heat kernel determines discrete riemannian metric. *Graphical Models 74*, 4 (2012), 121–129. 5

Appendix A: Technical Details

Proof of Theorem 4.1

Proof First note that $\tilde{\Phi}_0$ is well-defined since by consistency, $\Phi_j Y_j = \Phi_j C_{ij} Y_i = \Phi_i Y_i$. The regularization constraint $\sum_i Y_i^T Y_i = Id$ therefore implies $\sum_i (\Phi_i M_i \tilde{\Phi}_0)^T \Phi_i M_i \tilde{\Phi}_0 = \tilde{\Phi}_0^T (\sum_i M_i) \tilde{\Phi}_0 = Id$. Now let $E = \sum_i Y_i^T \Lambda_i Y_i$ to be a diagonal matrix (implicitly corresponds to the eigenvalues of the latent shape). Note that Λ_i is a non-negative diagonal matrix, thus E admits an eigen-decomposition $EU = U\Lambda_0$ and we let $\Phi_0 = \tilde{\Phi}_0 U$. Direct computation yields that $\Phi_0^T (\sum_i M_i) \Phi_0 = Id$, and $\Phi_0^T (\sum_i L_i) \Phi_0 = \Lambda_0$. Thus it follows from $\Phi_0^T M \Phi_0 \Lambda_0 = \Lambda_0 = \Phi_0^T L \Phi_0$ that $L\Phi_0 = M\Phi_0\Lambda_0$. On the other hand, it is easy to verify that the eigenfunctions of (L, M) satisfies the consistency constraint and the normalization, therefore they are equivalent. □

Proof of Lemma 6.1

Proof We first prove that:

$$\|D_i - D_j\|_{\text{Fro}}^2 - \|P_i(F) - P_j(F)\|_{\text{Fro}}^2 =$$
$$\|(D_i - D_j) - (P_i(F) - P_j(F))\|_{\text{Fro}}^2.$$

It is easy to verify that $(FF^T)^2 = (FF^T)$, since $F^T F = Id$. In other words, FF^T is a projection operator, then so is $Id - FF^T$. For the sake of simplicity, we denote in the following $D_i - D_j$ and $Id - FF^T$ by Δ, K respectively. Obviously Δ, K are both symmetric matrices, and $K^2 = K$. Then the above equivalence can be rerewritten as

$$\mathrm{Trace}(\Delta^T\Delta) - \mathrm{Trace}(K^T\Delta^T\Delta K) = \mathrm{Trace}[(\Delta - \Delta K)^T(\Delta - \Delta K)],$$

which amounts to $\mathrm{Trace}(K^T\Delta^T\Delta K) = \mathrm{Trace}(K^T\Delta^T\Delta)$.
Finally, the equivalence follows from

$$\begin{aligned}\mathrm{Trace}(K^T\Delta^T\Delta K) &= \mathrm{Trace}(\Delta^T\Delta K K^T) \\ &= \mathrm{Trace}(\Delta^T\Delta K^2) = \mathrm{Trace}(\Delta^T\Delta K).\end{aligned}$$

Finally, the difference is equal to

$$\|\Delta K\|_{\mathrm{Fro}}^2 = \mathrm{Trace}(K\Delta^2 K) = \mathrm{Trace}(F^T\Delta^2 F(F^T F)) = \mathrm{Trace}(F^T\Delta^2 F)$$

□

Appendix B: Details on CCLB

Consistent Latent Basis Extraction In [WHG13], the authors consider the minimizing the following energy with respect to inconsistency of a given FMN:

$$E(Y) = \sum_{(i,j)\in\mathcal{G}} \|C_{ij}Y_i - Y_j\|_F^2, Y = (Y_1^T, Y_2^T, \cdots, Y_n^T)^T, \quad (9)$$

where C_{ij} are functional maps of size $k \times k$. To avoid degenerate solutions, the authors introduce a regularization term $Y^T Y = I$. Under this constraint, minimizing $E(Y)$ reduces to finding the eigenfunctions of a matrix W, whose $(i, j)-$th block is given as follows:

$$W_{i,j} = \begin{cases} \sum\limits_{(i,j')\in\mathcal{G}} (I + C_{ij'}^T C_{ij'}) & \text{if } i = j, \\ -(C_{ji} + C_{ij}^T) & \text{otherwise.} \end{cases} \quad (10)$$

And we collect $Y = [Y_1^T, Y_2^T \cdots Y_3^T]^T$, the smallest (closest to 0) $k_c \le k$ eigenbasis of W, and let $Y_i \in \mathbb{R}^{k\times k_c}$ be the consistent latent basis on S_i.

Limit Shape of Shapes with Different Triangulations In Section 4, we compute the ground truth limit shape of the collection in Figure 2(a) according to Theorem 4.1. Now we compare it to the limit shape of the remeshed version (each of around 5000 vertices) of those shapes, computed via Algorithm 1. In particular, the maps between the remeshed shapes are obtained by mapping each of them to a common template, which are neither injective nor surjective. We plot the 11^{th} to the 21^{st} eigenfunctions of the ground-truth limit shape, the computed limit shape, and one of the real shape respectively on the top, middle and bottom row in the left panel of Figure 12, and in the right panel we plot the respective spectra. It is evident that, the computed eigenbasis and spectrum of the limit shape approximate to the ground-truth, despite the basis compression and noisy FMN.

Projected Shape Difference We show a simple illustration of the projected shape difference in Figure 13. Given a sphere, we first compute the shape difference, D, on it with respect to a deformed sphere. We pick three function f_1, f_2, f_3 on the sphere and map them to the deformed one using the functional map. We define $d_O(A,B) = \|A - B\|_{\mathrm{Fro}}/\|B\|_{\mathrm{Fro}}$, which in turn indicates how far D and the projected versions w.r.t. f_i from the identity operator. Intuitively, projecting f_1, which is supported by the most deformed region, leads to the most significant shrink in operator distance, while projecting out f_3 leads to little change.

Appendix C: Supplementary Shape Analysis Results

Example of Biased Results of the Original Shape Differences

In this example, we align a collection of cats and dogs shown in Figure 14 *without any maps across them* using the original and the characteristic shape differences. For the former, we assume that a pair of shapes, e.g., the boxed animals in Figure 14, to be used as bases in each cluster, and compute the eigenvalues of the respective shape differences as descriptors. On the other hand, we also use the eigenvalues of the CSDs as the descriptor for each shape in the collection, without any a priori information. Note that, when using the approach of [ROA*13] even after fixing the corresponding base shapes, *none* of the base shape choices lead to the correct result. We demonstrate one such result in the bottom row of Figure 14, obtained by fixing the base shapes to be the ones shown in the blue boxes. In contrast, using the characteristic shape differences results in the ground-truth alignment (middle row). Note that the same experiment has been conducted in [ROA*13] (see Figure 13 therein), however, to obtain the exact alignment, the authors used *all pairwise* shape differences.

Details on the Experiment in Figure 9

First of all, to demonstrate the robustness of our method with respect to noise in map estimations, we show one typical inaccurate computed map in Figure 15, where exists symmetrical mismatched parts. Indeed, 13 out of the 56 maps estimated in this experiment contain visually evident mismatched parts.

For detecting the extrinsic cross-collection variability as shown in the right-most column of Figure 9, we consider the following inner product on shape S defined as follows:

$$E(i,j) = \begin{cases} -E(i,j) & \text{if } i \neq j, \\ \sum_{i\neq j} E(i,j) & \text{i = j.} \end{cases} \quad (11)$$

where $E(i,j) = \|v_i - v_j\|^2 M(i,i)M(j,j)$, in which v_i, v_j are two vertices on S and $M(i,i)$ is the lumped area mass round v_i. The we define a 'pseudo' characteristic shape difference encoding the extrinsic deformation on S as $Y^T\Phi^T E\Phi Y$, where Y is the CCLB with respect to S, and Φ is the eigenbasis on S. We call it a pseudo CSD because the limit shape is not embedded in $\mathbb{R}^3$, thus one can not in general define such inner product on the limit shape. However, as shown in Figure 9, such approach indeed captures the extrinsic deformation, while being insensitive to the intrinsic deformation captured by the area-based CSDs.

Appendix D: Neural Network Details

Data Preparation

In [WLS*16], body variations are controlled with 12 latent parameters $\in [0,1]$, which informally encode shape attributes such

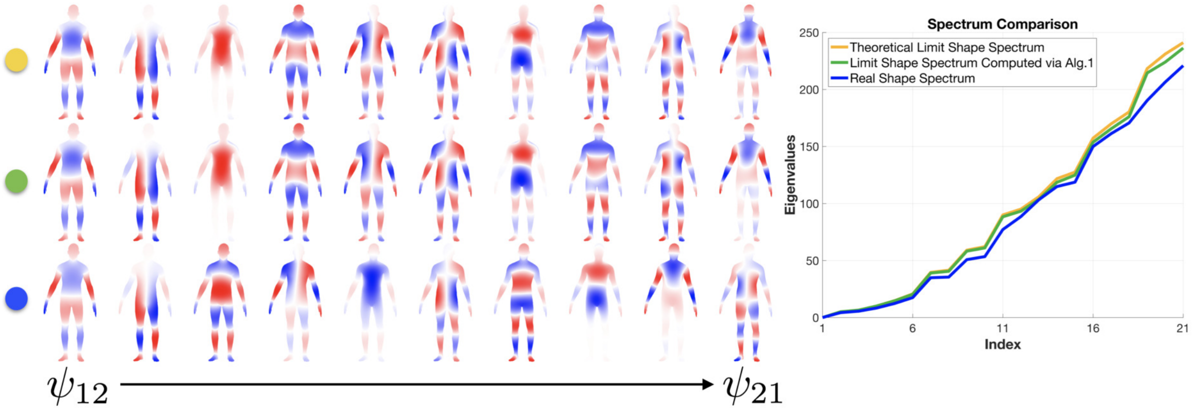


Figure 12: *Left: the 12th to 21st eigenfunctions of the ground-truth limit (top); that of the computed limit shape (middle); that of the real shape (bottom). The color goes from negative values (blue) to positive values (red), the signs of eigenfunctions are synced; Right: the spectra comparison. .*

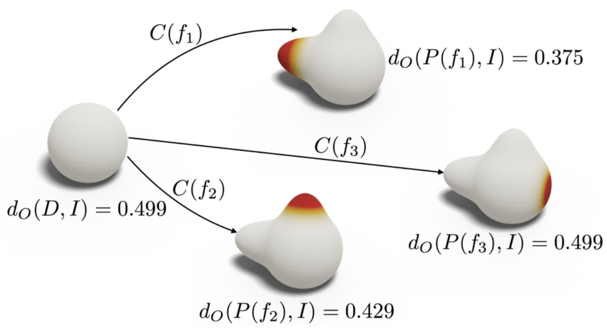


Figure 13: *After projecting out the function f_1, f_2, f_3, the distances from the corresponding projected shape differences to the identity operator I varies (See text for details).*

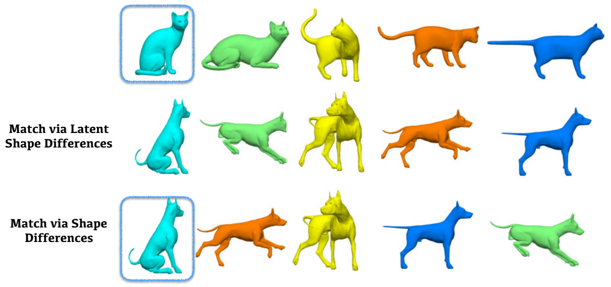

Match via Latent Shape Differences
Match via Shape Differences

Figure 14: *Simultaneous analogies between a collection of cats and dogs without maps across them. The ground-truth correspondences are indicated by the color-coding. In the middle row, the characteristic shape differences recover the ground-truth alignment. On the other hand, the shape differences fail to recover — one failure example using the shape differences with the boxed base shape are shown on the bottom.*

as height, leg-girth, belly protrusion, etc. To generate our meshes we sample uniformly i.i.d. each of the latent parameters and consider eight modifications of the standard T-pose. See Figure 16 for a sample of the resulting meshes.

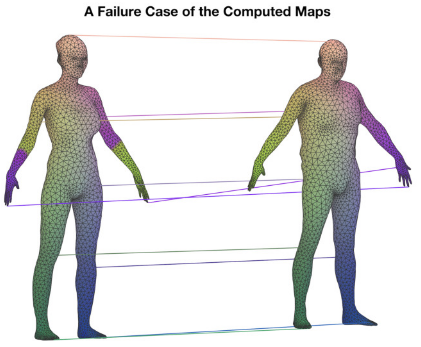


Figure 15: *An example of the inaccurate maps in our map estimation. Note also that the shapes are of different triangulations, and therefore no trivial bijective map is available.*

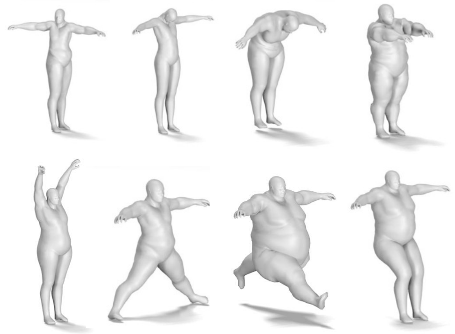


Figure 16: *Example synthetically generated meshes used within the leaning-based pipeline of Section 7.3, displaying a randomly selected mesh of each pose-class.*

Point-Cloud Architectures

We used three configurations for making point-base architectures. In a spirit similar to [QSMG16] we implemented all (3-layer deep) encoders as 1-D convolutions with filter size 1, i.e., treating each point independently. The output of the last encoding layer was further processed by a feature-wise max-pool which was further processed by an FC-ReLU decoder. Table 2 shows the exact number of parameters (columns) in each consecutive layer for the three configurations (rows).

Version	Encoder (# filters)	Decoder (# Neurons)
A	{32, 64, 64}	{64, 12}
B	{64, 128, 128}	{64, 12}
C	{64, 128, 128}	{64, 128, 12}

Table 2: *Size of layers in point-based architectures for the versions that formed the baseline of the regression experiments. The further right a parameter is displayed the deeper the underlying layer of architecture is.*

We trained each of these architectures with learning rates of {0.001, 0.002, 0.005, 0.007, 0.01}. The learning rate of 0.005 gave the best performance in the regression experiments..

MLPs

We used FC-ReLU MLPS for which the last 3 layers had {50, 100, 12} neurons respectively. The number of neurons of the first layer was calibrated according to the size of the input difference matrix. Table 3 shows their correspondence.

# Latent-Bases	5	10	20	30	40	50
# Neurons	369	185	62	29	17	11

Table 3: *Number of neurons in first layer of MLP-architectures based on the size corresponding to the #Latent-Bases.*

CNNs

The encoding part of our CNNs was comprised by two convolutional layers leading to a single FC-ReLU layer with 12 neurons. See Table 4 and Table 5 for the parameters of the convolutional layers when the input was 20 × 20 difference matrices and 40 × 40, respectively.

Layer	# Filters	Kernel-size	Stride
First	10	(2, 2)	1
Second	10	(4, 4)	2

Table 4: *CNN parameters with* 20×20 *input.*

Training Details

Through all the experiments we use a 75%, 15%, 10% train-test-val split of the resulting dataset.

For training we used stochastic gradient descent with Adam [KB14] ($\beta_1 = 0.9$) and batch-size of 50 throughout all experiments. Moreover we normalized the differences matrices by subtracting their average wrt. the training split. For the regression task, the networks operating with CSD were trained with a learning rate of 0.007.

The reported MSE is the average over five random data-splits and weight initializations of the neural nets. The networks are trained maximally for 500 epochs and the displayed MSE correspond to the model (epoch) that optimized the validation split.

Layer	# Filters	Kernel-size	Stride
First	10	(3, 3)	2
Second	10	(4, 4)	2

Table 5: *CNN parameters with* 40×40 *input.*

Additional Results on Regression-based Learning

We present the result of training with the CSDs using the ground-truth maps in Figure 17. See the caption for details

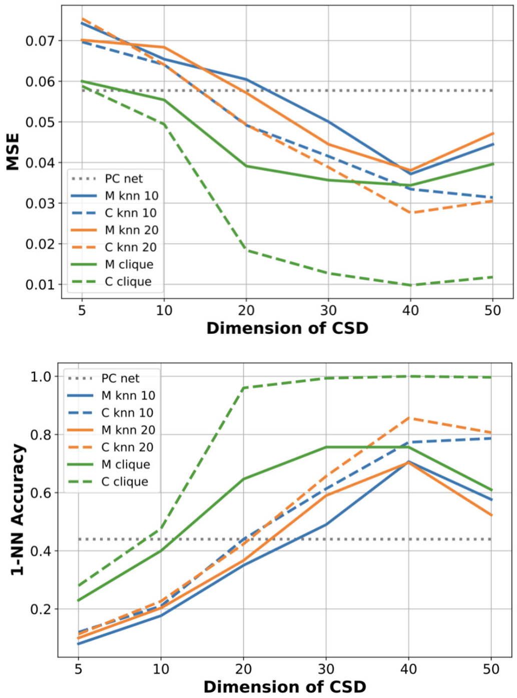


Figure 17: *Regression-based comparison of different input modalities and neural-nets, when using* ***ground-truth maps****. On the top plot the y-axis depicts the Mean-Square-Error (MSE) on the* ***test*** *split. On the bottom plot the y-axis depicts the fraction of* ***test*** *shapes for which their predicted parameter vector was the nearest neighbor to their corresponding ground-truth parameter vector; among* ***all*** *ground-truth parameter vectors of the test split. For both plots: the x-axis corresponds to the number of rows (equally, columns) the underlying CSD of each shape has. The dashed line is the point-based baseline (best of* 15 *such models). Models starting with M (solid lines) are MLPs and starting with C (dashed lines) CNNs. knn 10 / knn 20: sparse 10/20-nearest-neighbor FMN topologies. Clique stands for the clique FMN topology. The results are averages of 5 random seeds.*