Landmark-Based Shape Encoding and Sparse-Dictionary Learning in the Continuous Domain

Daniel Schmitter and Michael Unser

***Abstract*— We provide a generic framework to learn shape dictionaries of landmark-based curves that are defined in the continuous domain. We first present an unbiased alignment method that involves the construction of a mean shape as well as training sets whose elements are subspaces that contain all affine transformations of the training samples. The alignment relies on orthogonal projection operators that have a closed form. We then present algorithms to learn shape dictionaries according to the structure of the data that needs to be encoded: 1) projection-based functional principal-component analysis for homogeneous data and 2) continuous-domain sparse shape encoding to learn dictionaries that contain imbalanced data, outliers, or different types of shape structures. Through parametric spline curves, we provide a detailed and *exact* implementation of our method. We demonstrate that it requires fewer parameters than purely discrete methods and that it is computationally more efficient and accurate. We illustrate the use of our framework for dictionary learning of structures in biomedical images as well as for shape analysis in bioimaging.**

***Index Terms*— Sparse coding, dictionary learning, PCA, sparsity, splines, segmentation.**

I. Introduction

GIVEN a training set $\{\boldsymbol{r}_k\}_{k=1,\dots,K}$ of K parametric curves $\boldsymbol{r}_k(t) \in L_2([0,1],\mathbb{R}^2)$ defined by a set of corresponding landmarks, we aim at learning a dictionary whose atoms best capture the shape variability of the training set. We first define for each curve $\boldsymbol{r}_k$ a subspace $S_k = \{\mathbf{A}\boldsymbol{r}_k + \boldsymbol{b} : \mathbf{A} \in \mathbb{R}^{2\times 2}, \boldsymbol{b} \in \mathbb{R}^2\}$ that contains all admissible affine or similarity transformations of $\boldsymbol{r}_k$. Next, we compute the mean shape $\boldsymbol{r}_{\text{mean}}$ that is closest to all subspaces S_k and project it back onto each S_k (see Figure 1) to obtain an aligned training set $\{\tilde{\boldsymbol{r}}_k = \mathcal{P}_k\boldsymbol{r}\}_{k=1,\dots,K}$, where $\mathcal{P}_k : L_2([0,1],\mathbb{R}^2) \to S_k$ is the orthogonal projection operator that projects a query curve $\boldsymbol{r}$ onto S_k. We use the aligned training data to learn dictionaries by either computing a continuous-domain functional principal-component analysis (fPCA) or for sparse shape encoding, depending on the structure of the data. Our approach allows one to construct dictionaries that contain atoms that are invariant to the specific affine transformation being used. For instance, if the geometric transformation is a similarity transformation, then the resulting fPCA does not depend on the location, size, or orientation of the original curves $\{\boldsymbol{r}_k\}_{k=1,\dots,K}$.

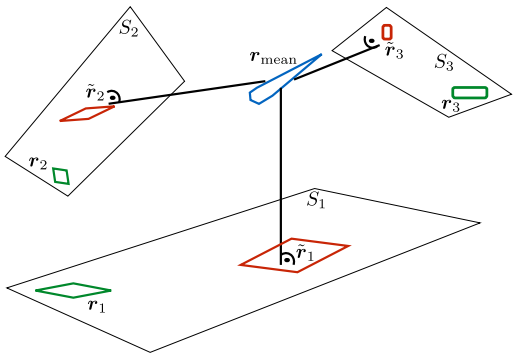

Fig. 1. Unbiased shape alignment of curves. For each curve $\boldsymbol{r}_k$ the vector space S_k is built w.r.t. an admissible geometric transformation. The shape $\boldsymbol{r}_{\text{mean}}$ that is closest to all subspaces S_k is computed and projected back to each subspace, which yields the aligned shapes $\tilde{\boldsymbol{r}}_k$ that define the data used to construct the shape dictionary.

A. Contribution

1) Mean-Shape Construction and Curve Alignment: We provide a method to construct vector spaces that contain all admissible affine transformations of a particular curve. Our model is generic and has the advantage that it allows one to specify which kind of transformation needs to be used, such as similarity, shearing, reflection, scaling, or others. Instead of defining the vector space through its explicit basis, we implicitly define it by characterizing the orthogonal projector onto the vector space. These projectors allow us to compute a mean shape, which we use to align a training set by "removing" from the data the affine transformation used to construct the vector space. The specificity of our alignment method is that it does not depend on the particular choice of a reference shape or template. We also provide a closed-form solution instead of an iterative method.

2) Dictionary Learning With Projection-Based Functional PCA: We show how to compute an fPCA for parametric curves with the aligned training set. The principal components are used as atoms to construct the learned dictionary.

3) Exact Implementation Using Spline Curves: We provide formulas for the exact implementation of our continuous-domain framework using splines. We derive the equivalent spline-based representation of the projectors and fPCA and

Manuscript received June 3, 2017; revised September 19, 2017; accepted October 6, 2017. Date of publication October 12, 2017; date of current version November 3, 2017. The work was supported by the Swiss National Science Foundation under Grant 200020-162343/1 The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tolga Tasdizen. *(Corresponding author: Daniel Schmitter.)*

The authors with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: daniel.schmitter@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2762582

show how our model is implemented at no additional cost compared to a purely discrete approach. Yet, we benefit from the fact that spline curves need fewer parameters than common landmark-based methods to accurately describe a shape.

4) Sparse Shape Encoding: We present a method that enforces sparsity to learn dictionaries that can be applied to training data unsuitable to be analyzed with L_2 methods. We provide formulas to express the continuous-domain L_2 norm of any spline curve as a discrete ℓ_2 norm. We show how to exploit these formulas to convert the continuous domain L_2-ℓ_1 sparse coding problem into a discrete ℓ_2-ℓ_1 optimization problem; this step is crucial for sparse shape encoding.

II. Related Work

A. Sparsity-Based Learning Methods

Sparse signal representation models that typically involve the minimum of an ℓ_1-norm provide more flexibility than ℓ_2-based methods to encode training data because 1) unlike methods related to principal component analysis (PCA), they do not enforce orthogonality on the basis vectors and 2) they are less sensitive to outliers or inhomogeneous data [1], [2]. Methods to learn sparse dictionaries, such as sparse PCA [3]–[5] have been proposed for image denoising [6] or to solve image-classification tasks [1]. In the context of shape analysis, sparse learning methods have been applied to medical imaging [7], [8]. However, since these algorithms are formulated in the discrete domain, they are penalized by the trade-off required to behave accurately and the number of shape descriptors (such as landmarks).

B. Statistical Shape Models

The ℓ_2-based learning methods to characterize shape data and capture its variability can be traced back to the classical Point Distribution Model (PDM), which is the basis of the Active Shape Model (ASM) [9], [10]. Landmark-based curves are aligned by minimizing the variance of the distance between corresponding points. Originally, the ASM was introduced to segment images. Its main difference with active contour models [11]–[13] is that it enforces deformations that are consistent with the training set. The ASM and related statistical shape models [14] usually require that the training set be aligned or registered to a common reference prior to the statistical analysis. Iterative methods, such as the popular Procrustes Analysis [15] are used to compute a mean shape from a properly aligned set of training data. A PCA is then applied to the renormalized training data to compute the modes that describe the variation within the data. Although different alignment strategies exist, it remains a challenge to reduce the bias that is introduced when computing the mean shape [16]. Moreover, these algorithms are iterative, which can be inconvenient if fast online methods are required. Furthermore, they do not allow for a flexible choice of the particular geometric transformation (*e.g.*, rigid-body, similarity, scaling) that is removed upon re-normalization. This restricts their application to a specific class of shapes.

The methods mentioned above are considered as discrete methods. Attempts to construct statistical shape models in the continuous domain have been proposed by making use of B-splines [17]; however, they do not fully exploit the L_2 Hilbert-space structure of parametric spline shapes.

Statistical shape models are closely related to shape analysis [18] or segmentation models because they are often used to incorporate prior information about shapes into an algorithm [19]–[22]. In this context, spline-based curve representations are convenient because they enable to implement smooth shapes in the continuous domain [23]–[26] with only few parameters.

III. Curve Projectors

Given a training set $\{\boldsymbol{r}_k\}_{k=1,\dots,K}$ of curves, it is necessary to first align the shapes in order to construct a dictionary. This step corresponds to the centering of the data vectors in a classical PCA. To guarantee an unbiased alignment, we propose to associate to each sample curve $\boldsymbol{r}_k$ a subspace that contains all admissible affine transformations of $\boldsymbol{r}_k$. Then, we compute the curve $\boldsymbol{r}_{\text{mean}}$ that is the closest to all subspaces and project it back to them to obtain the aligned curves $\{\tilde{\boldsymbol{r}}_k\}_{k=1,\dots,K}$ (see Figure 1). In the following, we first describe the theory to formulate affine spaces of curves and projection operators.

A. The Hilbert Space $\mathcal{H}$ Containing All Parametric Curves

We describe a 2D parametric curve as $\boldsymbol{r}(t) = (r_x(t), r_y(t))$, where $t \in [0, 1]$. The normalization of the parameter domain to $[0, 1]$ can always be done without loss of generality. We denote by $\mathcal{H} : L_2([0, 1], \mathbb{R}^2)$ the Hilbert space associated with the standard L_2-inner product $\langle \boldsymbol{r}_k, \boldsymbol{r}_l \rangle := \int_0^1 \boldsymbol{r}_k^{\mathsf{T}}(t)\boldsymbol{r}_l(t)\mathrm{d}t$ that contains all 2D parametric curves. The corresponding norm is defined as $\|\boldsymbol{r}\|_{L_2} := \sqrt{\langle \boldsymbol{r}, \boldsymbol{r} \rangle}$.

B. Shape Subspaces of $\mathcal{H}$

We define a subspace as the space that contains all admissible geometric transformations of a reference curve $\boldsymbol{r}^{\text{ref}}$. Such a subspace can be defined as a finite-dimensional vector space $\mathcal{S}^{\text{ref}}$ of dimension I, whose basis $\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,I}$ consists of elements $\boldsymbol{e}_i^{\text{ref}}$, which themselves are curves that depend on $\boldsymbol{r}^{\text{ref}}$. Hence, every element (*i.e.*, curve) living in $\mathcal{S}^{\text{ref}}$ can be expressed as a linear combination of the basis elements. Thus,

$$S^{\text{ref}} = \left\{ \sum_{i=1}^{I} u_i \boldsymbol{e}_i^{\text{ref}}(\cdot) : u_i \in \mathbb{R} \right\} \tag{1}$$

is a subspace of the Hilbert space $\mathcal{H}$. We now illustrate this concept with the following example.

1) Example - Affine Vector Space: The affine transformation of a 2D curve $\boldsymbol{r}$ can be expressed as $\mathbf{A}\boldsymbol{r} + \boldsymbol{b}$, where $\mathbf{A} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ is a (2×2) matrix with elements $a_i \in \mathbb{R}$, $i = [1 \dots 4]$ and $\boldsymbol{b} = (b_1, b_2) \in \mathbb{R}^2$ is a translation vector.

TABLE I
BASES OF VECTOR SPACES

Transformation	Basis $\{\boldsymbol{e}_i^{\text{ref}}\}$ w.r.t. $\boldsymbol{r}^{\text{ref}}$
uniform scaling	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
non-uniform scaling	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
translation	$\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
similarity	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} -r_y^{\text{ref}} \\ r_x^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
affine	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} r_y^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ r_x^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 0 \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
shear + uniform scaling	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} r_y^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$
reflection + similarity	$\left\{\begin{pmatrix} r_x^{\text{ref}} \\ -r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} r_y^{\text{ref}} \\ r_x^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}$

By evaluating the matrix-vector product explicitly, we obtain

$$\mathbf{A}\boldsymbol{r}(t) + \boldsymbol{b} = a_1 \begin{pmatrix} r_x(t) \\ 0 \end{pmatrix} + a_2 \begin{pmatrix} r_y(t) \\ 0 \end{pmatrix} + a_3 \begin{pmatrix} 0 \\ r_x(t) \end{pmatrix} + a_4 \begin{pmatrix} 0 \\ r_y(t) \end{pmatrix} + b_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Therefore, the affine space associated to the 2D reference curve $\boldsymbol{r}^{\text{ref}}$ is a six-dimensional vector space (*i.e.*, $I = 6$) whose basis is given by

$$\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,6} = \left\{\begin{pmatrix} r_x^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} r_y^{\text{ref}} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ r_x^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 0 \\ r_y^{\text{ref}} \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\},$$

where we have omitted the parameter t to shorten the notation. Note that the choice of the basis is not unique. However, different bases w.r.t. to a given transformation describe the same space.

C. Construction of Vector Spaces

The vector spaces that are the most useful are summarized in Table I. They are defined by the bases $\{\boldsymbol{e}_i\}_{i=1,\dots,I}$ that construct a vector space $\mathcal{S}^{\text{ref}}$ for transformations in 2D. Taking a reference curve $\boldsymbol{r}^{\text{ref}} = (r_x^{\text{ref}}, r_y^{\text{ref}})$ and choosing one of the transformations given in Table I, the corresponding vector space is spanned by the indicated basis. While the definition of those spaces appears to be rather simple *a posteriori*, we are not aware of prior work that explicitly exploits this formulation.

D. Orthogonal Projectors

We now consider the projection operator $\mathcal{P} : \mathcal{H} \to \mathcal{S}$, $\boldsymbol{r} \mapsto \mathcal{P}\boldsymbol{r}$, that projects an arbitrary curve in $\mathcal{H}$ onto the vector space $\mathcal{S}$ with basis $\{\boldsymbol{e}_i\}_{i=1,\dots,I}$ and dimension I. Thus, a vector space can either be explicitly defined by $\mathcal{S}$ or implicitly by $\mathcal{P}$. It is expressed in its most general way as $\mathcal{P}\boldsymbol{r}(t) = \sum_{i=1}^{I} \boldsymbol{e}_i(t)\langle\tilde{\boldsymbol{e}}_i, \boldsymbol{r}\rangle$, where $\{\tilde{\boldsymbol{e}}_i\}_{i=1,\dots,I} \in \mathcal{S}$ is the unique dual basis with respect to $\{\boldsymbol{e}_i\}_{i=1,\dots,I}$ such that $\langle \boldsymbol{e}_i, \tilde{\boldsymbol{e}}_j\rangle = \delta_{i-j}$, with δ_{i-j} being the Kronecker delta. The operator $\mathcal{P}$ is an orthonormal projector and belongs to the class of orthogonal projection operators.

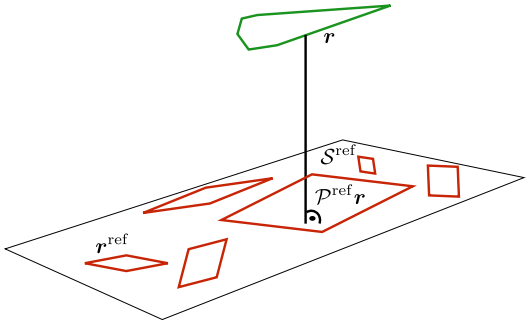

Fig. 2. Illustration of an orthogonal projection onto a vector space. The plane denoted by S^{ref} represents the subspace defined by the reference shape $\boldsymbol{r}^{\text{ref}}$. S^{ref} represents a subspace that contains all curves $\boldsymbol{r}^{\text{ref}}$ up to a class of transformations (*e.g.*, rotations, scaling, or translations of $\boldsymbol{r}^{\text{ref}}$). Projecting a query curve $\boldsymbol{r}$ (green curve) orthogonally onto S^{ref} amounts to identifying the rotated, scaled, and translated quadrilateral $\boldsymbol{r}^{\text{ref}}$ that is closest to $\boldsymbol{r}$ w.r.t. a chosen distance measure. The curve obtained by the orthogonal projection is denoted as $\mathcal{P}^{\text{ref}}\boldsymbol{r}$.

Orthogonal projectors are of special interest to us because they minimize the distance between the query curve $\boldsymbol{r} \in \mathcal{H}$ and its projection $\mathcal{P}\boldsymbol{r}$ onto $\mathcal{S}$ w.r.t. the norm induced by the L_2-inner product (see Figure 2). Proposition 1 provides a mean to directly compute the orthogonal projector $\mathcal{P}$ given a basis $\{\boldsymbol{e}_i\}_{i=1,\dots,I}$ spanning the vector space $\mathcal{S}$.

Proposition 1: The orthogonal projector $\mathcal{P}^{\text{ref}} : \mathcal{H} \to \mathcal{S}^{\text{ref}}$ that minimizes the distance between the curve $\boldsymbol{r} \in \mathcal{H}$ and the I-dimensional vector space $\mathcal{S}^{\text{ref}}$ is specified by

$$\mathcal{P}^{\text{ref}}\boldsymbol{r}(t) = \langle \boldsymbol{K}_{\mathcal{P}^{\text{ref}}}(t, \cdot), \boldsymbol{r}\rangle,$$

where $\boldsymbol{K}_{\mathcal{P}^{\text{ref}}}(t,s) = \sum_{i=1}^{I} \boldsymbol{e}_i^{\text{ref}}(t) \otimes \tilde{\boldsymbol{e}}_i^{\text{ref}}(s)$ is the kernel of the operator $\mathcal{P}^{\text{ref}}$ and $\{\tilde{\boldsymbol{e}}_i^{\text{ref}}\}_{i=1,\dots,I}$ is the dual basis of $\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,I}$. Its elements are given by

$$\tilde{\boldsymbol{e}}_i^{\text{ref}} = \left[\mathbf{G}^{\text{ref}\,-1}\right]_{i,1} \boldsymbol{e}_1^{\text{ref}} + \cdots + \left[\mathbf{G}^{\text{ref}\,-1}\right]_{i,I} \boldsymbol{e}_I^{\text{ref}},$$

where $\mathbf{G}^{\text{ref}}$ is the Gram matrix of the basis $\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,I}$. Here, $\otimes$ denotes the tensor product between two vectors and is defined as $\boldsymbol{e}_i(t) \otimes \boldsymbol{e}_j(s) = \boldsymbol{e}_i(t)\boldsymbol{e}_i^{\mathsf{T}}(s)$.

The derivation of Proposition 1 is provided in Appendix A. We say that $\mathcal{P}^{\text{ref}}$ projects $\boldsymbol{r} \in \mathcal{H}$ onto the I-dimensional *invariant* subspace $\mathcal{S}^{\text{ref}}$. In particular, for any $\boldsymbol{r}^{\mathcal{S}^{\text{ref}}} \in \mathcal{S}^{\text{ref}}$, we have that $\boldsymbol{r}^{\mathcal{S}^{\text{ref}}} = \mathcal{P}^{\text{ref}}\boldsymbol{r}^{\mathcal{S}^{\text{ref}}}$.

IV. MEAN SHAPE AND ALIGNMENT

In the case where we are dealing with several reference curves (*i.e.*, a training set of reference shapes), we define one vector space $S_k := \mathcal{S}_k^{\text{ref}}$ for each curve $\boldsymbol{r}_k := \boldsymbol{r}_k^{\text{ref}}$. Merging all these subspaces results in a large space of transformations of different curves.

Since, in the training set, some shape configurations might occur more frequently than others, we want to construct the dominant or mean shape given the training data and a class of transformations. We assume that all subspaces have the same dimension I and formalize the problem as finding the curve that is closest to all the subspaces S_k, each being specified by its corresponding projector $\mathcal{P}_k := \mathcal{P}_k^{\text{ref}} : \boldsymbol{r} \mapsto \sum_{i=1}^{I} \boldsymbol{e}_i^k(t)\langle\tilde{\boldsymbol{e}}_i^k, \boldsymbol{r}\rangle$ (see Figure 1). This problem can be formulated

in a variational form if we impose the condition that the mean shape should have unit norm. Although arbitrary, this requirement does not influence the result; in practice, we are only interested in the shape up to a scaling factor. The mean curve $\boldsymbol{r}_{\text{mean}}$ is determined by maximizing the sum of all k projections of $\boldsymbol{r}_{\text{mean}}$ onto the subspaces $\mathcal{S}_k^{\text{ref}}$, which is equivalent to minimizing the sum of distances between $\boldsymbol{r}_{\text{mean}}$ and its projections onto $\mathcal{S}_k^{\text{ref}}$.

The curve $\boldsymbol{r}_{\text{mean}}$ that is closest to all subspaces S_k for $k = 1, \ldots, K$ is then obtained by solving

$$\boldsymbol{r}_{\text{mean}} = \arg\max_{\boldsymbol{r}} \sum_{k=1}^{K} \|\mathcal{P}_k \boldsymbol{r}\|_{L_2}^2 \quad \text{s.t. } \|\boldsymbol{r}\|_{L_2}^2 = 1, \tag{2}$$

which is equivalent to the eigenvalue problem

$$\sum_{k=1}^{K} \mathcal{P}_k \boldsymbol{r}_{\text{mean}}(t) = \lambda \boldsymbol{r}_{\text{mean}}(t) \quad \text{s.t. } \langle \boldsymbol{r}_{\text{mean},p}, \boldsymbol{r}_{\text{mean},q} \rangle = \delta_{p-q}, \tag{3}$$

where we have used the fact that all the $\mathcal{P}_k$ are orthonormal, which implies that $\mathcal{P}_k^* \mathcal{P}_k = \mathcal{P}_k$, where $\mathcal{P}_k^*$ is the adjoint of $\mathcal{P}_k$. A derivation of (3) is provided in Appendix B.

A. Solutions of the Eigenequation

To solve (2), we invoke Propositon 1 and reformulate Problem (3) as

$$\sum_{k=1}^{K} \mathcal{P}_k \boldsymbol{r}_{\text{mean}}(t) = \sum_{k=1}^{K} \langle \boldsymbol{K}_{\mathcal{P}_k}(t, \cdot), \boldsymbol{r}_{\text{mean}} \rangle = \lambda \boldsymbol{r}_{\text{mean}}(t). \tag{4}$$

Equation (4) is a Volterra equation whose kernel $\boldsymbol{K}_{\mathcal{P}}$ consists of a finite sum. In Theorem 1, we characterize the solutions of (4) as the principal components of the eigenequation (3).

Theorem 1: Let the $(K \cdot I) \times (K \cdot I)$ matrix Γ be defined as

$$[\Gamma]_{(k-1)\cdot I+i,(j-1)\cdot K+l} = \langle \tilde{\boldsymbol{e}}_i^{(k)}, \boldsymbol{e}_j^{(l)} \rangle, \tag{5}$$

where $k, l \in [1, \ldots, K]$ and $i, j \in [1 \ldots I]$. Then, the pth eigencurve of (3) is given as

$$\boldsymbol{r}_{\text{mean},p}(t) = \sum_{k=1}^{K} \sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)} \gamma_{ik}^{(p)}, \tag{6}$$

where $\gamma_{ik}^{(p)}$ is the entry indexed by $(i-1) \cdot K + k$ of the pth eigenvector of the matrix Γ.

The proof is given in Appendix C. We show in Appendix D how to interpret this result in practice.

1) Unbiased Curve Alignment: Finally, we associate to the training set $\{\boldsymbol{r}_k\}_{k=1,\ldots,K}$ the aligned curves

$$\{\tilde{\boldsymbol{r}}_k = \mathcal{P}_k \boldsymbol{r}_{\text{mean}}\}_{k=1,\ldots,K}, \tag{7}$$

as illustrated in Figure 1, where $\boldsymbol{r}_{\text{mean}}$ is the mean shape. The projection of the mean shape onto $\mathcal{S}_k$ amounts to choosing the affine transformation of each data curve $\boldsymbol{r}_k$ that brings it closest to $\boldsymbol{r}_{\text{mean}}$ within each vector space $\mathcal{S}_k$ (see Figure 2). It is worth noticing that the proposed method for aligning the curves does not depend on the location of any member of the training set within each subspace $\mathcal{S}_k$. Hence, in that sense, it is unbiased as well as invariant w.r.t. to the geometrical transformation that is chosen.

V. Projection-Based Functional PCA for Curves

We now construct an fPCA on the aligned training set (7). Since the curves $r \in \mathcal{H}$ are defined in the continous domain, it is not possible to apply a discrete-domain PCA to our data.[1] Here, our data is of dimension "$2\infty \times K$". Therefore, we use operators instead of matrices to perform an fPCA.

Definition 1: The (compact) data operator $\mathbf{X} : \mathbb{R}^K \to L_2([0,1], \mathbb{R}^2)$ is the operator whose kernel consists of K aligned curves as

$$\mathbf{X} = [\tilde{\boldsymbol{r}}_1(t) \cdots \tilde{\boldsymbol{r}}_K(t)],$$

where $\tilde{\boldsymbol{r}}_k$ is defined in (7). The adjoint $\mathbf{X}^* : L_2([0,1], \mathbb{R}^2) \to \mathbb{R}^K$ satisfies

$$\langle \boldsymbol{r}, \mathbf{X}\mathbf{v} \rangle_{L_2([0,1],\mathbb{R}^2)} = \langle \mathbf{X}^* \boldsymbol{r}, \mathbf{v} \rangle_{\ell_2(\mathbb{R}^K)}, \tag{8}$$

with $\mathbf{v} \in \mathbb{R}^K$ and $\boldsymbol{r} \in L_2([0,1], \mathbb{R}^2)$. We emphasize that each of the two inner products in (8) have their own distinct definition.

We are looking for the optimal orthogonal base curves $\{\boldsymbol{\xi}_1(t), \ldots, \boldsymbol{\xi}_K(t)\}$, $\boldsymbol{\xi}_k \in \mathcal{H}$ for $k = 1, \ldots, K$, that decorrelate the training set. They are given by the eigencurves of the scatter operator $\mathbf{X}\mathbf{X}^* : L_2([0,1], \mathbb{R}^2) \to L_2([0,1], \mathbb{R}^2)$. Analogous to the discrete PCA, we can exploit the property that

- the non-vanishing eigenvalues of the scatter operator $\mathbf{X}\mathbf{X}^*$ and of the Gram matrix $\mathbf{X}^*\mathbf{X} \in \mathbb{R}^{K \times K}$, which corresponds to the correlation matrix in discrete PCA, are identical;
- the eigencurves $\{\boldsymbol{\xi}_k(t)\}_{k=1,\ldots,K}$ of $\mathbf{X}\mathbf{X}^*$ are immediately obtained from the eigenvectors $\boldsymbol{v} \in \mathbb{R}^K$, as specified in Proposition 2.

Proposition 2: The eigencurves $\boldsymbol{\xi}_k \in L_2([0,1], \mathbb{R}^2)$ of the scatter operator $\mathbf{X}\mathbf{X}^* : L_2([0,1], \mathbb{R}^2) \to L_2([0,1], \mathbb{R}^2)$ are specified by

$$\mathbf{X}\mathbf{X}^*\{\boldsymbol{\xi}_k\}(t) = \lambda_k \boldsymbol{\xi}_k(t),$$

while the eigenvectors $\boldsymbol{v}_k \in \mathbb{R}^K$ of the Gram matrix $\mathbf{X}^*\mathbf{X} \in \mathbb{R}^{K \times K}$ are given by

$$(\mathbf{X}^*\mathbf{X})\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k,$$

where the λ_k are the non-vanishing eigenvalues of $\mathbf{X}^*\mathbf{X}$ and are identical to the non-vanishing eigenvalues of $\mathbf{X}\mathbf{X}^*$. These entities are related by

$$\boldsymbol{\xi}_k = \frac{1}{\sqrt{\lambda_k}} \mathbf{X}\boldsymbol{v}_k \text{ and } \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} \mathbf{X}^* \boldsymbol{\xi}_k.$$

Furthermore, the relation

$$\boldsymbol{v}_k^\mathsf{T} \boldsymbol{v}_l = \langle \boldsymbol{\xi}_k, \boldsymbol{\xi}_l \rangle = \delta_{k-l}$$

holds.

The Gram matrix has size $K \times K$ and is computed as

$$\mathbf{X}^*\mathbf{X} = \begin{pmatrix} \langle \tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_1 \rangle & \ldots & \langle \tilde{\boldsymbol{r}}_1, \tilde{\boldsymbol{r}}_K \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{\boldsymbol{r}}_K, \tilde{\boldsymbol{r}}_1 \rangle & \ldots & \langle \tilde{\boldsymbol{r}}_K, \tilde{\boldsymbol{r}}_K \rangle \end{pmatrix}.$$

[1] In the discrete domain a training set with K curves — each curve being defined by Q landmarks or samples given by their coordinates — is represented as a $2Q \times K$ data matrix and then a discrete domain PCA is performed [27].

Now, we can easily compute the principal curves by specifying the data array $\mathbf{Z}$ as

$$\mathbf{Z} = \mathbf{X}\mathbf{V}, \tag{9}$$

where $\mathbf{Z} = [z_1(t) \cdots z_K(t)]$ and $\mathbf{V} = [\boldsymbol{v}_1 \dots \boldsymbol{v}_K]$ is the orthonormal matrix containing the eigenvectors of the Gram matrix $\mathbf{X}^*\mathbf{X} \in \mathbb{R}^{K\times K}$. They can also be computed via the relation

$$\mathbf{Z} = [\sqrt{\lambda}_1 \boldsymbol{\xi}_1(t) \cdots \sqrt{\lambda}_I \boldsymbol{\xi}_K(t)].$$

For a more in-depth description of fPCA using compact operators, we refer the reader to [28].

VI. Implementation With Landmark-Based Spline Curves

We now illustrate how our framework can be implemented using spline curves. For simplicity, we consider that the curves all have the same number N of control points and are constructed with the same basis function φ.

A. Parametric Spline-Based Curves

We consider spline curves of the form

$$\boldsymbol{r}(t) = \begin{pmatrix} r_x(t) \\ r_y(t) \end{pmatrix} = \sum_{n=0}^{N-1} \boldsymbol{c}[n]\varphi_n(t), \tag{10}$$

where φ is a compactly supported spline-based generator function and $N \in \mathbb{Z}^+$ represents the number of *control points* of the curve. The spline coefficients are given by $\{\boldsymbol{c}[n] = (c_x[n], c_y[n])\}_{n=0,\dots,N-1}$. To guarantee a stable and unique representation of a spline curve (10) by its control points, φ needs to generate a Riesz basis [29] as, for instance, polynomial B-splines do.

1) Affine Covariance: To represent a curve independently from its location and orientation, the representation needs to be affine covariant so that

$$\mathbf{A}\,\mathbf{r}(t) + \boldsymbol{b} = \sum_{n=0}^{N-1} (\mathbf{A}\,\mathbf{c}[n] + \boldsymbol{b})\ \varphi_n(t).$$

It is easy to show that affine invariance is guaranteed if and only if φ satisfies the partition-of-unity condition $\sum_{n\in\mathbb{Z}} \varphi_n(t) = 1$ for all $t \in \mathbb{R}$.

B. Inner Product of Spline-Based Curves

We use a simple but powerful expression to compute the L_2-inner product $\langle \boldsymbol{r}_1, \boldsymbol{r}_2\rangle$ between spline-based curves. We first compute it for the 1D case and then generalize it to higher dimensions.

1) 1D Inner Product: We consider spline-based (coordinate) functions of the form $x(t) = \sum_{n=0}^{N-1} c_x[n]\varphi_n(t)$. The L_2-inner product is then expressed as

$$\begin{aligned}\langle x_1, x_2\rangle &= \int_0^1 x_1(t)x_2(t)\mathrm{d}t \\ &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} c_{1x}[n]c_{2x}[m]\int_0^1 \varphi_n(t)\varphi_m(t)\mathrm{d}t. \end{aligned}\tag{11}$$

We collect all coefficients of the function x_i in the vector of length N, $\mathbf{c}_{ix} = (c_{ix}[0], \dots, c_{ix}[N-1])$ with $i = 1$ or 2. We then define

$$[\boldsymbol{\Phi}]_{n,m} := \int_0^1 \varphi_n(t)\varphi_m(t)\mathrm{d}t. \tag{12}$$

Now, (11) is expressed as $\langle x_1, x_2\rangle = \mathbf{c}_{1x}^{\mathsf{T}}\boldsymbol{\Phi}\mathbf{c}_{2x}$, where $\boldsymbol{\Phi}$ is the $(N \times N)$ *correlation matrix* of φ_n. For an implementation (11) can be crucial: since the entries of the matrix $\boldsymbol{\Phi}$ can be precomputed, the evaluation of the integral associated with the inner product (11) boils down to a matrix-vector multiplication, which reduces the computational time considerably.

2) 2D Inner Products: To simplify the 2D inner product, we similarly define

$$\mathbf{c}_i = (\mathbf{c}_{ix}, \mathbf{c}_{iy}), \tag{13}$$

which is now a vector of length $2N$. The corresponding inner product is

$$\langle \boldsymbol{r}_1, \boldsymbol{r}_2\rangle = \mathbf{c}_1^{\mathsf{T}}\boldsymbol{\Psi}\mathbf{c}_2 = \langle \mathbf{c}_1, \mathbf{c}_2\rangle_{\boldsymbol{\Psi}}, \tag{14}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} \end{bmatrix} \tag{15}$$

and $\mathbf{0}$ is a null matrix with the same dimensions as $\boldsymbol{\Phi}$ defined by (12). We show in [30] how (12) is computed when the curves $\boldsymbol{r}$ are periodic. (We use different fonts to distinguish $\mathbf{c}$ in (14) from $\boldsymbol{c}$ in (10).)

C. Orthogonal Spline Projectors

Using (14) to compute inner products of spline curves, we now specify the projection operator that corresponds to Proposition 1. A fundamental aspect of our construction is that both the query curve $\boldsymbol{r}$ that is being projected and the basis $\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,I}$ of the subspace $\mathcal{S}^{\text{ref}}$ as defined in (1) are spline curves of the form given by (10). Hence, each curve $\boldsymbol{e}(t) \in \mathcal{H}$ is uniquely determined by its corresponding vector of control points $\mathbf{c}_{\boldsymbol{e}} \in \mathbb{R}^{2N}$. We define the matrix

$$\mathbf{C}^{\text{ref}} = [\mathbf{c}_{\boldsymbol{e}_1^{\text{ref}}} \cdots \mathbf{c}_{\boldsymbol{e}_I^{\text{ref}}}]. \tag{16}$$

It has dimension $(2N \times I)$ and contains the control points of the curves $\{\boldsymbol{e}_i^{\text{ref}}\}_{i=1,\dots,I}$ that define a basis of $\mathcal{S}^{\text{ref}}$. To simplify the notation, we collect all basis functions in the vector

$$\boldsymbol{\varphi}(t) := (\varphi_0(t), \dots, \varphi_{N-1}(t)). \tag{17}$$

The corresponding spline projector is then specified by Theorem 2.

Theorem 2: Let $\boldsymbol{r}(t) = \boldsymbol{\varphi}(t)^{\mathsf{T}}\boldsymbol{c}$. Then,

$$\mathcal{P}^{\text{ref}}\boldsymbol{r}(t) = \begin{pmatrix} \boldsymbol{\varphi}(t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(t) \end{pmatrix}^{\mathsf{T}} \mathbf{P}^{\text{ref}}\mathbf{c},$$

where $\mathbf{P}^{\text{ref}} : \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ is the $(2N \times 2N)$ projection matrix defined as

$$\mathbf{P}^{\text{ref}} = \mathbf{C}^{\text{ref}}(\mathbf{C}^{\text{ref}\,\mathsf{T}}\Psi\mathbf{C}^{\text{ref}})^{-1}\mathbf{C}^{\text{ref}\,\mathsf{T}}\boldsymbol{\Psi}$$

and $\mathbf{0}$ is an N-dimensional null vector.

Theorem 2 provides a direct method to compute the control points of the projected curve. Note that the projection of the vector $\mathbf{c}$ of control points is itself not orthogonal. However, it corresponds to the orthogonal projection of $\boldsymbol{r}$ in the L_2-sense. Therefore, we have $(\mathbf{P}^{\text{ref}})^2 = \mathbf{P}^{\text{ref}}$ and $\mathbf{P}^{\text{ref}\,\mathsf{T}} \neq \mathbf{P}^{\text{ref}}$. Theorem 2 shows that $\mathbf{P}^{\text{ref}}$ is an oblique projector from $\mathbb{R}^{2N}$ onto the I-dimensional invariant subspace of $\mathbb{R}^{2N}$ defined by the basis $\{\mathbf{c}_{\boldsymbol{e}_i^{\text{ref}}}\}_{i=1,\dots,I}$. This means that $\mathcal{P}^{\text{ref}}$ is an orthogonal projector in the L_2-sense and is efficiently implemented via the oblique projector $\mathbf{P}^{\text{ref}}$ in $\mathbb{R}^{2N}$.

We now provide examples that illustrate how some of the projectors listed in Table I are implemented using splines and Theorem 2. Thereby, the vector of control points $\mathbf{c}^{\text{ref}} = (\mathbf{c}_x^{\text{ref}}, \mathbf{c}_y^{\text{ref}})$ of a reference curve $\boldsymbol{r}^{\text{ref}}$ is specified in accordance with (13).

1) Scaling Projector (Without Translation): The scaling projector can be expressed by solving $\min_a \|a\boldsymbol{r}^{\text{ref}} - \boldsymbol{r}\|_{L_2}^2$ such that $\mathcal{P}^{\text{ref}}\boldsymbol{r}(t) = a\boldsymbol{r}^{\text{ref}}(t)$, where $a \in \mathbb{R}$ and $\boldsymbol{r}^{\text{ref}}$ is the reference curve that defines the vector space. Its well-known solution is $a = \frac{\langle \boldsymbol{r}^{\text{ref}}, \boldsymbol{r}\rangle}{\langle \boldsymbol{r}^{\text{ref}}, \boldsymbol{r}^{\text{ref}}\rangle}$. Using (14), the corresponding spline projector is specified by $\mathbf{P}^{\text{ref}} = \mathbf{c}^{\text{ref}} \frac{\mathbf{c}^{\text{ref}\,\mathsf{T}}\boldsymbol{\Psi}}{\langle \mathbf{c}^{\text{ref}}, \mathbf{c}^{\text{ref}}\rangle_{\boldsymbol{\Psi}}}$, which corresponds to the solution obtained by the direct application of Theorem 2.

2) Affine Transformation: The example illustrated in Section III-B corresponds to

$$\{\mathbf{c}_{\boldsymbol{e}_i^{\text{ref}}}\}_{i\in[1\dots6]} = \left\{\begin{pmatrix}\mathbf{c}_x^{\text{ref}}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{c}_y^{\text{ref}}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{0}\\ \mathbf{c}_x^{\text{ref}}\end{pmatrix}, \begin{pmatrix}\mathbf{0}\\ \mathbf{c}_y^{\text{ref}}\end{pmatrix}, \begin{pmatrix}\mathbf{1}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{0}\\ \mathbf{1}\end{pmatrix}\right\},$$

where $\mathbf{0}$ and $\mathbf{1}$ are vectors of size N (which is the size of $\mathbf{c}_x^{\text{ref}}$ or $\mathbf{c}_y^{\text{ref}}$) and whose elements are all 0 or 1, respectively. The spline projector is then computed by the application of Theorem 2.

3) Similarity: The similarity transformation is defined as the scaling of a curve $\boldsymbol{r}$ by a factor a combined with a rotation described by the rotation matrix $\mathbf{R}_\theta$ (applied to $\boldsymbol{r}^{\text{ref}}$) and a translation given by $\boldsymbol{b} = (b_1, b_2)$. It is expressed as

$$\begin{aligned} a\mathbf{R}_\theta\boldsymbol{r}^{\text{ref}} + \boldsymbol{b} &= \begin{pmatrix} a\cos(\theta)r_x^{\text{ref}} - a\sin(\theta)r_y^{\text{ref}} + b_1 \\ a\sin(\theta)r_x^{\text{ref}} + a\cos(\theta)r_y^{\text{ref}} + b_2 \end{pmatrix} \\ &= \alpha\begin{pmatrix} r_x^{\text{ref}} \\ r_y^{\text{ref}}\end{pmatrix} + \beta\begin{pmatrix} -r_y^{\text{ref}} \\ r_x^{\text{ref}}\end{pmatrix} + b_1\begin{pmatrix}1\\0\end{pmatrix} + b_2\begin{pmatrix}0\\1\end{pmatrix}, \end{aligned}$$

where $a \in \mathbb{R}$, $\alpha = a\cos\theta$, and $\beta = a\sin\theta$. To construct the corresponding projector, we choose $\boldsymbol{e}_1^{\text{ref}} = (r_x^{\text{ref}}, r_y^{\text{ref}})$, $\boldsymbol{e}_2^{\text{ref}} = (-r_y^{\text{ref}}, r_x^{\text{ref}})$, $\boldsymbol{e}_3^{\text{ref}} = (1, 0)$, and $\boldsymbol{e}_4^{\text{ref}} = (0, 1)$, which corresponds to the basis $\mathbf{c}_{\boldsymbol{e}_1^{\text{ref}}} = (\mathbf{c}_x^{\text{ref}}, \mathbf{c}_y^{\text{ref}})$, $\mathbf{c}_{\boldsymbol{e}_2^{\text{ref}}} = (-\mathbf{c}_y^{\text{ref}}, \mathbf{c}_x^{\text{ref}})$, $\mathbf{c}_{\boldsymbol{e}_3^{\text{ref}}} = (\mathbf{1}, \mathbf{0})$, and $\mathbf{c}_{\boldsymbol{e}_4^{\text{ref}}} = (\mathbf{0}, \mathbf{1})$.

D. Example

We compare the affine space with the space defined by the similarity transformation. We construct the two corresponding projectors w.r.t. the reference spline curve that represents the white-matter structure of the brain, as shown in the left of Figure 3. We then project the curve shown in the right of Figure 3 (the *corpus callosum*) separately onto the affine, as well as onto the vector space defined by the similarity transformation. Among all shapes enclosed by the given subspace defined by the reference shape (*i.e.*, white matter), the projector chooses the one closest to the *corpus callosum* (see Figure 4).

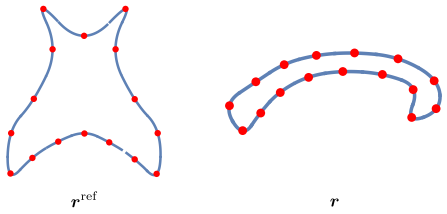

Fig. 3. Left: representation of a white matter segment of a brain. Right: contour of a *corpus callosum* (brain structure). The blue contour is represented as a spline curve and the red dots are its landmarks (*i.e.*, the 2D spline coefficients given by $\{\boldsymbol{c}[k]\}_{k\in\mathbb{Z}}$).

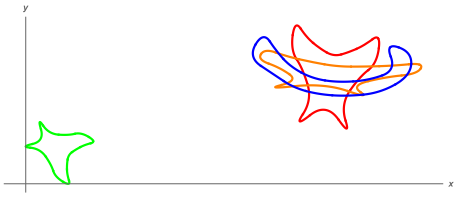

Fig. 4. Affine vs. similarity. The *corpus callosum* (blue) is registered onto the white matter (green). The orange curve is the closest affine transformation of the white matter (green) w.r.t. the *corpus callosum* (blue). The red curve is the closest deformed white matter (green) w.r.t. the *corpus callosum* (blue) using a similarity transformation.

E. Mean Spline Shape

To compute the mean shape $\boldsymbol{r}_{\text{mean}}$ using splines, we again take advantage of the unicity of the representation of spline curves by their coefficients (Riesz-basis property). We directly compute the vector of control points that defines $\boldsymbol{r}_{\text{mean}}$. Proposition 3 characterizes the spline-based solution that corresponds to the eigenvalue problem stated in (2).

Proposition 3: Assume a training set of K spline curves $\boldsymbol{r}_k^{\text{ref}}$ of the form (10), where each curve defines a vector space $\mathcal{S}_k^{\text{ref}}$ through the spline coefficients given by the $(2N \times I)$ matrix $\mathbf{C}_k := \mathbf{C}_k^{\text{ref}}$ as specified by (16). Then, the vector of control points $\mathbf{c}_{\text{mean}}$ of the spline curve $\boldsymbol{r}_{\text{mean}}$ is given as the solution of the eigenequation

$$\sum_{k=1}^{K} \mathbf{C}_k(\mathbf{C}_k^{\mathsf{T}}\boldsymbol{\Psi}\mathbf{C}_k)^{-1}\mathbf{C}_k^{\mathsf{T}}\boldsymbol{\Psi}\mathbf{c}_{\text{mean}} = \lambda\mathbf{c}_{\text{mean}}.$$

The proof is provided in Appendix E.

F. Functional PCA for Spline Curves

Since $\{\varphi_n\}_{n=0,\dots,N-1}$ forms a Riesz basis, the data array $\mathbf{X}$ in Definition 1 is fully specified by the matrix

$$\boldsymbol{\Omega} = [\mathbf{c}_1 \cdots \mathbf{c}_K] \tag{18}$$

of control points that define the curves $\{\tilde{\boldsymbol{r}}_k\}_{k=1,\dots,K}$. Using (14), the Gram matrix of $\mathbf{X}$ is computed as

$$\mathbf{X}^*\mathbf{X} = \boldsymbol{\Omega}^{\mathsf{T}}\boldsymbol{\Psi}\boldsymbol{\Omega} \tag{19}$$

and, hence, the $(2N \times K)$ matrix $\boldsymbol{\Omega}_{\mathbf{Z}}$ that contains the control points of the principal curves is

$$\boldsymbol{\Omega}_{\mathbf{Z}} = \boldsymbol{\Omega}\mathbf{V}, \tag{20}$$

where $\mathbf{V}$ is the orthonormal matrix that contains the eigenvectors of the Gram matrix as detailed in Section V. The principal curves z_k are finally obtained as

$$z_k(t) = ([\Omega_{\mathbf{Z}}]_{*,k})^{\mathsf{T}} \begin{pmatrix} \boldsymbol{\varphi}(t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(t) \end{pmatrix}, \tag{21}$$

where $[\Omega_{\mathbf{Z}}]_{*,k}$ denotes the kth column of $\Omega_{\mathbf{Z}}$. More generally, we have

$$\mathbf{Z} = \Omega_{\mathbf{Z}}^{\mathsf{T}} \begin{pmatrix} \boldsymbol{\varphi}(t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(t) \end{pmatrix}. \tag{22}$$

VII. Sparse Shape Encoding

PCA uses the complete data to compute the principal curves. This makes it prone to outliers which might compromise robustness when learning a shape dictionary. We now propose a dictionary learning approach that only uses a sparse subset of the data to encode the shapes. For this purpose, we first derive a property specific to the spline representation of curves. It allows us to *exactly* measure the continuous-domain L_2 norm using a discrete-domain ℓ_2 norm.

A. L_2-ℓ_2 Norm Equality

Theorem 3: For any spline curve $\boldsymbol{r}(t)$ specified by the vector of control points $\mathbf{c}$, and for any data array $\mathbf{D} = [\boldsymbol{d}_1(t) \cdots \boldsymbol{d}_K(t)]$ whose elements are parametric spline curves described by the matrix of control points $\Omega_{\mathbf{D}} = [\mathbf{c}_{\boldsymbol{d}_1} \cdots \mathbf{c}_{\boldsymbol{d}_K}]$ and any $\boldsymbol{\alpha} \in \mathbb{R}^K$ we have the norm equality

$$\|\boldsymbol{r} - \mathbf{D}\boldsymbol{\alpha}\|_{L_2}^2 = \|\tilde{\boldsymbol{c}} - \tilde{\mathbf{D}}\boldsymbol{\alpha}\|_{\ell_2}$$

with

$$\tilde{\boldsymbol{c}} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\mathbf{c}, \tag{23}$$

$$\tilde{\mathbf{D}} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\Omega_{\mathbf{D}}, \tag{24}$$

and

$$\Psi = \mathbf{Q}\Lambda\mathbf{Q}^{-1}, \tag{25}$$

where $\mathbf{Q}$ is an orthonormal matrix whose columns are the unit-norm eigenvectors of Ψ and Λ is the diagonal matrix that contains the eigenvalues of Ψ defined by (15).

Proof: We develop the L_2 norm as

$$\begin{aligned} \|\boldsymbol{r} - \mathbf{D}\boldsymbol{\alpha}\|_{L_2}^2 &= \left\| \begin{pmatrix} \boldsymbol{\varphi}(t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(t) \end{pmatrix}^{\mathsf{T}} (\mathbf{c} - \Omega_{\mathbf{D}}\boldsymbol{\alpha}) \right\|_{L_2}^2 \\ &= (\mathbf{c} - \Omega_{\mathbf{D}}\boldsymbol{\alpha})^{\mathsf{T}} \Psi (\mathbf{c} - \Omega_{\mathbf{D}}\boldsymbol{\alpha}), \end{aligned} \tag{26}$$

with $\boldsymbol{\varphi}$ as defined in (17). Since Ψ is a positive-semidefinite symmetric matrix, it admits an eigen-decomposition of the form

$$\Psi = \mathbf{Q}\Lambda\mathbf{Q}^{-1} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}, \tag{27}$$

where $\mathbf{Q}$ is an orthonormal matrix that satisfies $\mathbf{Q}^{-1} = \mathbf{Q}^{\mathsf{T}}$, whose columns are the unit-norm eigenvectors of Ψ, and Λ is the diagonal matrix that contains the eigenvalues of Ψ. Therefore, we have

$$\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1} = \left(\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\right)^{\mathsf{T}}, \tag{28}$$

which allows us to express (26) as

$$\begin{aligned} \|\boldsymbol{r} - \mathbf{D}\boldsymbol{\alpha}\|_{L_2}^2 &= \|\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}(\mathbf{c} - \Omega_{\mathbf{D}}\boldsymbol{\alpha})\|_{\ell_2}^2 \\ &= \|\underbrace{\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\mathbf{c}}_{\tilde{\mathbf{c}}} - \underbrace{\mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\Omega_{\mathbf{D}}}_{\tilde{\mathbf{D}}}\boldsymbol{\alpha}\|_{\ell_2}^2. \end{aligned} \tag{29}$$

■

B. Continuous-Domain Sparse Dictionary Learning

The projection-based fPCA described in Section V is a purely L_2-based method. It is well known that such methods are sensitive to outliers, as well as to imbalanced or inhomogeneous data sets. Hence, there exist practical settings where those models are less suitable. Another limitation of fPCA is the orthogonality constraint on the eigencurves, which might be unnecessary and too restrictive in certain scenarios.

Here again, we consider the training set $\mathbf{X} = [\boldsymbol{x}_1(t) \cdots \boldsymbol{x}_K(t)]$ of parametric curves that are defined in the continuous domain as specified in Theorem 3. However, we now aim at constructing a dictionary $\mathbf{D}(t) = [\boldsymbol{d}_1(t) \cdots \boldsymbol{d}_J(t)]$ with $J \leq K$, where $\{\boldsymbol{d}_j(t)\}_{j=1,\ldots,J}$ is a set of parametric curves such that $\mathbf{D}$ yields the optimal value of the continuous-domain *sparse coding* problem. This problem is defined in analogy to its discrete counterpart [2], [31] as

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha} \in \mathbb{R}^J}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{x}_k - \mathbf{D}\boldsymbol{\alpha}_k\|_{L_2}^2 + \lambda\|\boldsymbol{\alpha}_k\|_{l_1} \right\} \tag{30}$$

for every $\boldsymbol{x}_k(t)$ in the training set, where $\lambda \in \mathbb{R}$ is a regularization parameter that controls sparsity. The problem (30) is well studied [32] and known as the Lasso [33] method or basis pursuit [34]. On one hand, if we enforce orthonormality on $\boldsymbol{\alpha}$ instead of sparsity (*i.e.*, $\langle \boldsymbol{\alpha}_k, \boldsymbol{\alpha}_l, \rangle = \delta_{k-l}$ and $\lambda = 0$), then we recover the exact fPCA solution (9) with $\boldsymbol{\alpha}_k = \boldsymbol{v}_k$. On the other hand, for $\lambda > 0$, we obtain a sparse vector $\boldsymbol{\alpha}_k$.

However, our goal here is to accurately approximate a shape $\boldsymbol{x}(t) \approx \mathbf{D}(t)\boldsymbol{\alpha}$ such that each curve $\boldsymbol{x}$ only uses a few elements of $\mathbf{D}$ for its representation. We make use of spline curves and invoke Theorem 3, which allows us to formulate the continuous-domain sparse-coding problem in the discrete domain as

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^J}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{x}_k - \mathbf{D}\boldsymbol{\alpha}_k\|_{L_2}^2 + \lambda\|\boldsymbol{\alpha}_k\|_{\ell_1} \right\} \tag{31}$$

$$= \underset{\boldsymbol{\alpha} \in \mathbb{R}^J}{\arg\min} \left\{ \frac{1}{2}\|\tilde{\boldsymbol{x}}_k - \tilde{\mathbf{D}}\boldsymbol{\alpha}_k\|_{\ell_2}^2 + \lambda\|\boldsymbol{\alpha}_k\|_{\ell_1} \right\}. \tag{32}$$

Here,

$$\tilde{\boldsymbol{x}}_k = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}[\Omega]_{*,k} \tag{33}$$

with $[\Omega]_{*,k}$ being the vector of control points of the kth curve of $\mathbf{X}$ as specified in (18) and

$$\tilde{\mathbf{D}} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^{-1}\Omega_{\mathbf{D}} = [\tilde{\boldsymbol{d}_1} \cdots \tilde{\boldsymbol{d}_J}], \tag{34}$$

with $\Omega_{\mathbf{D}}$ being the matrix of control points that describe the parametric curves, in other words the atoms that form the continuous domain dictionary $\mathbf{D}(t)$.

To solve the discrete-domain sparse-coding problem, we prevent $\tilde{\mathbf{D}}$ from becoming arbitrarily large by enforcing the ℓ_2-norm of its column vectors to not exceed unity.

As suggested in [1], [2], this allows us to define the convex set of possible dictionaries as

$$\mathcal{C} := \{\tilde{\mathbf{D}} \in \mathbb{R}^{2N\times J} \quad \text{s.t.} \quad \|\tilde{\boldsymbol{d}}_j\|_{\ell_2} \leq 1,\, j = 1, \ldots, J\}, \tag{35}$$

where N is the number of control points of a spline curve (10). Now, $\tilde{\mathbf{D}}$ is found by solving the joint-optimization problem

$$(\tilde{\mathbf{D}}^*, \boldsymbol{\alpha}^*) = \underset{\tilde{\mathbf{D}}\in\mathcal{C},\boldsymbol{\alpha}\in\mathbb{R}^J}{\arg\min} \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{2}\|\tilde{\boldsymbol{x}}_k - \tilde{\mathbf{D}}\boldsymbol{\alpha}_k\|_{\ell_2}^2 + \lambda\|\boldsymbol{\alpha}_k\|_{\ell_1}\Big), \tag{36}$$

which is convex w.r.t. the two variables $\tilde{\mathbf{D}}$ and $\boldsymbol{\alpha}$ when one of them is fixed. Finally, from (34), we see that

$$\boldsymbol{\Omega}_{\mathbf{D}} = \mathbf{Q}\boldsymbol{\Lambda}^{-1/2}\mathbf{Q}^{-1}\tilde{\mathbf{D}}$$

and therefore, the continuous-domain dictionary is computed through

$$\mathbf{D}(t) = \begin{pmatrix}\boldsymbol{\varphi}(t) & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\varphi}(t)\end{pmatrix}^{\mathsf{T}} \boldsymbol{\Omega}_{\mathbf{D}} \tag{37}$$

with $\boldsymbol{\varphi}$ as defined in (17).

1) Optimization: The joint-optimization problem (36) can be solved by alternating methods which keep one variable fixed while minimizing the other, as described in [31], [35], [36]. Here we make use of the online optimization algorithm that is based on stochastic approximations [37], [38] and implemented in the popular SPAMS library written by Mairal *et al.* [1], [2]. It minimizes sequentially a quadratic local approximation of the expected cost function and is well suited to the efficient handling of large training sets. Since the focus of this article is not the optimization itself, we refer the reader to [1], [2] for a detailed description of the algorithm and its implementation.

VIII. Comparison With Existing Methods

A. Linear Methods

The classical approach to learn dictionaries is to consider K shapes that are described by an ordered set of N points or landmarks in $\mathbb{R}^2$ [9]. The shapes themselves are represented as one large vector $\boldsymbol{r}_k \in \mathbb{R}^{2N}$ with $k \in [1 \ldots K]$. They are geometrically normalized by aligning them to a common reference in order to remove some effects of rigid-body transformations. The alignment to the reference shape $\boldsymbol{r}^{\text{ref}}$ is computed as $\tilde{\boldsymbol{r}}_k = \mathbf{A}\boldsymbol{r}_k + \mathbf{b}$, where $\mathbf{A}$ is an affine transformation matrix and $\boldsymbol{b} \in \mathbb{R}^2$ a translation vector such that they solve $\min_{\mathbf{A},\boldsymbol{b}} \|\boldsymbol{r}^{\text{ref}} - \mathbf{A}\boldsymbol{r}_k - \boldsymbol{b}\|_{\ell_2}^2$. A standard PCA is then applied to the set $\{\tilde{\boldsymbol{r}}_k\}_{k=1\ldots K}$ of aligned shapes. Aside from operating with data that are necessarily discrete, the standard approach has the drawback of being potentially biased because distances between normalized shapes generally differ from distances between non-normalized shapes.

The fundamental difference between the classical approach and our method lies in the different concepts that define projective geometry and affine geometry. We exploit the fact that the solution of $\min_{\mathbf{A},\boldsymbol{b}} \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$ can be expressed (in closed form) as the orthogonal projection $\mathbf{P}^x\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$, a property that holds for both discrete- and continuous-domain curves. This allows us to express the affine transformation as a projection onto a space that does not depend on the specific element $\boldsymbol{x}$ that lives in that space.

B. Closed-Form Solution for Continuous and Discrete Curves

Our approach in this paper is expressed in the continuous domain. In some applications, however, curves are defined by a discrete set of points. In this case, the solutions for spline-based curves can be applied because a continuously defined curve can always be built parametrically using the linear B-spline [24] as basis function (see Section IX-A3 for an example).

1) Equivalent Spline Solution Using Uniform Samples: One of the benefits of using a spline-based representation of curves is that it allows one to represent curves in the continuous domain with a small number N of control points. This becomes apparent when noticing that, for a uniformly discretized curve $\boldsymbol{r}$ given by the ordered set of points $\{\boldsymbol{r}(\frac{q}{Q})\}_{q=0,\ldots,Q}$ with $(Q+1)$ samples, we have that

$$\lim_{Q\to\infty}\frac{1}{Q}\sum_{q=0}^{Q}\Big|\boldsymbol{r}_1\Big(\frac{q}{Q}\Big) - \mathbf{A}\boldsymbol{r}_2\Big(\frac{q}{Q}\Big) - \boldsymbol{b}\Big|^2 = \int_0^1 |\boldsymbol{r}_1(t) - \mathbf{A}\boldsymbol{r}_2(t) - \boldsymbol{b}|^2 \mathrm{d}t.$$

We see that, while the continuously defined curve $\boldsymbol{r}(t)$ is expressed with N control points and corresponds to the projection matrix $\mathbf{P}$ of size $(2N \times 2N)$, the discrete curve $\boldsymbol{r}(\frac{q}{Q})$ is described with $Q \gg N$ points whose corresponding projection matrix is of size $(2Q \times 2Q)$. This shows that a continuous-domain spline-based approach can be implemented at no additional cost compared to a discrete approach that would depend on N points, although the continuously defined curve is equivalent to a discrete setting where the number of points tends towards infinity. Hence, to be equivalent, we would have to use many more discrete points.

IX. Validation and Experiments

A. Shape Analysis of Biological Microscopic Structures

In microscopy, typically, different samples of the same organism are studied as for instance a colony of cells or bacteria. Characterizing representative shapes of such colonies is important to study, for instance, the reaction of an organism when exposed to a certain type of drug or chemical substance or to elucidate their behavior in specific environments. Next, we provide an example of shape analysis using real biological data.

1) Learning Shape Priors: We have manually outlined the twenty chromosomes shown in the microscopic image in Figure 5 (top). The outlining has been done by interpolating twelve landmarks on the contours of the chromosomes with the basis functions proposed in [39], [40]. This procedure allowed us to obtain a spline-based curve description of each chromosome with landmarks that are corresponding throughout the data set.

The chromosomes share a similar symmetric approximate rod-shaped structure; however, they differ in size, orientation,

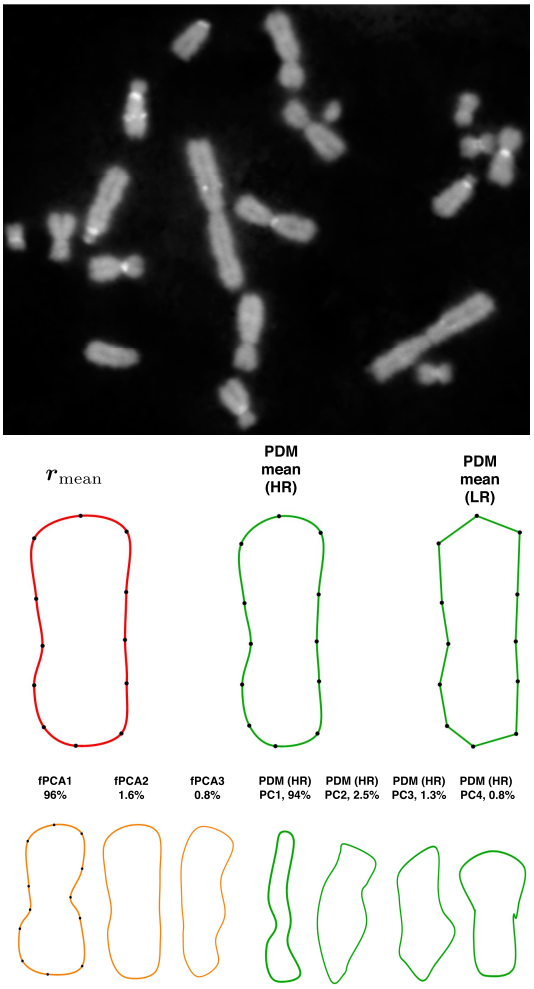

Fig. 5. Shape analysis of chromosome data. Top row: The data set consists of twenty chromosomes. They have been manually outlined by placing landmarks on the contours followed by spline interpolation. Middle row, left curve: $\boldsymbol{r}_{\text{mean}}$ obtained by Proposition 3. Middle row, green: Mean shapes obtained with the PDM with HR and LR. Bottom row: first eigenshapes obtained with fPCA (orange) and PDM (green). In "%" the shape variability is indicated, which is computed as $\lambda_i / \sum \lambda_i$, where λ_i stands for the ith eigenshape. "PC" stands for "principal component".

and location. Using our proposed framework, we first computed the aligned training set $\{\tilde{\boldsymbol{r}}_k\}_{k=1,\ldots,20}$ using the similarity transformation and then computed $\boldsymbol{r}_{\text{mean}}$ as given by Theorem 3. The resulting learned shape (Figure 5, red shape in middle row) can be further used either for classification (see Section IX-B) or as a trained shape prior for segmentation problems [41], [42]. It characterizes the population in terms of its shape and, hence, can be viewed as an average shape.

2) Learned Shape vs. Functional PCA vs. Point Distribution Model: To test the accuracy of the learned shape prior, we compared it to the first eigenshape obtained through the projection-based fPCA described in Section V and the mean shape obtained with the classical PDM (see Section II). The PDM being a discrete method based on linear interpolation between landmarks, we have computed two corresponding mean shapes: one of low resolution (LR) that corresponds to the number of landmarks used for the two continuous-domain models and another of high resolution (HR), where we have increased the number of samples fifty-fold, by inserting forty-nine samples between each original landmark (Figure 5, middle row). For each of the three models, fPCA, LR, and HR, we have computed the normalized correlation $\frac{\langle \boldsymbol{r}_{\text{model}}, \boldsymbol{r}_{\text{data}} \rangle}{\|\boldsymbol{r}_{\text{model}}\|_{L_2} \|\boldsymbol{r}_{\text{data}}\|_{L_2}}$ between the most representative shape $\boldsymbol{r}_{\text{model}}$ and each curve $\boldsymbol{r}_{\text{data}}$ in the data set. Here, $\boldsymbol{r}_{\text{model}}$ stands for either 1) $\boldsymbol{r}_{\text{mean}}$, 2) the mean shape obtained with the PDM, or 3) the first eigenshape "fPCA1" of the fPCA. The results are shown in Table II. We see that our method to compute the *learned* shape $\boldsymbol{r}_{\text{mean}}$ as the curve being closest to all the subspaces generated by the shapes of the data set, captures best shape variability. Further, the continuous-domain methods seem to yield a higher accuracy than the PDM. However, by increasing the resolution of the PDM we can approach the accuracy of the continuous-domain models, which validates the theoretical argument provided in Section VIII-B.

TABLE II
NORMALIZED CORRELATION BETWEEN PRINCIPAL SHAPES AND CHROMOSOME DATA

Data	$\boldsymbol{r}_{\text{mean}}$	fPCA	PDM (HR)	PDM (LR)
1	**0.958**	0.955	0.954	0.844
2	0.986	**0.988**	0.955	0.845
3	**0.986**	0.972	0.956	0.845
4	**0.984**	0.962	0.955	0.845
5	0.970	**0.974**	0.953	0.840
6	**0.982**	0.974	0.954	0.843
7	**0.995**	0.985	0.957	0.847
8	**0.987**	0.983	0.955	0.843
9	**0.985**	0.984	0.956	0.843
10	0.985	**0.986**	0.954	0.844
11	**0.981**	0.974	0.953	0.842
12	**0.960**	0.946	0.954	0.844
13	**0.973**	0.969	0.952	0.839
14	**0.965**	0.948	0.953	0.841
15	**0.981**	0.980	0.957	0.846
16	0.973	**0.983**	0.954	0.842
17	**0.977**	0.955	0.956	0.844
18	0.986	**0.987**	0.957	0.846
19	**0.996**	0.989	0.958	0.847
20	**0.994**	0.984	0.958	0.847
Mean	**0.980**	0.974	0.955	0.844
STD	0.010	0.013	0.001	0.002

3) Shape Reconstruction Using Projection-Based Functional PCA vs. Point Distribution Model: We now compare the shapes reconstructed through projection-based fPCA to those obtained by the PDM. From (9), we see that fPCA would allow for perfect reconstruction if all the eigencurves were used. In this section, however, we use the first four eigenvectors of the fPCA to approximate the data as $\boldsymbol{r}_{\text{data}}(t) \approx \boldsymbol{r}_{\text{recon}}^{\text{fPCA}}(t) = \sum_{i=1}^{4} a_i \boldsymbol{z}_i^{\text{fPCA}}(t)$. The $a_i \in \mathbb{R}$ are the coefficients that allow for the optimal approximation. The choice of using 4 eigenvectors is arbitrary but sufficient for our purpose since we already know from Section IX-A1 and Figure 5 that the first eigenshape captures 96% shape variability. For comparison, we compute the approximation obtained with the HR PDM, also using the first four eigenvectors. The PDM is thus expressed as $\boldsymbol{r}_{\text{data}} \approx \boldsymbol{r}_{\text{recon}}^{PDM} = \overline{\boldsymbol{r}} + \sum_{i=1}^{4} b_i \boldsymbol{z}_i^{\text{PDM}}(t)$ with $b_i \in \mathbb{R}$ being the optimal approximation coefficients and $\overline{\boldsymbol{r}}$ the mean shape computed with the PDM. Since the PDM is discrete, we interpolate the landmarks with the uniform linear B-spline to obtain a continuous-domain representation.

TABLE III

RECONSTRUCTION ERROR $\|\boldsymbol{r}_{\text{data}} - \boldsymbol{r}_{\text{recon}}\|_{L_2}^2 / (\|\boldsymbol{r}_{\text{data}}\|_{L_2} \|\boldsymbol{r}_{\text{recon}}\|_{L_2})$ FOR CHROMOSOME DATA

Data	fPCA (4)	PDM (4,HR)
1	**0.016**	0.085
2	**0.004**	0.047
3	**0.004**	0.036
4	**0.009**	0.036
5	**0.013**	0.072
6	**0.007**	0.052
7	**0.001**	0.022
8	**0.005**	0.042
9	**0.002**	0.050
10	**0.002**	0.050
11	**0.002**	0.057
12	**0.016**	0.075
13	**0.012**	0.064
14	**0.014**	0.065
15	**0.005**	0.048
16	**0.012**	0.075
17	**0.015**	0.047
18	**0.006**	0.042
19	**0.003**	0.021
20	**0.002**	0.019
Mean	**0.008**	0.050
STD	0.005	0.018

This allows us to compute and compare L_2 reconstruction errors as reported in Table III. Again, the results suggest that the continuous-domain model (*i.e.*, fPCA) yields higher accuracy and captures shape variability more efficiently than the PDM. The reconstructed shapes are shown in Figure 6.

B. Shape Classification

If different groups of shapes are compared with each other, then the learned shape described in Section IV can be used as representative of each group. In a standard shape-classification setting, the mean shape $\boldsymbol{r}_{\text{mean}}$ can be viewed as a trained shape, where the curves used to compute this shape constitute the training set.

1) Classification in Medical Imaging: This experiment is part of a clinical study where the structural and potential functional changes of the pelvic-floor hiatus (PFH) are examined after a woman has given birth to one or several children [43]. 3D ultrasound volumes of 245 women were acquired and grouped into 61 *nulliparae* (women who did not give birth to children) and 184 *multiparae* (women who gave birth to one or several children). For both groups, images were acquired when the women were *at rest* or while *contracting* the PFH. The PFH is outlined on a specific 2D section of the ultrasound volume using the following procedure: A clinician draws key points on the image which have particular anatomical meaning. Curves are then computed by interpolating the ordered set of key points using spline interpolators [39], [40], as shown in Figure 7 (top row).

The qualitative analysis w.r.t. shape differences of different patient groups is important to clinicians. It reveals similarities (or differences) and, at the same time, removes within-group variability. In the present case, we constructed spline-based vector spaces using an affine transformation of the spline curves (*i.e.*, a vector space of dimension six). The mean shapes were computed for the four subgroups (*nulliparae* and *multiparae*, *at rest* or *contraction*). They are shown in the bottom row of Figure 7 and strongly indicate that the shape of the PFH probably does not change after giving birth although its size, perimeter, and surface do [44].

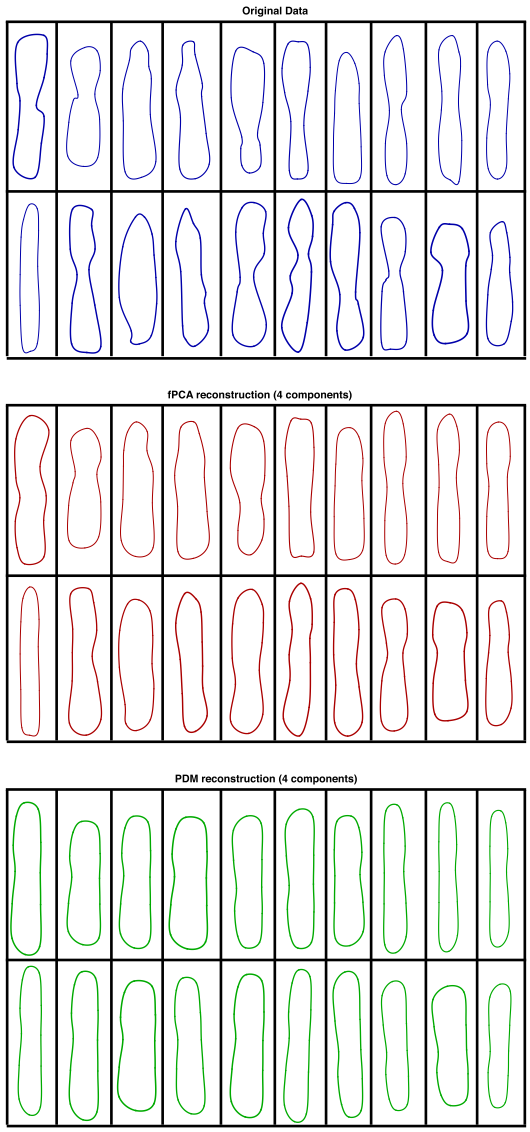

Fig. 6. Reconstructions of the chromosome data set. Top rows: original data. Middle rows: shapes that have been reconstructed using our proposed projection-based fPCA. Bottom rows: reconstruction using the PDM.

C. Sparse Dictionary Learning in Medical Imaging

We now want to construct a dictionary that encodes curves of several types. Our training set contains 150 outlines of brain structures, each representing one among the following five different types of shapes: *sagittal ventricle* (SV), *sagittal corpus callosum* (CC), *sagittal brain stem* (BS), *coronal ventricle* (CV), *axial ventricle* (AV). Samples of each brain structure are shown in Figure 8. The data set consists of thirty samples per brain structure and within each type, we have correspondence between landmarks.

However, the correspondence is no longer guaranteed between types. Furthermore, the types that represent CV and AV appear to be similar up to scaling and rotation (see Figure 8). Hence, the data set can also be considered

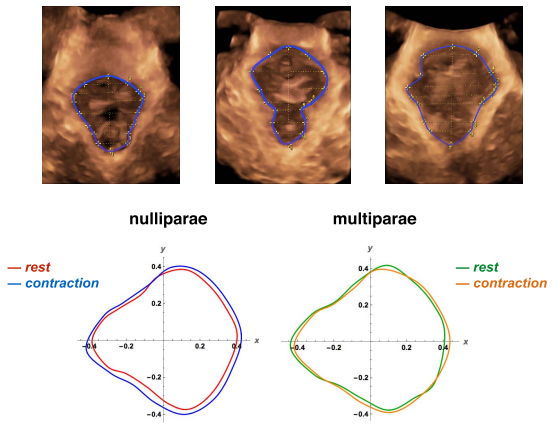

Fig. 7. Top row: 3D ultrasound volumetric data. The top-left image shows the PFH area of a patient at contraction, whereas the middle and right images show two different patients' PFH area at rest. The blue curves represent the outline of the PFH that has been constructed by spline interpolation of an ordered set of points drawn by a clinician on the image. Bottom row: The comparison of *nulliparae* vs. *multiparae* women reveals that there is no qualitative difference in the shapes $\boldsymbol{r}_{\text{mean}}^{\text{nulliparae}}$ and $\boldsymbol{r}_{\text{mean}}^{\text{multiparae}}$ between the two groups (although the sizes are different), independently from the state (*at rest* or *contraction*) Image courtesy Dr. med. Sylvain Meyer, Urogynaecology Unit and Obstetrics Department, CHUV Lausanne, EHC, Morges, Switzerland.

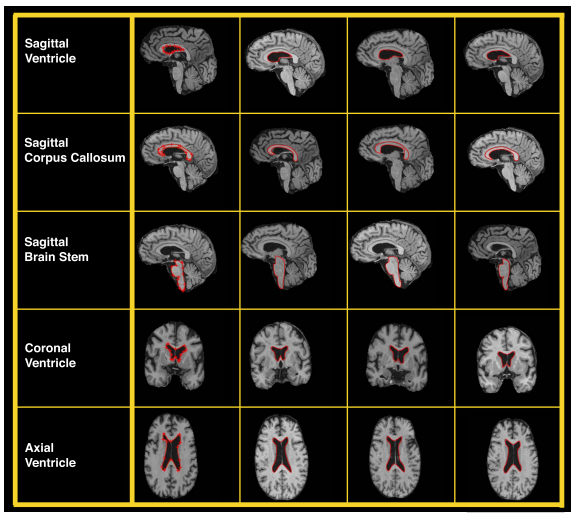

Fig. 8. Shape library of brain structures. Each row represents a shape type. To illustrate the within-group variability, four samples per type are shown.

as imbalanced besides being inhomogeneous. (It is well known that L_2-based methods are error-prone when dealing with imbalance or inhomogeneity in a data set. Meanwhile, sparse or ℓ_1-based methods tend to be more efficient in such cases.)

We have applied our method to learn a dictionary for sparse shape encoding. We computed two dictionaries, a first one with only 5 atoms (D5) and a second one with ten atoms (D10). We expect that each atom of D5 resembles one of the five shape types. The atoms of the two dictionaries are shown in Figure 9. The regularization parameter λ has been chosen empirically. As a control experiment, we also performed a (L_2-based) fPCA and used it to construct a dictionary that consists of the first ten eigencurves.

To validate our method regarding its ability to model unseen samples we have built a testing set that consists in twenty-five

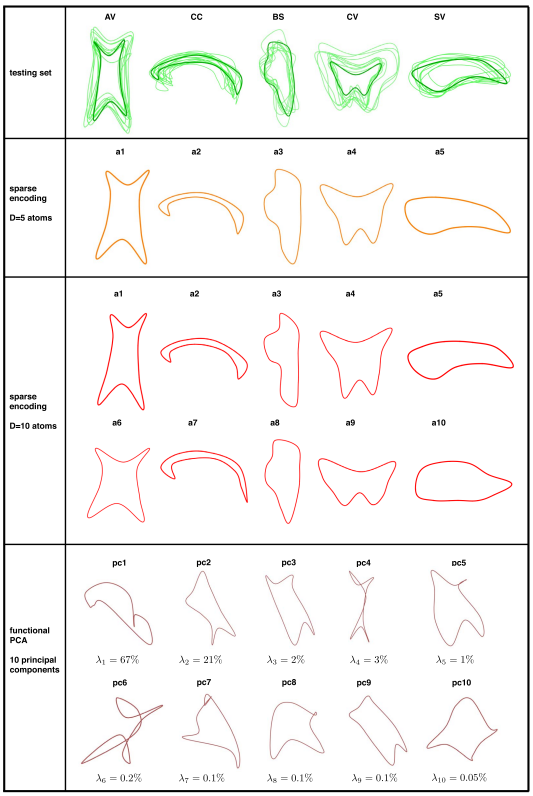

Fig. 9. Atoms of the learned shape dictionaries. Top row: samples of the training set. Second and third row: atoms a_i of dictionary D5 and D10. Bottom row: principal components (pc) obtained with the projection-based fPCA. The values of the λ_i are computed as described in the caption of Figure 5.

TABLE IV
CORRELATION BETWEEN THE BEST ATOM OF THE LEARNED DICTIONARY AND THE TESTING SET (ROUNDED VALUES)

		test 1	test 2	test 3	test 4	test 5	Mean	STD
SV	D5	0.99	**0.98**	**0.98**	**0.98**	**0.99**	**0.98**	0.00
	D10	**0.99**	0.97	0.97	0.97	0.98	0.98	0.01
	fPCA	0.85	0.86	0.88	0.89	0.87	0.87	0.01
CC	D5	**0.99**	0.99	0.98	0.99	**0.99**	0.99	0.00
	D10	0.99	**0.99**	**0.99**	**0.99**	0.98	**0.99**	0.00
	fPCA	0.80	0.78	0.84	0.81	0.76	0.80	0.02
BS	D5	0.99	**0.97**	**0.99**	**0.98**	**0.99**	**0.98**	0.01
	D10	**0.99**	0.96	0.99	0.98	0.99	0.98	0.01
	fPCA	0.90	0.89	0.91	0.92	0.90	0.90	0.01
CV	D5	0.98	0.98	**0.99**	**0.99**	0.99	0.99	0.00
	D10	**0.99**	**0.98**	0.99	0.99	**0.99**	**0.99**	0.00
	fPCA	0.94	0.97	0.95	0.95	0.95	0.95	0.01
AV	D5	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**	0.00
	D10	**0.99**	**0.99**	0.99	0.99	0.99	0.99	0.00
	fPCA	0.98	0.97	0.97	0.97	0.98	0.97	0.00

shapes which all differ from the shapes of the training set. Each group contains five samples (denoted as "test 1" to "test 5").

In a first step, we have computed the correlation between the most similar atom of the dictionary (for D5, D10, and fPCA) and every sample. The results are summarized in Table IV. It becomes apparent that the L_2-method fails when dealing with inhomogeneous data, as expected. The accuracy of the D5 and the D10 dictionary is similar. It is both qualitatively (Figure 9) and quantitatively high.

In a second step, we have reconstructed fifteen shapes from the testing set. They correspond to three different types:

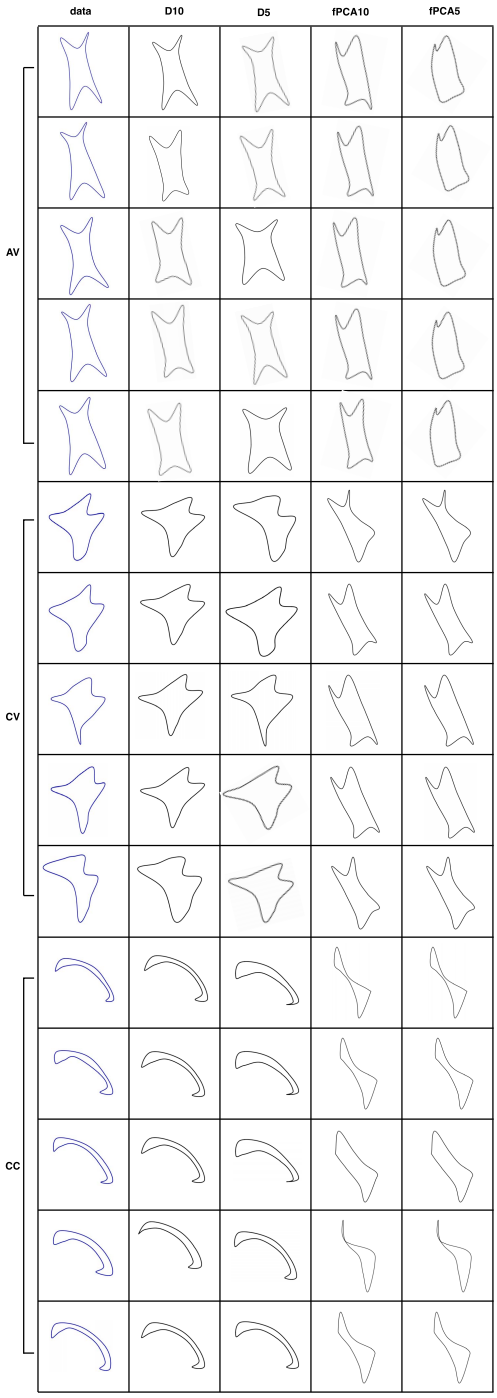

Fig. 10. Reconstructed testing set. The first column shows the testing set, whereas the second (D10) and third (D5) columns show the reconstruction with the sparse methods. The last two columns correspond to the reconstruction using fPCA.

five axial ventricles, five coronal ventricles, and five corpora callosa. We used the learned dictionaries and the corresponding sparse codes for the reconstruction process. We computed the normalized reconstruction error and compared it to the pure L_2 method (*i.e.*, the projection-based fPCA), where we used five as well as ten eigenshapes for the approximation. The reconstructed testing set is shown in Figure 10 and the errors are listed in Table V. We notice that the reconstruction with D10 tends to yield more accurate results than D5,

TABLE V

RECONSTRUCTION ERROR $\|\boldsymbol{r}_{\text{data}} - \boldsymbol{r}_{\text{recon}}\|^2_{L_2}/(\|\boldsymbol{r}_{\text{data}}\|_{L_2}\|\boldsymbol{r}_{\text{recon}}\|_{L_2})$

data	D10	D5	fPCA5	fPCA10
av1	**0.007**	0.063	0.223	0.777
av2	**0.013**	0.054	0.213743	0.65489
av3	0.0195	**0.004**	0.241	0.711
av4	**0.006**	0.127	0.196	0.558
av5	**0.008**	0.029	0.259	0.755
av mean	**0.010**	0.056	0.226	0.691
av std	0.005	0.046	0.024	0.087
cv1	**0.008**	0.065	0.218	0.218
cv2	0.010	**0.003**	0.205	0.205
cv3	0.022	**0.002**	0.214	0.214
cv4	**0.009**	0.011	0.191	0.191
cv5	**0.009**	0.057	0.447	0.447
cv mean	**0.011**	0.028	0.255	0.255
cv std	0.005	0.030	0.107	0.107
cc1	0.005	**0.003**	0.664	0.664
cc2	0.010	**0.004**	0.705	0.705
cc3	**0.009**	0.035	0.736	0.736
cc4	**0.012**	0.024	0.365	0.365
cc5	**0.008**	0.021	0.637	0.637
cc mean	**0.009**	0.018	0.621	0.621
cc std	0.002	0.013	0.148	0.148

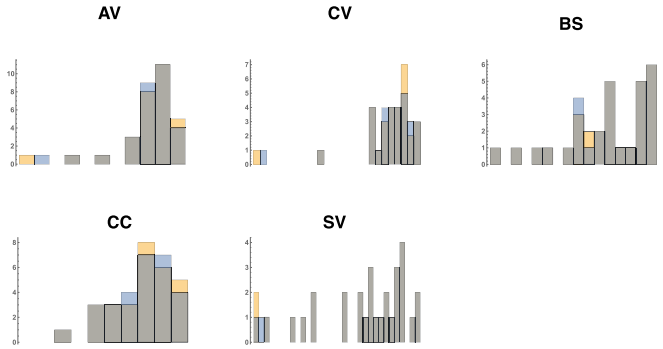

Fig. 11. Distributions (*within* each shape type) of the distances between the aligned data used to learn the shape dictionary 'D5' (see Figure 9) and the mean shape (yellow) or the best atom in 'D5' (blue). The grey area indicates the overlap between the two distributions. We see that for each type, the histograms differ by at most 3 out of 30 samples per type. The result suggests robustness of our algorithm w.r.t. different kinds of distributions found in the training data.

which is expected. Again, the projection-based fPCA fails to yield satisfying results. Furthermore, our solution indicates robustness w.r.t. the initial distributions of the training data: A comparison of the distances (*within* each type) between the mean shape and the aligned data as well as the distances between the best atom of 'D5' and the aligned data show that the corresponding histograms only differ by at most 3 out of 30 samples (Figure 11).

X. CONCLUSION

We have presented a unified framework for dictionary learning in the continuous domain, the data consisting of landmark-based parametric curves. We have provided closed-form solutions for the unbiased alignment of the training data and showed how shapes are learned for different types of applications such as the characterization of homogeneous, inhomogeneous, or imbalanced data. The alignment is based on a new method to compute mean shapes. It can also be used to construct shape priors in the context of segmentation problems. We have derived formulas for an exact and

fast implementation of the proposed framework using spline curves. Our examples and validation experiments highlight the advantages of our model compared to state-of-the-art discrete frameworks. Furthermore, our model can be easily extended and applied to 3D parametric curves that are defined by landmarks by noticing that the inner product between parametric spline surfaces can also be expressed as a matrix-vector multiplication.

APPENDIX

A. Derivation of Proposition 1

Proposition 1 follows from a standard result in functional analysis that states that the kernel of $\mathcal{P}$ is computed by

$$\begin{aligned}\mathcal{P}\boldsymbol{\phi}(t) &= \sum_{i=1}^{I} \boldsymbol{e}_i(t)\langle\tilde{\boldsymbol{e}}_i, \boldsymbol{\phi}\rangle = \langle\sum_{i=1}^{I} \boldsymbol{e}_i(t)\tilde{\boldsymbol{e}}_i^{\mathsf{T}}(\cdot), \boldsymbol{\phi}\rangle \\ &= \langle \boldsymbol{K}_{\mathcal{P}}(t,\cdot), \boldsymbol{\phi}\rangle \end{aligned} \tag{38}$$

and, therefore, $\boldsymbol{K}_{\mathcal{P}}(t,s) = \sum_{i=1}^{I} \boldsymbol{e}_i(t) \otimes \tilde{\boldsymbol{e}}_i(s)$.

B. Derivation of the Eigenequation (3)

We first notice that $\|\mathcal{P}\boldsymbol{r}\|_{L_2}^2 = \langle\mathcal{P}\boldsymbol{r}, \mathcal{P}\boldsymbol{r}\rangle = \langle\mathcal{P}^*\mathcal{P}\boldsymbol{r}, \boldsymbol{r}\rangle = \langle\mathcal{P}\boldsymbol{r}, \boldsymbol{r}\rangle$. Then, the eigenequation follows from a standard result in functional analysis. It can be shown that $\mathcal{P}$ is a *compact* operator since it is an orthogonal projector in a Hilbert space onto the finite dimensional subspace $\mathcal{S}$. Hence, the functional $\frac{\langle\mathcal{P}\boldsymbol{r},\boldsymbol{r}\rangle}{\langle\boldsymbol{r},\boldsymbol{r}\rangle}$ subject to $\langle\boldsymbol{r},\boldsymbol{r}\rangle = \|\boldsymbol{r}\|_{L_2}^2 = 1$ has a maximum. Under this constraint, we set $\max_{\boldsymbol{r}} \frac{\langle\mathcal{P}\boldsymbol{r},\boldsymbol{r}\rangle}{\langle\boldsymbol{r},\boldsymbol{r}\rangle} = \lambda \Leftrightarrow \langle\mathcal{P}\boldsymbol{r},\boldsymbol{r}\rangle = \lambda\langle\boldsymbol{r},\boldsymbol{r}\rangle = \langle\lambda\boldsymbol{r},\boldsymbol{r}\rangle$, which implies that λ is an eigenvalue of $\mathcal{P}$, *i.e.*, $\mathcal{P}\boldsymbol{r} = \lambda\boldsymbol{r}$. Furthermore, the unit-norm condition on the eigencurves can be generalized as $\langle\boldsymbol{r}_p, \boldsymbol{r}_q\rangle = \delta_{p-q}$, which is based on the fact that eigenfunctions corresponding to different eigenvalues are orthogonal.

Applying the same derivation to all the projectors $\mathcal{P}_k$ yields the stated eigenequation.

C. Proof of Theorem 1

Proof: The eigenequation (3) is developed as

$$\begin{aligned}\sum_k \mathcal{P}_k\boldsymbol{\phi}(t) &= \langle\sum_{k=1}^{K} \boldsymbol{K}_{\mathcal{P}_k}(t,\cdot), \boldsymbol{\phi}\rangle \\ &= \langle\sum_{k=1}^{K}\sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)} \otimes \tilde{\boldsymbol{e}}_i^{(k)}(\cdot), \boldsymbol{\phi}\rangle = \lambda\boldsymbol{\phi}(t).\end{aligned}$$

We identify

$$\begin{aligned}\boldsymbol{\phi}(t) &= \frac{1}{\lambda}\langle\sum_{k=1}^{K}\sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)} \otimes \tilde{\boldsymbol{e}}_i^{(k)}(\cdot), \boldsymbol{\phi}\rangle \\ &= \frac{1}{\lambda}\sum_{k=1}^{K}\sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)}\langle\tilde{\boldsymbol{e}}_i^{(k)}, \boldsymbol{\phi}\rangle = \sum_{k=1}^{K}\sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)}\gamma_{ik},\end{aligned}$$

where $\gamma_{ik} = \frac{\langle\tilde{\boldsymbol{e}}_i^{(k)},\boldsymbol{\phi}\rangle}{\lambda}$. Hence,

$$\lambda\gamma_{ik} = \langle\tilde{\boldsymbol{e}}_i^{(k)}, \sum_{l=1}^{K}\sum_{j=1}^{I} \boldsymbol{e}_j(\cdot)^{(l)}\gamma_{jl}\rangle = \sum_{l=1}^{K}\sum_{j=1}^{I} \gamma_{jl}\langle\tilde{\boldsymbol{e}}_i^{(k)}, \boldsymbol{e}_j^{(l)}\rangle. \tag{39}$$

We define the $(K \cdot I) \times (K \cdot I)$ matrix

$$[\Gamma]_{(k-1)\cdot I+i,(j-1)\cdot K+l} = \langle\tilde{\boldsymbol{e}}_i^{(k)}, \boldsymbol{e}_j^{(l)}\rangle, \tag{40}$$

where $k, l \in [1, \ldots, K]$ and $i, j \in [1 \ldots I]$. Now, we collect all the γ_{ik} in one large vector $\boldsymbol{\gamma}$, to establish relation (39) as eigenvalue problem $\Gamma\boldsymbol{\gamma} = \lambda\boldsymbol{\gamma}$ and, hence, we can compute $\boldsymbol{\phi}(t) = \sum_{k=1}^{K}\sum_{i=1}^{I} \boldsymbol{e}_i(t)^{(k)}\gamma_{ik}$. ■

D. Vector Space Including a Translation

If in the construction of the K projectors a basis that includes the translation $\boldsymbol{b}$ given by $\{\boldsymbol{e}_{b_x}, \boldsymbol{e}_{b_y}\} = \{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\}$ is used, then both $\boldsymbol{e}_{b_x}$ and $\boldsymbol{e}_{b_y}$ are eigencurves and, hence, solutions of the eigenequation (3) with eigenvalue equal to K. This is easy to see since, for such a projector, $\mathcal{P}\boldsymbol{e}_{b_x} = 1 \cdot \boldsymbol{e}_{b_x}$ and, therefore, $\sum_{k=1}^{K} \mathcal{P}_k\boldsymbol{e}_{b_x} = K \cdot \boldsymbol{e}_{b_x}$. The same result holds true for $\boldsymbol{e}_{by}$. In this case, $\boldsymbol{r}_{\text{mean}}$ is chosen to be the third eigencurve, since the first two are constants (*i.e.*, 2D points).

E. Proof of Theorem 3

Using (14), we develop

$$\begin{aligned}&\Big(\sum_{k=1}^{K}\langle\mathcal{P}_k\boldsymbol{r},\boldsymbol{r}\rangle = \lambda\langle\boldsymbol{r},\boldsymbol{r}\rangle\Big) \Leftrightarrow \Big(\sum_{k=1}^{K}\mathbf{c}^{\mathsf{T}}\mathbf{P}_k^{\mathsf{T}}\boldsymbol{\Psi}\boldsymbol{c} = \lambda\mathbf{c}^{\mathsf{T}}\boldsymbol{\Psi}\boldsymbol{c}\Big) \\ &\Leftrightarrow \Big(\sum_{k=1}^{K}\mathbf{P}_k^{\mathsf{T}}\boldsymbol{\Psi}\boldsymbol{c} = \lambda\boldsymbol{\Psi}\boldsymbol{c}\Big) \Leftrightarrow \Big(\boldsymbol{\Psi}^{-1}\sum_{k=1}^{K}\mathbf{P}_k^{\mathsf{T}}\boldsymbol{\Psi}\boldsymbol{c} = \lambda\mathbf{c}\Big),\end{aligned} \tag{41}$$

where $\mathbf{c}$ is the vector of control points of $\boldsymbol{r}$. Maximizing (41) w.r.t. $\mathbf{c}$ and using the expression provided by Theorem 2 for the spline projector, (41) boils down to the eigenvalue problem $\sum_{k=1}^{K}\mathbf{C}_k(\mathbf{C}_k\boldsymbol{\Psi}\mathbf{C}_k^{\mathsf{T}})^{-1}\mathbf{C}_k^{\mathsf{T}}\boldsymbol{\Psi}\mathbf{c} = \lambda\mathbf{c}$. ■

REFERENCES

[1] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756008

[2] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2009, pp. 689–696. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553463

[3] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2004.

[4] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 1561–1568.

[5] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, vol. 10, no. 3, pp. 515–534, Jul. 2009.

[6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[7] S. Zhang, Y. Zhan, and D. N. Metaxas, "Deformable segmentation via sparse representation and dictionary learning," *Med. Image Anal.*, vol. 16, no. 7, pp. 1385–1396, 2012.

[8] S. Zhang, Y. Zhan, Y. Zhou, M. Uzunbas, and D. N. Metaxas, "Shape prior modeling using sparse representation and online dictionary learning," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2012, pp. 435–442.

[9] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[10] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.
[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1987.
[12] A. Blake and M. Isard, *Active Contours: The Application of Techniques From Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*, 1st ed. New York, NY, USA: Springer-Verlag, 1998.
[13] X. Bresson, S. Esedoḡlu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *J. Math. Imag. Vis.*, vol. 28, no. 2, pp. 151–167, 2007.
[14] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. Hoboken, NJ, USA: Wiley, 1998.
[15] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 53, no. 2, pp. 285–339, 1991.
[16] F. J. Rohlf, "Bias and error in estimates of mean shape in geometric morphometrics," *J. Hum. Evol.*, vol. 44, no. 6, pp. 665–683, 2003.
[17] D. Rueckert, A. F. Frangi, and J. A. Schnabel, "Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1014–1025, Aug. 2003.
[18] M. Styner and G. Gerig, "Medial models incorporating object variability for 3D shape analysis," in *Information Processing in Medical Imaging*. Berlin, Germany: Springer, 2001, pp. 502–516.
[19] A. Kelemen, G. Székely, and G. Gerig, "Elastic model-based segmentation of 3-D neuroradiological data sets," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 828–839, Oct. 1999.
[20] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Hilton Head, SC, USA, Jun. 2002, pp. 316–323.
[21] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 295–313, 2002.
[22] M. Gastaud, M. Barlaud, and G. Aubert, "Combining shape prior and statistical features for active contour segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 726–734, May 2004.
[23] M. A. T. Figueiredo, J. M. N. Leito, and A. K. Jain, "Unsupervised contour representation and estimation using B-splines and a minimum description length criterion," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1075–1087, Jun. 2000.
[24] P. Brigger, J. Hoeg, and M. Unser, "B-spline snakes: A flexible tool for parametric contour detection," *IEEE Trans. Signal Process.*, vol. 9, no. 9, pp. 1484–1496, Sep. 2000.
[25] F. Precioso and M. Barlaud, "B-spline active contours for fast video segmentation," in *Proc. Int. Conf. Image Process.*, Thessaloniki, Greece, Oct. 2001, pp. 777–780.
[26] F. Precioso and M. Barlaud, "B-spline active contour with handling of topology changes for fast video segmentation," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 6, p. 793184, Jan. 2002.
[27] M. Unser, B. L. Trus, and A. C. Steven, "Normalization procedures and factorial representations for classification of correlation-aligned images: A comparative study," *Ultramicroscopy*, vol. 30, no. 3, pp. 299–310, Jul./Aug. 1989.
[28] C. Vonesch, F. Stauber, and M. Unser, "Steerable PCA for rotation-invariant image recognition," *SIAM J. Imag. Sci.*, vol. 8, no. 3, pp. 1857–1873, 2015.
[29] M. Unser, "Sampling—50 years after Shannon," *Proc. IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
[30] A. Badoual, D. Schmitter, and M. Unser, "An inner-product calculus for periodic functions and curves," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 878–882, Jun. 2016.
[31] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, vol. 19, P. B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. Cambridge, MA, USA: MIT Press, 2007, pp. 801–808.
[32] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[34] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, Jan. 2001.
[35] K. Engan, S. O. Aase, and J. H. Husoy, "Frame based signal compression using method of optimal directions (MOD)," in *Proc. IEEE ISCAS*, vol. 4. May/Jun. 1999, pp. 1–4.
[36] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[37] L. Bottou, "On-line learning and stochastic approximations," in *On-Line Learning in Neural Networks*, D. Saad, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1998, pp. 9–42.
[38] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems*, vol. 20, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2008, pp. 161–168.
[39] D. Schmitter, R. Delgado-Gonzalo, and M. Unser, "Trigonometric interpolation kernel to construct deformable shapes for user-interactive applications," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 2097–2101, Nov. 2015.
[40] D. Schmitter, R. Delgado-Gonzalo, and M. Unser, "A family of smooth and interpolatory basis functions for parametric curve and surface representation," *Appl. Math. Comput.*, vol. 272, no. 1, pp. 53–63, Jan. 2016.
[41] R. Delgado-Gonzalo, D. Schmitter, V. Uhlmann, and M. Unser, "Efficient shape priors for spline-based snakes," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3915–3926, Nov. 2015.
[42] D. Schmitter and M. Unser, "Similarity-based shape priors for 2D spline snakes," in *Proc. 12th IEEE Int. Symp. Biomed. Imag., Nano Macro (ISBI)*, Brooklyn, NY, USA, Apr. 2015, pp. 1216–1219.
[43] D. Baud, S. Meyer, Y. Vial, P. Hohlfeld, and C. Achtari, "Pelvic floor dysfunction 6 years post-anal sphincter tear at the time of vaginal delivery," *Int. Urogynecol. J.*, vol. 22, no. 9, pp. 1127–1134, 2011.
[44] G. A. van Veelen, K. J. Schweitzer, and C. H. van der Vaart, "Ultrasound imaging of the pelvic floor: Changes in anatomy during and after first pregnancy," *Ultrasound Obstetrics Gynecol.*, vol. 44, no. 4, pp. 476–480, 2014.
[45] K. V. Mardia, "Measures of multivariate skewness and kurtosis with applications," *Biometrika*, vol. 57, no. 3, pp. 519–530, 1970.
[46] M. J. Corless and A. Frazho, *Linear Systems and Control: An Operator Perspective*. New York, NY, USA: Taylor & Francis, 2003.

Daniel Schmitter received the master's degree in bioengineering and biomedical technologies from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2013, where he is currently pursuing the Ph.D. degree in electrical engineering with the Biomedical Imaging Group. He was with the Advanced Clinical Imaging Technology Group, Siemens, at the Center for Biomedical Imaging, Switzerland, where he was one of the main contributors involved on brain-imaging software and related shape-characterization algorithms. He is currently with on spline-based shape representation and shape modeling. He is the inventor of augmented CAD and the founder of Mirrakoi, a spin-off company from EPFL that distributes the augmented CAD technology.

Michael Unser is currently Professor and Director with Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, the use of splines for image processing, and stochastic processes. He has published over 250 journal papers on those topics. From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, USA, conducting research on bioimaging. He was recently elected to the board of the Proceedings of the IEEE. He co-organized the first IEEE International Symposium on Biomedical Imaging (ISBI2002) and was the Founding Chair of the technical committee of the IEEE-SP Society on Bio Imaging and Signal Processing. He is a Fellow of the IEEE in 1999, EURASIP fellow in 2009, and a member of the Swiss Academy of Engineering Sciences. He is the recipient of several international prizes including three IEEE-SPS Best Paper Awards and two Technical Achievement Awards from the IEEE 2008 SPS and EMBS 2010. He has held various editorial positions in the journals of the IEEE SIGNAL PROCESSING SOCIETY, the IEEE TRANSACTION MEDICAL IMAGING, and THE IEEE TRANSACTION IMAGE PROCESSING.