

A Functional Approach to Curve Alignment and Shape Analysis

Issam-Ali Moindjié^{*1,2}, Cédric Beaulac¹, and Marie-Hélène Descary¹

¹Department of mathematics, Université du Québec à Montréal, Canada

²LAMPS, Université de Perpignan Via Domitia, France

^{*}Corresponding author: issam-ali.moindjie@univ-perp.fr

Abstract

In many image analysis problems, the contours of objects carry important statistical information about shape. Such contours are typically affected by deformation variables including scaling, translation, rotation, and reparametrization. Previous studies in statistical shape analysis have mainly focused on analyzing contours and shapes through discrete observations. While this approach might offer computational advantages, it overlooks the continuous nature of these objects and their underlying geometric structure. It also ignores potential dependencies between the deformation variables and their effect on the shape, which may result in a loss of statistical information and reduced interpretability. In this paper, we introduce a novel framework for analyzing shapes within the context of Functional Data Analysis (FDA). Basis expansion techniques are employed to derive analytic solutions for the estimation of deformation variables, namely scaling, translation, rotation, and reparametrization, thereby achieving curve alignment. A generative model for random contours is then developed using principal component analysis techniques. Numerical experiments on simulated data and the *MPEG-7* database demonstrate that our method successfully identifies deformation parameters and captures the underlying distribution of random contours in settings where traditional FDA methods fail.

1 Introduction

The analysis of images has become increasingly important with advances in acquisition and storage technologies. While representing images as pixel matrices has enabled powerful learning methods, this representation does not explicitly capture the fundamental components of images: the objects they depict, characterized by their shapes, colors, and textures. In this work, we focus on the statistical analysis of object shapes.

If we are interested in analyzing shapes in images, a natural representation is through binary images, as illustrated in Figure 1(b). These binary images, often referred to as masks or silhouettes, typically result from image segmentation and consist of pixels taking two values:

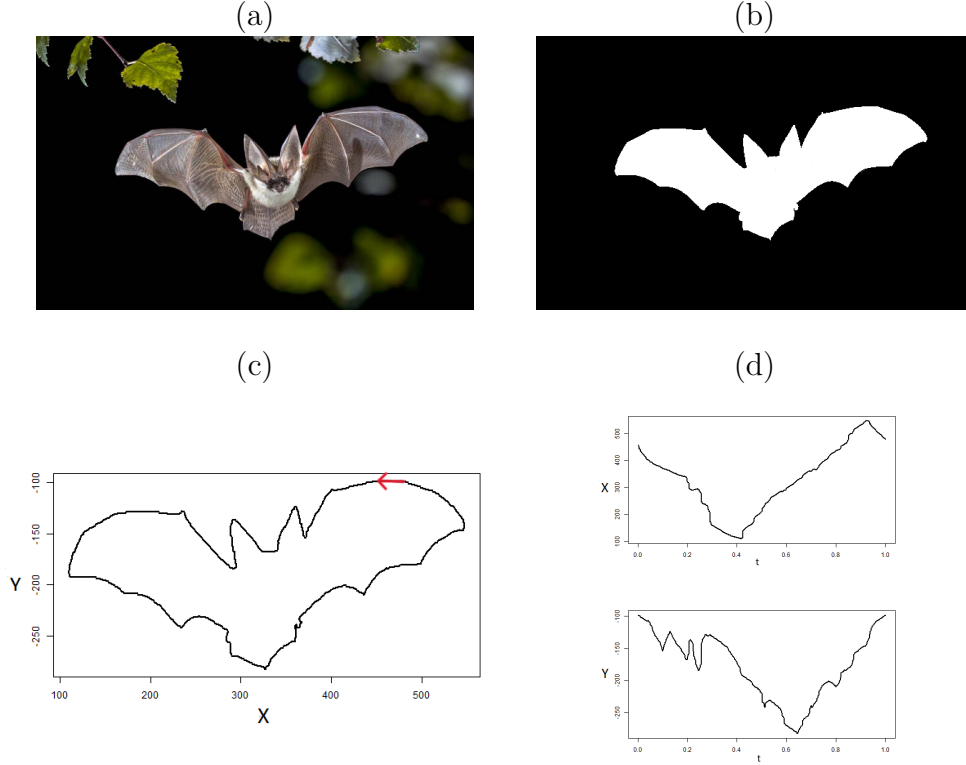


Figure 1: From an image to the planar curve $\mathbf{C}$: (a) original image (nl, 2020), (b) binarized image, (c) contour of the object in the image, and (d) the coordinate functions $X(t)$ and $Y(t)$. The red arrow in (c) indicates the starting point and the orientation of the traveling used to obtain the coordinate functions in (d).

one indicating regions outside the object of interest and the other indicating regions inside, thereby representing shapes as regions rather than explicit boundaries (or contours). Srivastava and Klassen (2016) identify other representations of shape, such as point clouds and ordered samples of points, and argue in favor of a functional representation; see Srivastava and Klassen (2016) for a detailed discussion. We also adopt a curve-based representation of shapes in images. Compared to binary images, the curve representation, which gives the contour of the silhouette image, leads to a substantial reduction in dimensionality while preserving the geometric structure. Moreover, if contours are extracted as collections of pixels, images of different resolutions yield contours with varying numbers of points. These issues can be addressed by smoothing the collection of points and projecting the resulting contours onto a common functional basis, thereby enabling a coherent statistical analysis of shapes.

We focus on the variable $\mathbf{C}$, which represents the contour of the main object in the image. This variable can be viewed as a random parametric planar curve:

$$\mathbf{C}(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix},$$

where $t \in [0, 1]$ represents the proportion of the curve that has been traversed, from the starting point ($t = 0$) to the end point ($t = 1$), and X, Y are the coordinate functions. We

restrict our attention to the case where the contour is closed, that is, $\mathbf{C}(0) = \mathbf{C}(1)$. Figure 1 illustrates the pipeline for extracting the planar curve $\mathbf{C}$ from an image.

The contour $\mathbf{C}$ is assumed to be a deformed version of a latent variable $\tilde{\mathbf{C}}$:

$$\mathbf{C}(t) = \rho \mathbf{O} \tilde{\mathbf{C}} \circ \gamma(t) + \mathbf{T}, \quad t \in [0, 1], \quad (1)$$

where $(\rho, \mathbf{O}, \mathbf{T}, \gamma)$ are deformation variables and $\tilde{\mathbf{C}}$ is the *shape* of $\mathbf{C}$ in the sense of Kendall (1989), i.e., what remains when all deformations are removed. The deformation variables act through scaling ($\rho \in \mathbb{R}^+$), translation ($\mathbf{T} \in \mathbb{R}^2$), rotation ($\mathbf{O} \in SO(2)$), and reparametrization ($\gamma \in \Gamma$, where Γ denotes a set of increasing functions from $[0, 1]$ to $[0, 1]$).

The analysis of $\mathbf{C}$ has been widely studied in the shape analysis literature. Early approaches relied on a finite set of points along the curve, referred to as landmarks, which are assumed to correspond across objects (Dryden and Mardia, 1998). More recent work considers $\mathbf{C}$ as a continuous planar curve (see, e.g., Younes (1998)) and introduces elastic shape analysis frameworks in which reparametrization functions γ are modeled as diffeomorphic transformations and estimated jointly with rotation. In practice, these approaches rely on discretizations of the curve and numerical procedures, such as dynamic programming, and do not admit closed-form solutions. Moreover, elastic reparametrizations introduce an additional layer of transformation that must be estimated and interpreted.

This work addresses the specific challenges arising when contours are automatically extracted from images, where phase variability due to arbitrary starting points is the dominant source of misalignment. In this context, we restrict the class of admissible transformations by focusing on global deformations, namely scaling, translation, rotation, and a simple reparametrization of the starting point. This choice leads to alignment procedures that are fully compatible with a functional representation of the curve and can be carried out without additional discretization beyond the chosen basis representation. While more flexible approaches may be incorporated at a later stage, they are not required to capture the main geometric variability induced by image-based contour extraction.

Functional Data Analysis (FDA) provides a natural framework for modeling planar curves as functional observations Ramsay and Silverman (2005). While several works have considered shape data within this framework (see, e.g., Stöcker et al. (2023); Dai and Müller (2018); Dai (2022)), the problem of jointly estimating deformation variables and modeling the resulting aligned curves has not been explicitly addressed.

In this paper, we adopt a functional data analysis framework to study the random planar curve defined in (1). In particular, $\mathbf{C}$ is viewed as a bivariate functional variable and its realizations are therefore treated as functional data.

In summary, the main contributions of this work are as follows:

- A functional data-based procedure to estimate the deformation variables from a bivariate random planar curve $\mathbf{C}$, focusing on phase variability induced by different starting points.
- A novel model for $\mathbf{C}$ that explicitly accounts for the deformation variables through two separate principal component analysis (PCA): one is performed on the aligned functional variable $(\rho \tilde{\mathbf{C}})$ and the other on the remaining deformation variables.

The remainder of the paper is organized as follows. Section 2 focuses on estimating the deformation variables, while Section 3 introduces a generative model for $\mathbf{C}$. Numerical experiments on simulated data are presented in Section 4, followed by an application to the *MPEG-7* database in Section 5. The paper concludes with a discussion in Section 6.

2 Alignment of planar closed curves

Before introducing our alignment procedure, we first illustrate why alignment is a necessary step for the statistical analysis of planar curves.

For illustration purposes, Figure 2 depicts realizations of $\mathbf{C}$ and their sample mean. The left column displays planar curves, while the corresponding coordinate functions are shown in the middle and right columns. The first row presents a sample of $n = 10$ "heart-shaped" planar curves exhibiting variations in rotation and reparametrization, while the second row displays the estimate of the mean function $\mathbb{E}(\mathbf{C})$, defined coordinate-wise as:

$$\mathbb{E}(\mathbf{C}(t)) = \begin{pmatrix} \mathbb{E}(X(t)) \\ \mathbb{E}(Y(t)) \end{pmatrix}, \quad t \in [0, 1].$$

As illustrated in Figure 2, this definition does not preserve the geometric structure of the shapes, yielding an average that is not representative of the underlying objects. This issue arises from the presence of deformation variables, such as rotation and reparametrization, which introduce non-linear variability that cannot be captured by linear functional methods.

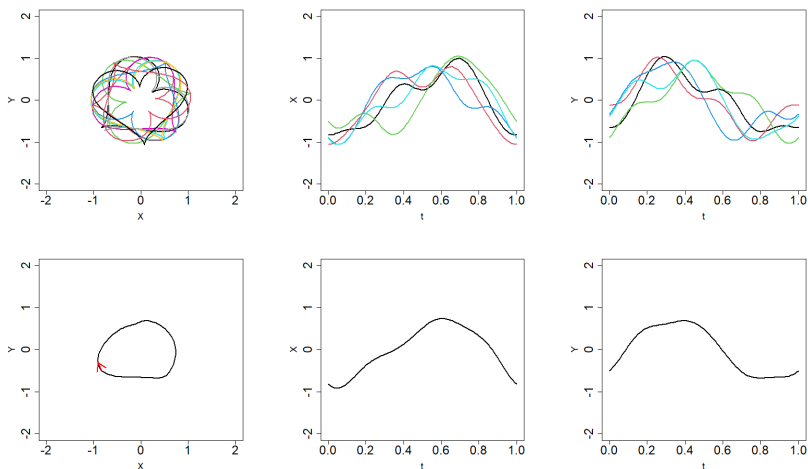


Figure 2: Examples of observations of $\mathbf{C}$ (first row) and their sample mean (second row). Columns present the planar curves, the first and the second coordinate functions, respectively.

A similar problem occurs when alignment is not addressed for a univariate functional variable X taking values in $L^2([0, 1], \mathbb{R})$. In this case, X can be expressed as $X = \tilde{X} \circ \gamma$, where γ is a warping function and $\tilde{X}$ represents the curve without phase variation. Hence, obtaining the shape $\tilde{\mathbf{C}}$ from $\mathbf{C}$ can be viewed as a generalization of the alignment procedure

for univariate functional variables to the bivariate case. In the univariate case, this problem has been the source of many developments in FDA (see Chap.7 in Ramsay and Silverman (2005) and Marron et al. (2015) for a review). For example, in Tucker et al. (2013) a flexible generative model of X is proposed. This method relies on a joint functional principal component analysis of bijective transformations of $\tilde{X}$ and γ , inspired by the square-root velocity framework. Using a transformation of $\gamma \in \Gamma$ is mandatory due to the complex structure of Γ , defined in this case as a space of diffeomorphism functions. Happ et al. (2019) discuss several choices of transformations, comparing their relative advantages and drawbacks. They found that square-root velocity inverse transformations might lead to warping functions with atypical structures, making them hard to interpret.

To solve this issue, in this manuscript we consider the following space of reparametrization functions

$$\Gamma = \{\gamma_\delta : [0, 1] \rightarrow [0, 1], \gamma_\delta(t) = \text{mod}(t - \delta, 1), \delta \in [0, 1]\}, \quad (2)$$

where $\text{mod}(\cdot, 1)$ is the modulo 1 function. It is thus assumed that each reparametrization function is uniquely characterized by its "starting point" $\delta \in [0, 1]$. The motivation for choosing this particular space is discussed in Section 2.2.

2.1 Model and parametrization

We recall the model introduced in (1):

$$\mathbf{C}(t) = \rho \mathbf{O} \tilde{\mathbf{C}} \circ \gamma(t) + \mathbf{T}, \quad t \in [0, 1],$$

where $\mathbf{C} = (X, Y)^\top$ is a bivariate functional variable taking values in the space of square-integrable functions $L^2([0, 1], \mathbb{R}^2)$. Since $L^2([0, 1], \mathbb{R}^2)$ is isomorphic to $L^2([0, 1], \mathbb{R}) \times L^2([0, 1], \mathbb{R})$, we equivalently view $\mathbf{C}$ as an element of the Hilbert space $\mathcal{H} = L^2([0, 1], \mathbb{R}) \times L^2([0, 1], \mathbb{R})$. Moreover, since $\mathbf{C}$ represents a closed curve, we assume that $\mathbf{C}(0) = \mathbf{C}(1)$. The space $\mathcal{H}$ is equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ defined as:

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{H}} = \int_0^1 \{f_1(t)g_1(t) + f_2(t)g_2(t)\} dt, \quad \mathbf{f} = (f_1, f_2)^\top, \mathbf{g} = (g_1, g_2)^\top \in \mathcal{H}.$$

The first step in analyzing a dataset of planar closed curves is to align the data, meaning to extract the shape $\tilde{\mathbf{C}}$ from $\mathbf{C}$ or, equivalently, to identify the deformation variables in model (1). We formulate the following hypothesis regarding these deformation variables:

- ρ is a positive random scalar variable;
- $\mathbf{T} = (T_1, T_2)^\top$ is a random vector in $\mathbb{R}^2$;
- The rotation matrix

$$\mathbf{O} := \mathbf{O}_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (3)$$

depends on a random angle θ in $[0, 2\pi]$. Note that the case of the reflection matrix $\mathbf{R}_\theta$ can be considered using the propriety:

$$\mathbf{R}_\theta = \mathbf{O}_\theta \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

This reduces to analyzing $(X, -Y)^\top$ in our framework instead of $\mathbf{C} = (X, Y)^\top$.

- The reparametrization function $\gamma := \gamma_\delta$ takes values in Γ , where Γ is defined as in (2).
- The random function $\tilde{\mathbf{C}} = (\tilde{X}, \tilde{Y})^\top$ takes values in $\mathbf{S}^\infty = \{\mathbf{f} \in \mathcal{H}, \|\mathbf{f}\|_{\mathcal{H}} = 1\}$, where $\|\cdot\|_{\mathcal{H}}$ is the norm induced by the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. For identification purposes, we will also assume that $\tilde{\mathbf{C}}$ is centered, i.e.

$$\int_0^1 \tilde{X}(t)dt = \int_0^1 \tilde{Y}(t)dt = 0.$$

Before presenting our alignment method, we first provide the intuition behind the definition of the function space Γ .

2.2 The space of reparametrization functions Γ

Recall that we define the space of reparametrization functions Γ in (2) as

$$\Gamma = \{\gamma_\delta : [0, 1] \rightarrow [0, 1], \gamma_\delta(t) = \text{mod}(t - \delta, 1), \delta \in [0, 1]\},$$

with $\text{mod}(x, 1) = x - \lfloor x \rfloor, \forall x \in \mathbb{R}$. Functions that belong to this space can be seen as a generalization of the well-known time-shift deformation warping functions for univariate functional data (see e.g. Marron et al. (2015) for details) to the case of planar closed curves. The modulo function allows considering the cyclic nature of the closed curve $\mathbf{C}$.

Figure 3 exhibits the effect of $\gamma_\delta \in \Gamma$ for different values of δ on a realization $\tilde{\mathbf{c}} = (\tilde{x}, \tilde{y})^\top$ of $\tilde{\mathbf{C}}$. We see the "rearrangement effect" of the reparametrization function: $\delta = 0$ defines the reference shape and the starting point of the coordinate functions of $\tilde{\mathbf{c}} \circ \gamma_\delta$ coincides with the point $t = \delta$ of the reference shape.

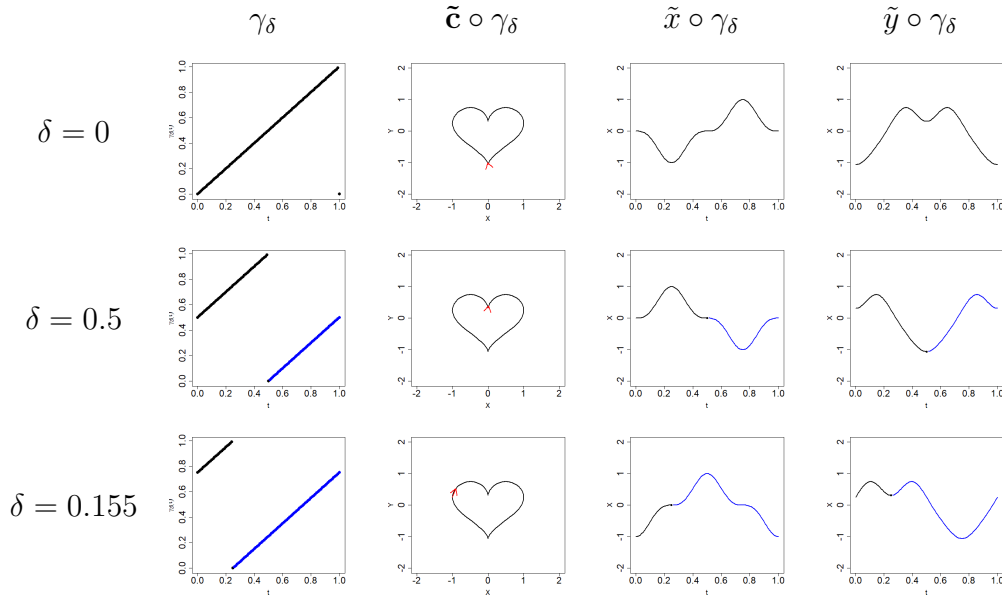


Figure 3: Plots of the functions γ_δ and of its composition with the shape $\tilde{\mathbf{c}}$ for different values of δ , resulting in different starting points (represented by the red arrows) and coordinate functions.

In contour datasets, the variation of δ arises from edge detection algorithms, which select the starting point as the argument of the coordinate functions associated with the smallest/largest distance from the origin. The issue with this approach is that the starting value depends on the transformation variables, particularly rotation. Figure 4 illustrates this effect when the smallest distance from the origin is considered: each row shows the same shape subjected to different rotations, leading to entirely different coordinate functions. Therefore, the challenge is to obtain δ jointly with the other transformation variables. We will elaborate on this point in Section 2.3.1.

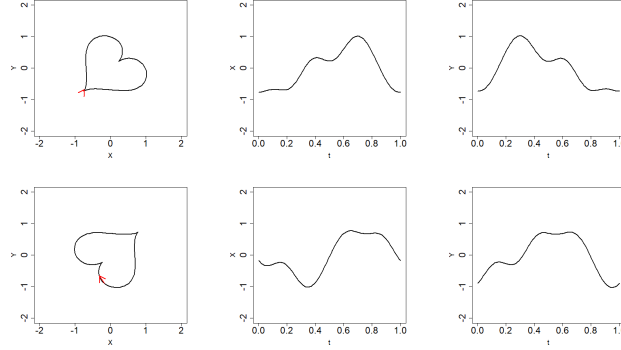


Figure 4: Illustration of the effect of rotation on a given shape and its coordinate functions when the starting value (illustrated by the red arrow on the shape) is defined as the point associated with the lowest values of the coordinate functions.

In addition to being well-suited for contour data, the space Γ has interesting mathematical properties, as demonstrated by the following result.

Proposition 2.1. *$(\Gamma, \circ)$ is a group and the isometry property $\|\mathbf{f} \circ \gamma_\delta\|_{\mathcal{H}} = \|\mathbf{f}\|_{\mathcal{H}}$ holds for $\mathbf{f} \in \mathcal{H}$ and $\gamma_\delta \in \Gamma$.*

While the class of reparametrization functions Γ defined in (2) focuses on variability induced by different starting points, it can be extended to incorporate more flexible transformations. In particular, one may consider reparametrizations of the form $\gamma \in \Gamma_0$, where Γ_0 is the space of diffeomorphism functions from $[0, 1] \rightarrow [0, 1]$, leading to elastic alignment frameworks similar to those studied in the literature (see, e.g., Tucker et al. (2013)). This can be done by modifying Γ as follows:

$$\tilde{\Gamma} = \{\tilde{\gamma}_{\gamma, \delta} : [0, 1] \rightarrow [0, 1], \tilde{\gamma}_{\gamma, \delta}(t) = \text{mod}(\gamma(t) - \delta, 1), \delta \in [0, 1] \text{ and } \gamma \in \Gamma_0\}.$$

However, such extensions introduce additional complexity both in modeling and estimation, and are not required to capture the dominant source of phase variability arising in contour data extracted from images. For this reason, we restrict our attention to the simpler class Γ , which provides an interpretable and tractable framework consistent with the data generation process. This perspective allows us to bridge practical contour extraction procedures with a principled statistical framework.

With this in mind, the rest of this section presents the process of determining the deformation variables $(\rho, \mathbf{O}_\theta, \mathbf{T}, \gamma_\delta)$.

2.3 Estimation of the deformation variables

Although our ultimate goal is to analyze a sample of planar closed curves—that is, to work with n realizations of the random function $\mathbf{C}$ —the alignment procedure is carried out on a curve-by-curve basis. Specifically, for each curve, the translation and scaling variables can be directly determined based on the assumptions made on the shape $\tilde{\mathbf{C}}$, as will be discussed in Section 2.3.2. As we will see in Section 2.3.3, the rotation and reparametrization variables are defined relative to a template, which may depend on other curves in the sample. However, once this template has been specified, the alignment procedure can be applied independently to each curve. Before presenting the estimation procedure, we first introduce the assumed basis representation of $\mathbf{C}$.

2.3.1 Representation of $\mathbf{C}$ using a basis expansion

Due to technological limitations, functional variables are observed on discrete grids in practice. In our setting, this discretization is induced by the image resolution. A crucial first step in functional data analysis is therefore smoothing, which consists of reconstructing a continuous representation of each discretely observed curve. Working with such a continuous representation, rather than directly with pixel-level data, provides several advantages. In particular, it allows for the analysis of images with different resolutions and yields a representation of the shape that is significantly more parsimonious than the original binary image.

A common approach to smoothing is to use a basis function expansion (Ramsay and Silverman, 2005, Chap. 3), i.e., to assume that $\mathbf{C}$ can be expressed as

$$\mathbf{C}(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} = \sum_{k=0}^M \begin{pmatrix} a_{k1} \\ a_{k2} \end{pmatrix} \psi_k(t), \quad t \in [0, 1], \quad (4)$$

where $\psi_0, \dots, \psi_M \in L^2([0, 1])$ are known basis functions, and $\mathbf{a}_k = (a_{k1}, a_{k2})^\top \in \mathbb{R}^2$ are the corresponding coefficients. Reconstructing the full function $\mathbf{C}$ from discrete observations thus reduces to estimating the coefficients $\mathbf{a}_k$, which can be achieved, for instance, by minimizing a least-squares criterion.

The choice of basis functions ψ_k depends on the application. In FDA, B-splines and Fourier basis functions are among the most commonly used options: B-splines are typically preferred for non-periodic data, whereas Fourier basis functions are well suited for periodic data (Ramsay and Silverman, 2005). In this work, we use Fourier basis functions due to the inherent periodicity of $\mathbf{C}$. Recall that the first $M + 1$ Fourier basis functions are defined as follows.

$$\psi_0(t) = 1, \quad \psi_k(t) = \begin{cases} \sqrt{2} \sin((k+1)\pi t) & \text{if } k \text{ odd,} \\ \sqrt{2} \cos(k\pi t) & \text{if } k \text{ even,} \end{cases} \quad (5)$$

for $t \in [0, 1]$, $k = 1, \dots, M$, with M even. These functions are orthonormal and satisfy $\int_0^1 \psi_k(t) dt = 0$ for all $k \geq 1$. For the remainder of the paper, ψ_k denotes the $(k+1)$ -th Fourier basis function defined in (5).

A key advantage of this choice is the well-known *time-shift property* of the Fourier basis, which, as we will show below, greatly simplifies the estimation of the reparametrization

variable. This property is formalized in the following lemma, which describes the effect of reparametrization on the Fourier basis functions.

Lemma 2.2. *Let $\boldsymbol{\psi} = (\psi_1, \dots, \psi_M)^\top$ and define $\boldsymbol{\psi}_\delta = (\psi_1 \circ \gamma_\delta, \dots, \psi_M \circ \gamma_\delta)^\top$, with $\gamma_\delta \in \Gamma$. Then we have*

$$\boldsymbol{\psi} = \boldsymbol{\beta}(\delta)\boldsymbol{\psi}_\delta, \quad (6)$$

where

$$\boldsymbol{\beta}(\delta) = \begin{pmatrix} \mathbf{O}_{2\pi\delta} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{O}_{4\pi\delta} & \dots & \mathbf{0} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{O}_{M\pi\delta} \end{pmatrix} \in \mathbb{R}^{M \times M},$$

is a sparse-orthogonal matrix and $\mathbf{O}_{k\pi\delta}, k \in \{2, 4, \dots, M\}$ is the rotation matrix defined in (3).

This result shows that reparametrization reduces to a simple linear transformation of the basis functions, where $\boldsymbol{\beta}(\delta)$ acts as a transfer matrix. Note that if $\delta \in \{0, 1\}$, then $\boldsymbol{\beta}(\delta)$ is the identity matrix, and the effect of reparametrization vanishes.

2.3.2 The translation and scaling transformations

To determine the translation and scaling variables, we rely on the definition of the shape variable $\tilde{\mathbf{C}}$ which is centered in $(0, 0)^\top$ and has unit norm. Direct computations, combined with the properties of the Fourier basis, yield

$$\begin{aligned} \mathbf{T} &= \int_0^1 \mathbf{C}(t)dt = \int_0^1 \sum_{k=0}^M \mathbf{a}_k \psi_k(t)dt = \mathbf{a}_0, \\ \rho &= \|\mathbf{C} - \mathbf{T}\|_{\mathcal{H}} = \left\| \sum_{k=1}^M \mathbf{a}_k \psi_k \right\|_{\mathcal{H}} = \sqrt{\sum_{k=1}^M \|\mathbf{a}_k\|_2^2}, \end{aligned}$$

with $\|\cdot\|_2$ the ℓ^2 -vector norm. These variables have a natural interpretation for analyzing an object's position within an image : the parameter $\mathbf{T}$ captures the spatial position of the object, while ρ reflects its scale, which is related to its distance from the camera.

Let $\mathbf{C}^*$ be the variable obtained after removing the effects of translation and scaling from $\mathbf{C}$:

$$\mathbf{C}^*(t) = \frac{1}{\rho} (\mathbf{C}(t) - \mathbf{T}) = \sum_{k=1}^M \boldsymbol{\alpha}_k \psi_k(t) = \boldsymbol{\alpha} \boldsymbol{\psi}(t), \quad t \in [0, 1] \quad (7)$$

with $\boldsymbol{\alpha}_k = \mathbf{a}_k / \rho \in \mathbb{R}^2$, for $k = 1, \dots, M$, and where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_M) \in \mathbb{R}^{2 \times M}$ and $\boldsymbol{\psi} = (\psi_1, \psi_2, \dots, \psi_M)^\top \in (L^2([0, 1]))^M$. This normalized representation will be used in the next section to estimate the rotation matrix $\mathbf{O}_\theta$ and the reparametrization function γ_δ .

Remark. *By construction, $\mathbf{C}^*$ belongs to the same space as $\tilde{\mathbf{C}}$, namely $\mathbf{S}^\infty$. In statistical shape analysis, $\mathbf{S}^\infty$ is called the "pre-shape space" (Dryden and Mardia (1998), Srivastava*

et al. (2010)) and it contains shapes that are possibly deformed by rotation and reparametrization. The "shape space", which could be denoted by $\mathbf{S}^\infty / \sim$, represents the quotient space of $\mathbf{S}^\infty$ with the equivalence relation $\sim$ defined as

$$\mathbf{f} \sim \mathbf{g} \iff \exists(\theta, \gamma) \in [0, 2\pi] \times \Gamma \text{ such that } \mathbf{f} = \mathbf{O}_\theta \mathbf{g} \circ \gamma, \quad (8)$$

where $\mathbf{f}, \mathbf{g} \in \mathbf{S}^\infty$. In other words, this relation says that $\mathbf{f} \sim \mathbf{g}$ if and only if $\mathbf{f}$ and $\mathbf{g}$ share the same shape.

2.3.3 The rotation and reparametrization transformations

Our strategy for estimating the rotation and reparametrization parameters consists in aligning the function $\mathbf{C}^*$ to a reference function $\boldsymbol{\mu} \in \mathbf{S}^\infty$. As in the univariate curve alignment setting, there is no ground truth for the choice of $\boldsymbol{\mu}$ (Marron et al., 2015). This reference can either be specified a priori or estimated from the data. A common choice, which we also adopt here, is to define $\boldsymbol{\mu}$ as the Fréchet mean of the shape variable, i.e. to define $\boldsymbol{\mu}$ as

$$\boldsymbol{\mu} = \mathbb{E}_d(\tilde{\mathbf{C}}) = \arg \min_{\mathbf{f} \in \mathbf{S}^\infty} \mathbb{E}[d(\mathbf{f}, \tilde{\mathbf{C}})^2], \quad (9)$$

where d denotes a suitable distance on $\mathbf{S}^\infty$. In Tucker et al. (2013), the authors consider an elastic distance d and propose an iterative procedure to approximate the corresponding Fréchet mean in the univariate setting. Such an approach is necessary since direct computation of the Fréchet mean is not feasible in practice, as it depends on the (unknown) aligned shapes $\tilde{\mathbf{C}}$. In Section 2.4, we propose a similar iterative strategy to estimate the Fréchet mean, using a distance d that is invariant to rotation and reparametrization. For the theoretical developments that follow, we assume that $\boldsymbol{\mu}$ is known and can be expressed as

$$\boldsymbol{\mu}(t) = \sum_{k=1}^M \mathbf{u}_k \psi_k(t) = \mathbf{u} \boldsymbol{\psi}(t),$$

with $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M) \in \mathbb{R}^{2 \times M}$.

By replacing $\mathbf{C}(t)$ in (7) with its representation in (1), we obtain that $\mathbf{C}^*$ can be written as

$$\mathbf{C}^*(t) = \mathbf{O}_\theta \tilde{\mathbf{C}} \circ \gamma_\delta(t), \quad t \in [0, 1]. \quad (10)$$

Determining the rotation matrix $\mathbf{O}_\theta$ and the reparametrization function γ_δ that align $\mathbf{C}^*$ to $\boldsymbol{\mu}$ is equivalent to finding the parameters θ and δ that minimize the distance, measured in terms of the $\|\cdot\|_{\mathcal{H}}$ norm, between $\mathbf{C}^* = \mathbf{O}_\theta \tilde{\mathbf{C}} \circ \gamma_\delta$ and $\boldsymbol{\mu}$. Equivalently, one might think of this problem as aligning the template $\boldsymbol{\mu}$ to the pre-shape $\mathbf{C}^*$, and thus finding the parameters θ and δ that minimize the distance between $\mathbf{O}_\theta \boldsymbol{\mu} \circ \gamma_\delta$ and $\mathbf{C}^*$. This leads to the optimisation problem

$$\min_{(\theta, \delta) \in [0, 2\pi] \times [0, 1]} \|\mathbf{O}_\theta \boldsymbol{\mu} \circ \gamma_\delta - \mathbf{C}^*\|_{\mathcal{H}}^2. \quad (11)$$

Using the fact that $\mathbf{O}_\theta^{-1} = \mathbf{O}_\theta^\top$ (since rotation matrices are orthogonal) and that the inverse of $\gamma_\delta \in \Gamma$ is $\gamma_{1-\delta}$ (as shown in the proof of Proposition 2.1), we can recover the shape variable $\tilde{\mathbf{C}}$ once $(\hat{\theta}, \hat{\delta})$ have been obtained, via $\tilde{\mathbf{C}} = \mathbf{O}_{\hat{\theta}}^\top \mathbf{C}^* \circ \hat{\gamma}_{1-\hat{\delta}}$.

Note that from the isometry propriety of Γ (Proposition 2.1), it follows that

$$\|\mathbf{f} \circ \gamma - \mathbf{g} \circ \gamma\|_{\mathcal{H}} = \|\mathbf{f} - \mathbf{g}\|_{\mathcal{H}}, \quad \text{for } \mathbf{f}, \mathbf{g} \in \mathcal{H} \text{ and } \gamma \in \Gamma.$$

As in the univariate case (Tucker et al., 2013), this invariance ensures that our estimators remain unchanged under a common reparametrization applied to both $\mathbf{C}$ and $\boldsymbol{\mu}$.

The next result builds on Lemma 2.2 to reformulate the objective function in (11), allowing it to be expressed in matrix form rather than in functional form.

Proposition 2.3. *The optimization problem (11) is equivalent to*

$$\min_{(\theta, \delta) \in [0, 2\pi] \times [0, 1]} \|\mathbf{O}_{\theta} \mathbf{u} \boldsymbol{\beta}(-\delta) - \boldsymbol{\alpha}\|_F^2 \quad (12)$$

where $\|\cdot\|_F$ is the Frobenius norm matrix and $\boldsymbol{\beta}(\cdot)$ the transfer matrix defined in Lemma 2.2.

Problems (11) and (12) do not admit closed-form solutions. To address this issue, we propose a novel alternating optimization algorithm inspired by the iterative closest point (ICP) algorithm (Zhang, 2021). The method operates directly in the functional space, without discretizing the curves, and is therefore well suited to the present framework. We refer to this procedure as the *Iterative Closest Function* (ICF) algorithm, which is described in the next section.

2.3.4 The ICF algorithm

Starting from an initial value of δ , the ICF algorithm alternates between the following two steps until convergence.

(i) **Estimation of θ for a given δ**

In this step, the estimator of θ is obtained by solving a Procrustes orthogonal problem:

$$\hat{\theta} = \arg \min_{\theta \in [0, 2\pi]} \|\mathbf{O}_{\theta} \boldsymbol{\mu} \circ \gamma_{\delta} - \mathbf{C}^*\|_{\mathcal{H}}^2. \quad (13)$$

The multivariate version of this problem has been extensively studied (see, e.g., Schönemann (1966)). In the functional setting, by arguments analogous to those in the classical case, one can show that $\hat{\theta}$ belongs to the set $\{\theta_1, \theta_2\}$, where θ_1 and θ_2 are the two solutions to the following equation:

$$\tan(\theta_k) = \langle \mathbf{C}^*, \boldsymbol{\mu} \circ \gamma_{\delta} \rangle_{\mathcal{H}}^{-1} \{ \langle X^*, \mu_1 \circ \gamma_{\delta} \rangle_{L^2} - \langle Y^*, \mu_2 \circ \gamma_{\delta} \rangle_{L^2} \}, k = 1, 2,$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2)^{\top}$ and $\mathbf{C}^* = (X^*, Y^*)^{\top}$. Once θ_1 and θ_2 have been computed, we evaluate the objective function in (13) at both values and select $\hat{\theta}$ as the one yielding the smallest value.

(ii) **Estimation of δ for a given θ**

When the rotation angle θ is known, problem (12) reduces to the following optimization problem:

$$\hat{\delta} = \arg \min_{\delta \in [0, 1]} \|\mathbf{O}_{\theta} \mathbf{u} \boldsymbol{\beta}(-\delta) - \boldsymbol{\alpha}\|_F^2. \quad (14)$$

Unlike step (i), this problem does not generally admit a closed-form solution. Instead, we solve it numerically using the result stated in the following proposition.

Proposition 2.4. *The solution $\hat{\delta}$ of problem (14) satisfies the equation*

$$\sum_{k \in \{1,3,\dots,M-1\}} w_k^{1,\theta} \sin((k+1)\pi\hat{\delta}) = \sum_{k \in \{1,3,\dots,M-1\}} w_k^{2,\theta} \cos((k+1)\pi\hat{\delta}), \quad (15)$$

where the coefficients $w_k^{1,\theta}$ and $w_k^{2,\theta}$ are derived from the matrix $(\mathbf{O}_\theta \mathbf{u})^\top \boldsymbol{\alpha}$.

For clarity, the exact expressions of the coefficients $(w_k^{l,\theta})_{k,l}$ and the proof of the proposition are provided in Appendix B.

We solve equation (15) with respect to $\hat{\delta}$ using a bisection method. This approach yields accurate results, as demonstrated in the numerical experiments of Section 4.

Remark. *The performance of the proposed ICF algorithm depends on the initialization of δ in step (i). We therefore recommend running the algorithm with multiple initial values and selecting the pair $(\hat{\theta}, \hat{\delta})$ that minimizes the objective function in (11).*

2.4 Estimation of the Fréchet mean

The Fréchet mean introduced in (9) depends on the choice of a distance on $\mathbf{S}^\infty$. In this work, motivated by (11) and Proposition 2.1, we define the function $d : \mathbf{S}^\infty \times \mathbf{S}^\infty \rightarrow \mathbb{R}^+$ as

$$d(\mathbf{f}, \mathbf{g}) = \min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{f} \circ \gamma_\delta - \mathbf{O}_\theta \mathbf{g}\|_{\mathcal{H}}.$$

This quantity can be interpreted as a Procrustes-type distance (Dryden and Mardia, 1998), as it is invariant to rotation and reparametrization:

$$d(\mathbf{f}, \mathbf{g}) = d(\mathbf{O}_{\theta_1} \mathbf{f} \circ \gamma_{\delta_1}, \mathbf{O}_{\theta_2} \mathbf{g} \circ \gamma_{\delta_2}), \quad \forall \delta_1, \delta_2 \in [0,1], \forall \theta_1, \theta_2 \in [0,2\pi].$$

In particular, this implies that $d(\mathbf{C}^*, \tilde{\mathbf{C}}) = 0$, and more generally that d is not a proper distance on $\mathbf{S}^\infty$, since the separability property does not hold:

$$d(\mathbf{f}, \mathbf{g}) = 0 \not\Rightarrow \mathbf{f} = \mathbf{g}, \quad \text{for } \mathbf{f}, \mathbf{g} \in \mathbf{S}^\infty.$$

However, d defines a proper distance when considered on the shape space $\mathbf{S}^\infty / \sim$ rather than on the pre-shape space $\mathbf{S}^\infty$ (see Lemma B.2 in appendix). Plugging the distance d into (9), one can show that

$$\boldsymbol{\mu} = \mathbb{E}_d(\tilde{\mathbf{C}}) = \arg \max_{\mathbf{f} \in \mathbf{S}^\infty} \mathbb{E} \left[\langle \tilde{\mathbf{C}}, \mathbf{f} \rangle_{\mathcal{H}} \right].$$

We now describe how to estimate $\boldsymbol{\mu}$ from a sample of n shapes $\tilde{\mathbf{C}}_1, \dots, \tilde{\mathbf{C}}_n$. We assume that each shape admits a Fourier basis expansion:

$$\tilde{\mathbf{C}}_i(t) = \sum_{k=1}^M \tilde{\mathbf{a}}_{i,k} \psi_k(t) = \tilde{\mathbf{A}}_i \boldsymbol{\psi}(t), \quad t \in [0,1], \quad i = 1, \dots, n,$$

where $\tilde{\mathbf{A}}_i \in \mathbb{R}^{2 \times M}$ is the corresponding coefficient matrix. We define the estimator of the Fréchet mean as $\hat{\boldsymbol{\mu}} = \hat{\mathbf{u}}\boldsymbol{\psi}$, where

$$\hat{\mathbf{u}} = \arg \max_{\mathbf{u} \in \mathbb{R}^{2 \times M}} \frac{1}{n} \sum_{i=1}^n \langle \tilde{\mathbf{A}}_i, \mathbf{u} \rangle_F, \quad (16)$$

and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product.

The optimization problem in (16) is non-convex and does not admit a closed-form solution, and must therefore be solved numerically. This problem is structurally similar to the Fréchet mean estimation problem encountered in classical statistical shape analysis, for which a numerical solution is proposed in Dryden (2023). We rely on their implementation to solve (16), using the general-purpose nonlinear optimization function *nlm* from the **stats** package in R (R Core Team, 2023). Additional details on the implementation are provided in our publicly available code (<https://github.com/imoindjie/Shape-FDA>).

2.5 Iterative estimation of shapes and of the Fréchet mean

From the previous sections on alignment using the ICF algorithm and on Fréchet mean estimation, it follows that there is a strong interdependence between shape estimation and mean estimation: the Fréchet mean is required to align the shapes, while aligned shapes are in turn needed to estimate the Fréchet mean. Consequently, poor estimates at either stage may adversely affect the overall procedure. To address this issue, we propose an iterative algorithm that alternates between shape alignment and Fréchet mean estimation.

The complete estimation pipeline, from raw data to aligned shapes and the Fréchet mean, is summarized below.

- **Input:**

- $\{(\mathbf{C}_i(t_{i1}), \dots, \mathbf{C}_i(t_{iJ_i}))\}_{i=1}^n$: observed curves on a discrete grid
- M : number of Fourier basis functions
- ξ : convergence threshold

- **Algorithm:**

1. **Smoothing**

For each $i \in \{1, \dots, n\}$, estimate the coefficient vectors $\mathbf{a}_{i,k}$, $k = 0, \dots, M$, defined in (4), by minimizing a least-squares criterion based on the discretized observations $(\mathbf{C}_i(t_{i1}), \dots, \mathbf{C}_i(t_{iJ_i}))$. The resulting smoothed curve is

$$\hat{\mathbf{C}}_i(t) = \sum_{k=0}^M \hat{\mathbf{a}}_{i,k} \psi_k(t), \quad t \in [0, 1].$$

2. **Centering and normalization**

For each $i \in \{1, \dots, n\}$, define the estimated pre-shape as

$$\hat{\mathbf{C}}_i^*(t) = \frac{1}{\hat{\rho}_i} \left(\hat{\mathbf{C}}_i(t) - \hat{\mathbf{T}}_i \right) = \sum_{k=1}^M \hat{\boldsymbol{\alpha}}_{i,k} \psi_k(t),$$

where $\hat{\mathbf{T}}_i = \hat{\mathbf{a}}_{i,0}$, $\hat{\rho}_i = \sqrt{\sum_{k=1}^M \|\hat{\mathbf{a}}_{i,k}\|_2^2}$, and $\hat{\boldsymbol{\alpha}}_{i,k} = \hat{\mathbf{a}}_{i,k}/\hat{\rho}_i$, $k = 1, \dots, M$.

3. Iterative alignment and Fréchet mean estimation

Initialize $\hat{\boldsymbol{\mu}}$ by randomly selecting one pre-shape from $\{\hat{\mathbf{C}}_1^*, \dots, \hat{\mathbf{C}}_n^*\}$. Then repeat the following steps until convergence, i.e., until $\eta \leq \xi$:

- a. For each $i \in \{1, \dots, n\}$, estimate the rotation angle θ_i and the reparametrization parameter δ_i using the ICF algorithm, described in Section 2.3.4, with $\hat{\boldsymbol{\mu}}$ as the template. Define the estimated shape as

$$\hat{\mathbf{C}}_i = \mathbf{O}_{\theta_i}^\top \hat{\mathbf{C}}_i^* \circ \gamma_{1-\delta_i}.$$

- b. Using the shapes $\hat{\mathbf{C}}_1, \dots, \hat{\mathbf{C}}_n$, compute an estimate $\hat{\boldsymbol{\mu}}$ of the Fréchet mean as described in Section 2.4. Then compute the distance-based variance

$$\eta = \frac{1}{n} \sum_{i=1}^n \left(\cos^{-1} \left(\langle \hat{\mathbf{C}}_i, \hat{\boldsymbol{\mu}} \rangle_{\mathcal{H}} \right) \right)^2.$$

3 Modeling $\mathbf{C}$ with a PCA-based approach

In the previous section, we introduced a procedure to estimate the deformation parameters associated with a planar curve and to recover its underlying shape. As a result, each curve $\mathbf{C}$ can be decomposed into a shape component and a set of deformation variables, namely $(\tilde{\mathbf{C}}, \rho, \theta, \delta, \mathbf{T})$.

We now address the problem of modeling the variability of these quantities. In order to preserve interpretability, we propose to model the shape and the deformation variables separately. This strategy is motivated by the fact that these two sources of variability play fundamentally different roles and may be of independent interest in applications.

Following this idea, and inspired by phase-amplitude modeling approaches of Tucker et al. (2013), we introduce the two random variables

$$\mathbf{Z}_1 = \rho \tilde{\mathbf{C}} \in \mathcal{H}, \quad \text{and} \quad \mathbf{Z}_2 = \left(\tan \left(\frac{\pi}{2} \left(\delta - \frac{1}{2} \right) \right), \tan \left(\frac{1}{4} \left(\frac{\theta}{2} - \pi \right) \right), \mathbf{T}^\top \right)^\top \in \mathbb{R}^4.$$

The variable $\mathbf{Z}_1$ is a linear transformation of $(\tilde{\mathbf{C}}, \rho) \in \mathbf{S}^\infty \times \mathbb{R}^+$ that preserves all the information while allowing us to work in the Hilbert space $\mathcal{H}$. This is convenient since the pre-shape space $\mathbf{S}^\infty$ does not naturally admit well-defined notions of mean or covariance. Similarly, we work with $\mathbf{Z}_2 \in \mathbb{R}^4$ instead of $(\delta, \theta, \mathbf{T})$, as the chosen transformations map bounded or periodic variables into an unconstrained Euclidean space, making standard multivariate techniques applicable. Note that both transformations are invertible.

Since $\mathbf{Z}_1$ is a bivariate functional variable, we model it using multivariate functional principal component analysis (MFPCA) (Jacques and Preda, 2014), which yields

$$\mathbf{Z}_1(t) \simeq \mathbb{E}[\mathbf{Z}_1(t)] + \sum_{k=1}^{M_1} \xi_k^{(1)} \phi_k(t), \quad t \in [0, 1], \quad (17)$$

where $\phi_1, \dots, \phi_{M_1} \in \mathcal{H}$ are the eigenfunctions of the covariance operator of $\mathbf{Z}_1$. For $\mathbf{Z}_2$, we use standard multivariate PCA, leading to

$$\mathbf{Z}_2 \simeq \mathbb{E}[\mathbf{Z}_2] + \sum_{k=1}^{M_2} \xi_k^{(2)} \mathbf{U}_k, \quad (18)$$

where $\mathbf{U}_1, \dots, \mathbf{U}_{M_2} \in \mathbb{R}^4$ are the eigenvectors of the covariance matrix of $\mathbf{Z}_2$. For $j = 1, 2$, the scores $\{\xi_k^{(j)}\}$ are centered and uncorrelated, with

$$\mathbb{E}(\xi_k^{(j)} \xi_l^{(j)}) = \begin{cases} \lambda_k^{(j)} & \text{if } k = l, \\ 0 & \text{otherwise,} \end{cases}$$

where $\lambda_k^{(j)}$ denotes the k -th eigenvalue of the covariance operator of $\mathbf{Z}_j$.

A possible alternative would be to perform a single PCA on the joint variable $(\mathbf{Z}_1, \mathbf{Z}_2)$ using methods for hybrid data (see, e.g., Chap. 10 of Ramsay and Silverman (2005)). While such an approach may lead to a more compact representation, it obscures the interpretation of the principal components. In particular, it becomes difficult to disentangle whether a given mode of variation is driven by shape or by deformation effects. By contrast, the proposed approach preserves this separation, thereby providing a more interpretable representation. This is especially advantageous in applications such as anomaly detection or classification, where identifying the source of variability is of primary importance.

3.1 A generative model for $\mathbf{C}$

The representations (17)–(18) provide a finite-dimensional parametrization of both shape and deformation variability through the score vectors $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$. To obtain a fully probabilistic model, it remains to specify a distribution for these scores.

A natural and widely used choice is to assume that the concatenated score vector

$$\boldsymbol{\xi} = (\xi_1^{(1)}, \dots, \xi_{M_1}^{(1)}, \xi_1^{(2)}, \dots, \xi_{M_2}^{(2)}) \in \mathbb{R}^{M_1+M_2}$$

follows a multivariate Gaussian distribution with mean zero and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{1,2} \\ \boldsymbol{\Sigma}_{1,2}^\top & \boldsymbol{\Sigma}_2 \end{pmatrix},$$

where $\boldsymbol{\Sigma}_j = \text{diag}(\lambda_1^{(j)}, \dots, \lambda_{M_j}^{(j)})$ and $\boldsymbol{\Sigma}_{1,2}$ captures possible dependencies between shape and deformation scores.

This specification defines a generative model for planar curves through the following procedure:

1. Sample $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$;
2. Construct $\mathbf{Z}_1$ and $\mathbf{Z}_2$ using (17) and (18);

3. Construct $\mathbf{C}$ from $\mathbf{Z}_1$ and $\mathbf{Z}_2$ via

$$\mathbf{C} = \mathbf{O}_{8 \tan^{-1}(Z_{22}) + 2\pi} \mathbf{Z}_1 \circ \gamma_{\frac{2}{\pi} \tan^{-1}(Z_{21}) + \frac{1}{2}} + (Z_{23}, Z_{24})^\top.$$

It is important to note that the construction step 2 relies on the mean functions and the eigenstructures of the covariance operators of $\mathbf{Z}_1$ and $\mathbf{Z}_2$. In practice, these quantities are unknown and are replaced by their empirical estimates obtained from the sample. The resulting model is therefore a plug-in generative model, where population quantities are approximated by their empirical counterparts.

This construction provides a flexible and interpretable model for random planar curves, in which shape and deformation variability are explicitly represented while allowing for dependence between them through the joint distribution of the scores.

4 Simulation study

In this section, we present a simulation study to assess the performance of the proposed alignment methodology. Since translation and scaling can be directly recovered from the Fourier coefficients, we focus on the more challenging task of estimating the rotation and reparametrization parameters, namely θ and γ . In particular, we evaluate the ability of the alignment procedures to recover the underlying shapes from their associated pre-shapes.

To this end, we compare the proposed ICF algorithm with two widely used alignment methods: the iterative closest point algorithm (ICP; see Zhang (2021)) and the elastic alignment approach of Srivastava et al. (2010), based on the square-root velocity function (SRVF) framework. A key distinction between these methods lies in the representation of the curves. Both ICP and SRVF-based approaches operate directly on discretely observed curves, whereas ICF first reconstructs a continuous representation using a Fourier basis and performs alignment in the corresponding functional space.

Since our goal is to compare the alignment procedures themselves, we use a common fixed template throughout this section instead of the Fréchet mean, whose estimation would introduce an additional source of variability.

4.1 Data simulation

We generate a sample of pre-shapes $\mathbf{C}_i^*$, $i = 1, \dots, n$, according to

$$\mathbf{C}_i^*(t) = \mathbf{O}_{\theta_i} \tilde{\mathbf{C}}_i \circ \gamma_{\delta_i}(t), \quad t \in [0, 1],$$

where the reparametrization parameters are sampled as $\delta_i \stackrel{i.i.d}{\sim} \mathcal{U}(0, 1)$, and the rotation angles are defined by $\theta_i = 2\pi\delta_i$.

Inspired by Weisstein (2003), the shape variables are constructed as $\tilde{\mathbf{C}}_i = \mathbf{H}_i / \|\mathbf{H}_i\|_{\mathcal{H}}$, where $\mathbf{H}_i$ is a heart-shaped parametric curve defined by

$$\mathbf{H}_i(t) = \begin{pmatrix} b_{i0} \sin^3(\pi(2t - 1)) \\ \sum_{k=1}^4 b_{ik} \cos(k\pi(2t - 1)) \end{pmatrix}, \quad t \in [0, 1], \quad (19)$$

with $(b_{i0}, \dots, b_{i4}) \stackrel{i.i.d}{\sim} \mathcal{N}((16, 13, -5, -2, -1)^\top / 16, \sigma^2 \mathbf{I}_5)$. Sampling these coefficients induces variability in the shapes $\tilde{\mathbf{C}}_i$, while σ controls the magnitude of this variability. We consider two levels of shape variability, $\sigma \in \{0.01, 0.1\}$.

We define the template as the normalized mean curve,

$$\boldsymbol{\mu} = \mathbb{E}(\mathbf{H}_i) / \|\mathbb{E}(\mathbf{H}_i)\|_{\mathcal{H}},$$

with a reference parametrization such that $\delta = 0$ corresponds to the pointed end of the heart.

The first row of Figure 5 displays the template $\boldsymbol{\mu}$ and its coordinate functions. The second and third rows show observations $\tilde{\mathbf{c}}_i$ of simulated shapes $\tilde{\mathbf{C}}_i$ for $\sigma = 0.01$ and $\sigma = 0.1$, respectively, illustrating the impact of σ on shape variability.

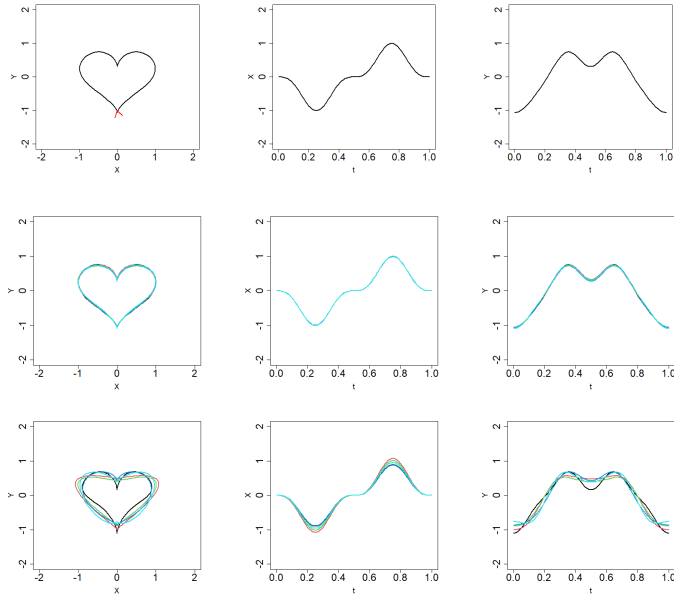


Figure 5: Template $\boldsymbol{\mu}$ and its coordinate functions (first row), and observations of simulated shapes and their coordinate functions for $\sigma = 0.01$ (second row) and $\sigma = 0.1$ (third row)

Figure 6 displays representative simulated pre-shapes $\mathbf{c}_i^*$ for $\sigma = 0.1$, together with their coordinate functions.

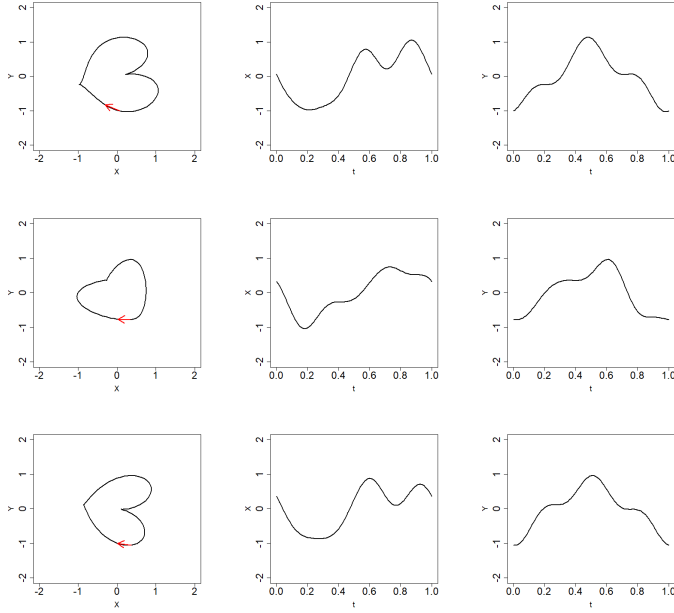


Figure 6: Observations of simulated pre-shapes and their coordinate functions for $\sigma = 0.1$.

For each value of σ , we generate a dataset $\mathbf{c}_1^*, \dots, \mathbf{c}_n^*$ of size $n = 200$. Each pre-shape $\mathbf{c}_i^*$ is then discretized by sampling $J_i = J$ points $\{\mathbf{c}_i^*(t_{i,j})\}_{j=1}^J$, where the sampling locations $t_{i,j}$ are independently drawn from a discrete uniform distribution over $\{0.01, \dots, 0.99\}$, with endpoints fixed at $t_{i,1} = 0$ and $t_{i,J} = 1$. To assess the impact of discretization, we consider $J \in \{15, 50, 90\}$.

These simulation settings allow us to evaluate the effect of both shape variability and sampling resolution on the performance of the alignment methods.

4.2 Numerical results

We apply the three alignment procedures, namely ICF, ICP, and SRVF-based alignment, to each simulated dataset. As discussed previously, the ICF method relies on a functional representation of the curves. Consequently, the discretized observations are first smoothed according to the reconstruction procedure described in Section 2.5. In all experiments, this smoothing step is performed using $M = 10$ Fourier basis functions. We use in-house implementations for both the ICF and ICP algorithms, while the SRVF-based alignment is carried out using the `fdasrvf` package Tucker (2026), provided by the original authors of the method.

Figure 7 illustrates alignment results for one pre-shape observation $\mathbf{c}^*$ of the dataset simulated with $\sigma = 0.1$. On top of the figure we display the template function, the observation $\mathbf{c}^*$ and the true reparametrization and shape functions, γ and $\tilde{\mathbf{c}}$, associated with it. Results are shown for two sampling resolutions J . For each resolution, we display the discretized pre-shape, and the estimated reparametrization and shape functions, $\hat{\gamma}$ and $\hat{\tilde{\mathbf{c}}}$, produced by each method. The objective is to recover the correct starting point of the curve and align the resulting shape with the reference template $\boldsymbol{\mu}$.

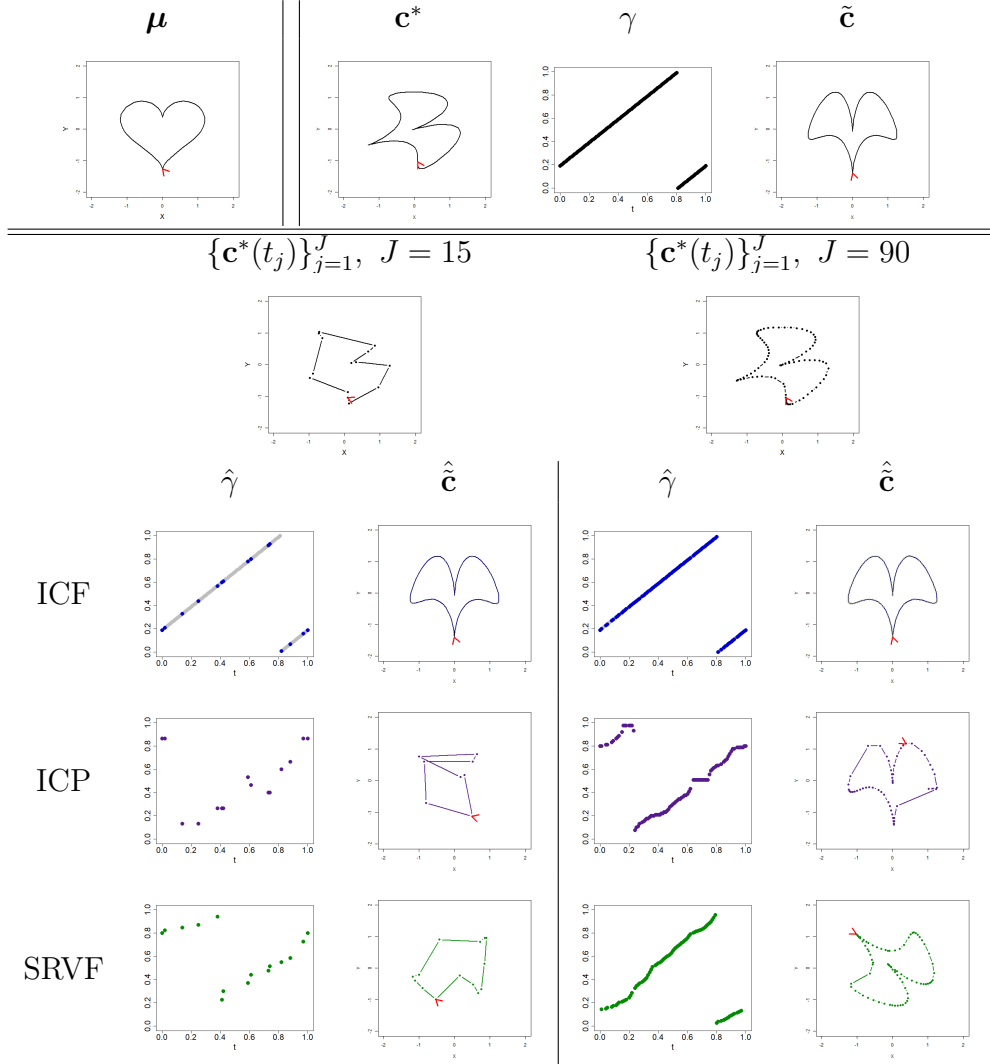


Figure 7: Example of a simulated pre-shape curve $\mathbf{c}^*$ together with its associated reparametrization function γ and shape function $\tilde{\mathbf{c}}$ (top row) under $\sigma = 0.1$. Results are shown for two sampling resolutions J . For each alignment method, we display the estimated reparametrization function $\hat{\gamma}$ and the estimated shape $\hat{\mathbf{c}}$ obtained after aligning the discretized pre-shape observations $\{\mathbf{c}^*(t_j)\}_{j=1}^J$ (second row) to the template μ (top left).

Several important differences between the methods can be observed from Figure 7. First, ICP is designed for unordered point clouds and therefore does not preserve the original parametrization of the curves. During the alignment process, point correspondences may be reassigned arbitrarily, which can substantially distort the underlying shape. This phenomenon is particularly visible when the number of sampled points is small ($J = 15$).

The figure also highlights the additional flexibility provided of elastic alignment methods. Although this flexibility can be advantageous in general settings, it may lead the SRVF-based procedure to favor elastic deformations instead of correctly identifying the starting point δ . In such cases, the shape can become artificially distorted rather than properly registered to the template. This effect is again more pronounced for small values of J .

Finally, we observe that the SRVF-based method may also struggle to accurately estimate the rotation parameter. This effect is present in both scenarios displayed in Figure 7. While the exact cause of this behavior is not fully understood, it may be related to the joint estimation of rotation and reparametrization in elastic alignment frameworks, where these two sources of variability can partially compensate for one another.

To quantitatively assess the accuracy of the estimated deformation parameters, we compute the following mean squared errors (MSE). Since rotations are periodic, the error associated with θ is measured through its representation on the unit circle:

$$\text{MSE}_\theta = \frac{1}{n} \sum_{i=1}^n \left\| \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} - \begin{pmatrix} \cos(\hat{\theta}_i) \\ \sin(\hat{\theta}_i) \end{pmatrix} \right\|_2^2,$$

while the error associated with the reparametrization function is defined as

$$\text{MSE}_\gamma = \frac{1}{n} \sum_{i=1}^n \frac{1}{J} \sum_{j=1}^J (\gamma_{\delta_i}(t_{i,j}) - \hat{\gamma}_i(t_{i,j}))^2,$$

where $\hat{\gamma}_i$ denotes the estimated reparametrization function for the i -th curve.

σ	J	MSE_γ			MSE_θ		
		ICF	ICP	SRVF	ICF	ICP	SRVF
0.01	15	0.03	0.15	0.14	0.40	1.15	1.97
0.01	50	0.03	0.18	0.04	0.31	0.59	2.01
0.01	90	0.04	0.17	0.02	0.46	0.31	2.04
0.10	15	0.03	0.15	0.15	0.32	1.28	1.75
0.10	50	0.02	0.18	0.05	0.26	0.36	1.65
0.10	90	0.03	0.18	0.04	0.35	0.37	1.58

Table 1: Mean squared errors associated with the estimation of the reparametrization function γ and the rotation parameter θ for different values of σ and J , obtained with the three alignment methods.

Table 1 summarizes the MSE values obtained under all simulation scenarios. A clear difference between ICF and ICP can be observed across all settings. As already mentioned, while ICF explicitly preserves the ordering of points along the curve, ICP operates on unordered point clouds and may alter point correspondences during alignment. This lack of

structural constraint leads to substantially larger errors for ICP, particularly for the estimation of γ and, when J is small, for the estimation of θ as well.

The comparison between ICF and the SRVF-based alignment is more nuanced for the estimation of γ . When the sampling resolution is low ($J = 15$), ICF clearly outperforms SRVF. As the number of sampled points increases, however, the performance of the SRVF method improves substantially and becomes comparable to, or slightly better than, that of ICF.

In contrast, a noticeably different behavior is observed for the estimation of the rotation parameter θ . Across all scenarios, the SRVF-based method exhibits significantly larger errors than both ICF and ICP, with MSE values remaining consistently high regardless of the sampling resolution. This observation supports the qualitative behavior already noted in Figure 7. As mentioned previously, this effect may stem from the joint estimation of rotation and reparametrization in elastic alignment frameworks, where the two deformation components can partially compensate for one another. However, a more thorough investigation would be required to precisely identify the origin of this effect.

5 Real data analysis

In this section, we analyze datasets from the *MPEG-7* database¹, which has been widely used in statistical shape analysis (see, for example, Srivastava et al. (2010), Srivastava and Klassen (2016), Carlier et al. (2016)). This database contains binarized images of 101 different object classes, each with approximately 20 instances.

In our study, we focus on five representative classes: butterflies, forks, bats, horseshoes, and spoons. For clarity of presentation, we report detailed results for the butterfly and fork datasets, while the results for the remaining classes are provided in Appendix A. Figure 8 displays sample images for these two classes. As can be observed, the images exhibit variability in rotation, scaling, and translation, leading to differences in orientation, size, and positioning.

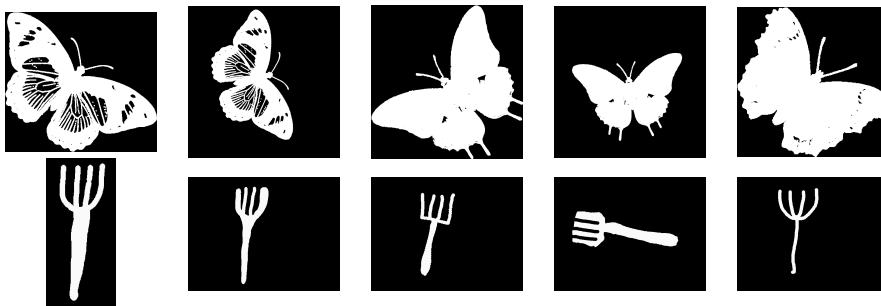


Figure 8: Examples of images from the database for the butterfly and fork objects.

Before applying our methodology, planar curves must be extracted from the binarized images. We perform this step using the OpenCV library in Python (Bradski, 2000). Figure 9 illustrates the resulting closed planar curves corresponding to the images in Figure 8.

¹<https://dabi.temple.edu/external/shape/MPEG7/dataset.html>

The extraction process introduces variability in the parametrization of the curves, as the starting points differ across observations, particularly for the fork shapes. In addition, the distribution of deformation variables appears to vary across object classes. For instance, butterfly shapes exhibit substantial variability in orientation, whereas fork shapes are mostly aligned in a similar direction. This suggests that the distribution of certain deformation parameters, such as rotation, may depend on the object class. More generally, this highlights that the deformations applied to a shape are not independent of the shape itself.

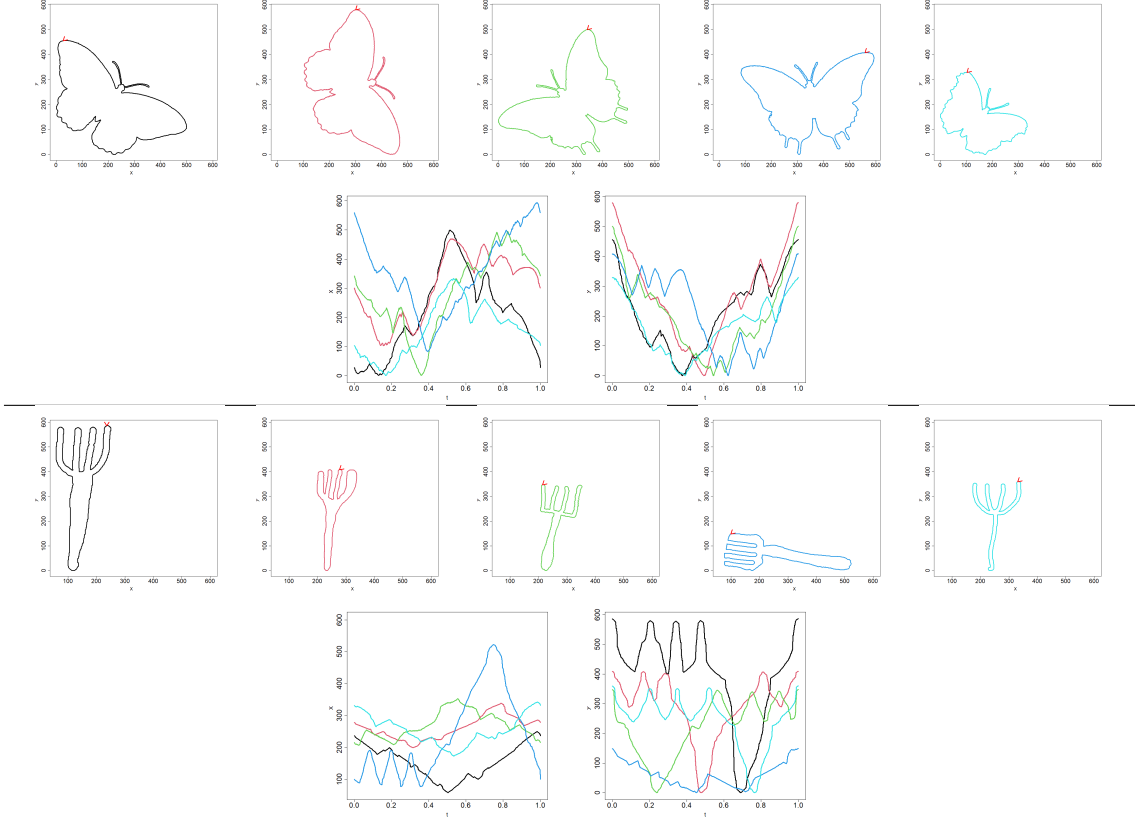


Figure 9: Planar closed curves extracted from the binarized images of Figure 8 (first row of each block) with their associate coordinate functions (second row of each block).

5.1 Results from the alignment procedure

Before aligning the planar curves, we first recover their functional form using a basis expansion with $M = 50$ Fourier functions. Let $\hat{\mathbf{c}}_1^s, \dots, \hat{\mathbf{c}}_{n_s}^s$ denote the resulting functions, where the index s refers to the dataset: $s = b$ for butterflies and $s = f$ for forks. We then estimate the translation and scaling parameters for each curve, yielding the standardized shapes $\hat{\mathbf{c}}_i^{*s}, i = 1, \dots, n_s$, illustrated in Figure 10.

To estimate the rotation and reparametrization parameters, we define a template $\boldsymbol{\mu}^s$. While a reference curve could be selected from the dataset, we instead use the Fréchet mean introduced in Section 2.4. The full iterative procedure described in Section 2.5 is then applied to jointly estimate the Fréchet mean and align the curves. This results in the estimated

shape curves $\hat{\mathbf{c}}_i^s, i = 1, \dots, n_s$, also shown in Figure 10. The coordinate functions appear well aligned, indicating that the proposed alignment procedure performs effectively. Moreover, despite the relatively small sample size, the estimated Fréchet means yield coherent and interpretable representative shapes. This is a non-trivial task in shape analysis Srivastava and Klassen (2016), as it requires jointly accounting for geometric variability and deformation effects, and further illustrates the stability of the proposed estimation procedure.

A key feature of our approach is that it provides explicit estimates of the deformation parameters for each curve, enabling a direct statistical analysis of these quantities. The estimated deformation parameters for both datasets are displayed in Figure 11. We observe clear differences in their distributions across object classes. For instance, in the butterfly dataset, the translation components exhibit a strong linear relationship, whereas in the fork dataset, the translation vectors are more tightly clustered around a central location. Similarly, the rotation parameters $\hat{\theta}_i$ and the reparametrization parameters show substantially greater variability in the butterfly dataset than in the fork dataset.

These observations are consistent with the visual variability seen in Figure 9 and highlight that deformation patterns are strongly shape-dependent. Unlike approaches that treat deformations as nuisance parameters, our framework allows them to be explicitly modeled and interpreted.

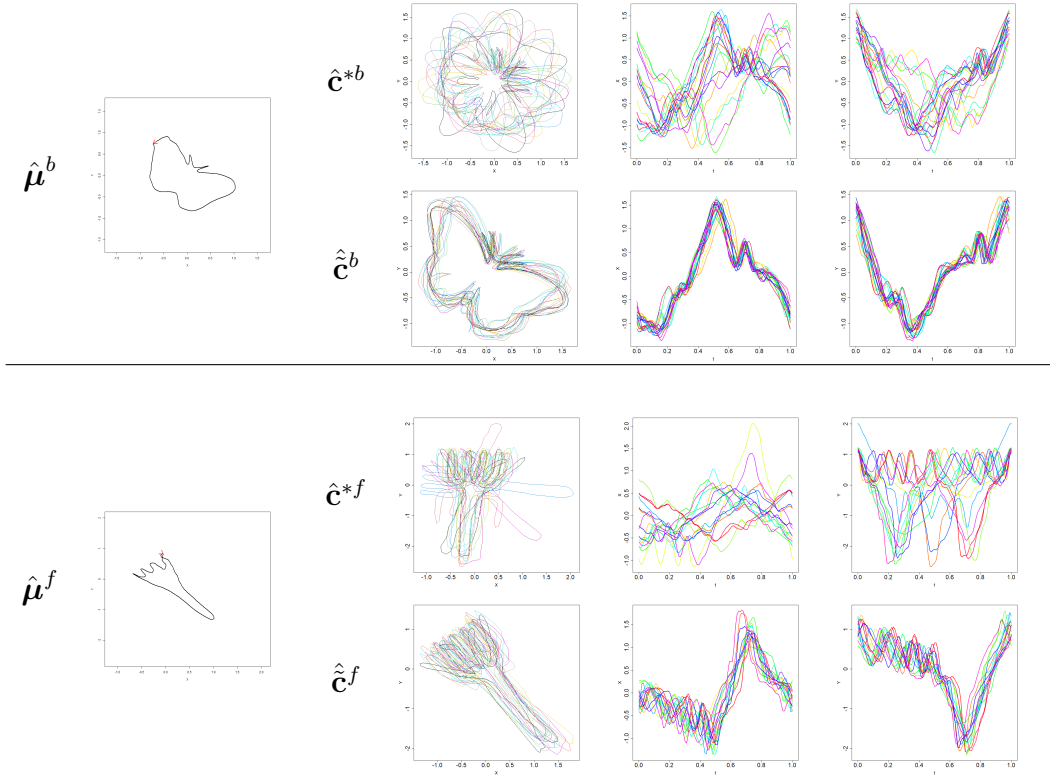


Figure 10: Illustration of the alignment procedure for the butterfly dataset (top block) and the fork dataset (bottom block). In each block, the left column displays the estimated Fréchet mean. The first row shows estimated pre-shapes together with their coordinate functions, while the second row presents the corresponding aligned shapes.

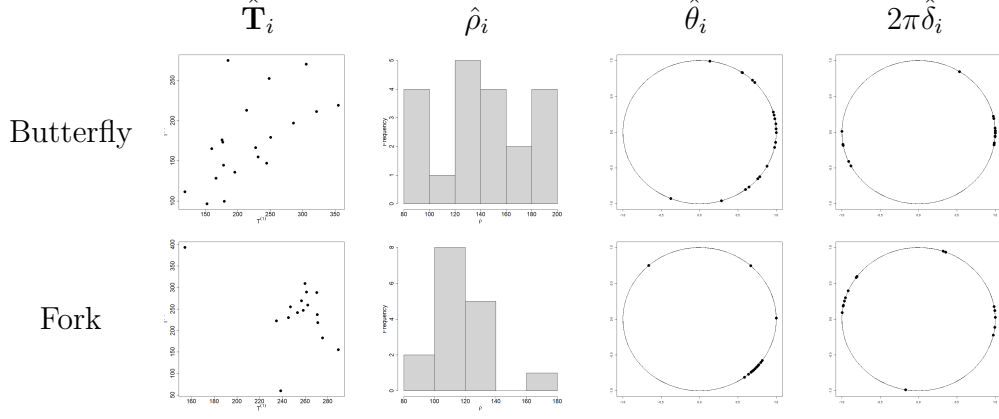


Figure 11: Plots of the estimated deformation parameters associated with each curve in the butterfly (first row) and fork (second row) datasets

5.2 Results from modeling C

We now apply the PCA framework introduced in Section 3 to analyze the butterfly and fork datasets. For each dataset, we construct the samples $\mathbf{z}_1 = (\mathbf{z}_{11}, \dots, \mathbf{z}_{n_s1})$ and $\mathbf{z}_2 = (\mathbf{z}_{12}, \dots, \mathbf{z}_{n_s2})$ where

$$\mathbf{z}_{i1} = \hat{\rho}_i \hat{\mathbf{c}}_i \text{ and } \mathbf{z}_{i2} = \left(\tan \left(\frac{\pi}{2} \left(\hat{\delta}_i - \frac{1}{2} \right) \right), \tan \left(\frac{1}{4} \left(\frac{\hat{\theta}_i}{2} - \pi \right) \right), \hat{\mathbf{T}}_i^\top \right)^\top,$$

for $i = 1, \dots, n_s$. The variables $\mathbf{z}_1$ and $\mathbf{z}_2$ are then analyzed separately using the PCA-based approach described in Section 3.

The first row of Figure 12 displays the estimated mean shape (in black) for the butterfly dataset, along with perturbations obtained by adding (in red) or subtracting (in blue) a multiple of the first three eigenfunctions. The second row presents the same visualization for the fork dataset. This type of representation, inspired by (Ramsay and Silverman, 2005, Chap. 8), provides insight into the main modes of variation captured by the model.

For the butterfly dataset, the dominant sources of variability are concentrated in the wings. The first eigenfunction primarily captures variations in wing size relative to body length, essentially accounting for the scaling effect ρ . The second and third eigenfunctions reflect more subtle changes in wing geometry, corresponding to intrinsic shape variability. This is particularly noticeable for the third eigenfunction, where large positive scores lead to short, pointed wings (red), while large negative scores produce elongated, rounded wings (blue).

For the fork dataset, the modes of variation are less immediately interpretable, although a similar pattern can still be observed. The first eigenfunction mainly captures variations in the overall size of the fork, again reflecting the scaling effect ρ . The second eigenfunction captures a key structural difference in the dataset by distinguishing between three-pronged and four-pronged forks, with large positive scores (red) associated with three-pronged shapes.

Overall, these results demonstrate that the proposed PCA framework captures meaningful and interpretable modes of shape variability, even in the presence of deformation effects.

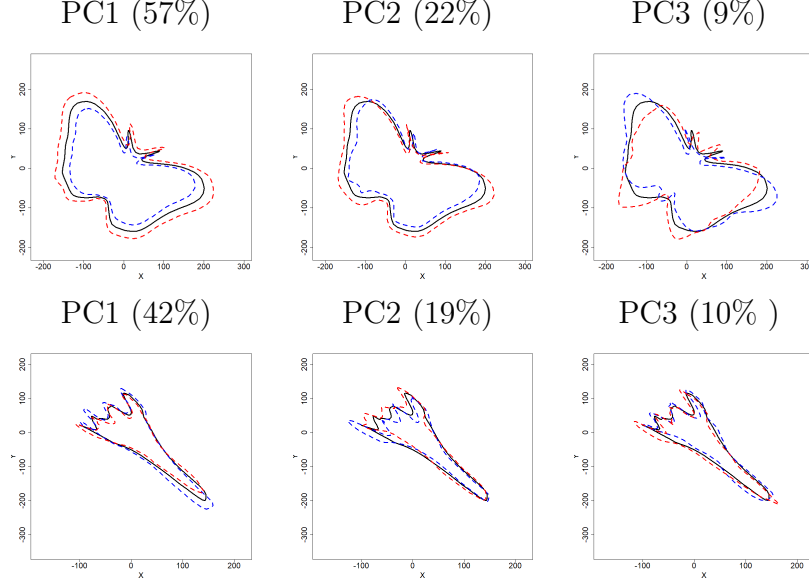


Figure 12: Plots of the estimated mean function $\bar{\mathbf{z}} = \sum_i \mathbf{z}_{i1}$ in black, of $\bar{\mathbf{z}} - 20\hat{\phi}_k$ in blue and of $\bar{\mathbf{z}} + 20\hat{\phi}_k$ in red, for $k = 1$ (first column), $k = 2$ (second column) and $k = 3$ (third column). The first row corresponds to the butterfly dataset, while the second row corresponds to the fork dataset. The number in parentheses in each plot title indicates the percentage of total variation explained by the corresponding eigenfunction.

To assess the ability of the proposed PCA model to capture the variability of planar curves, we generate new samples from the estimated model. The underlying idea is that a well-fitted model should produce synthetic shapes that are both realistic and representative of the original datasets.

We define our generative model as described in Section 3.1. In particular, we simulate the score vectors $\boldsymbol{\xi} \in \mathbb{R}^{M_1+M_2}$ from the estimated joint Gaussian model. The number of retained principal components, M_1 for $\mathbf{Z}_1$ and M_2 for $\mathbf{Z}_2$, is chosen such that the cumulative proportion of explained variance reaches at least 90%. This leads to $M_1 = 4$ and $M_2 = 4$ for the butterfly dataset, and $M_1 = 7$ and $M_2 = 4$ for the fork dataset.

For comparison, we also consider a baseline approach based on multivariate functional PCA (MFPCA), applied directly to the smoothed curves $\hat{\mathbf{c}}_i^s$, $i = 1, \dots, n_s$, with $s \in \{b, f\}$, following Happ and Greven (2018). As before, we retain a number of principal components, denoted M , such that at least 90% of the variance is explained. This results in $M = 3$ for the butterfly dataset and $M = 4$ for the fork dataset.

This approach serves as a natural baseline to assess the impact of the alignment step. Since geometric variability and deformation effects are not explicitly disentangled, we expect these sources of variation to be confounded, which may limit the ability of the model to generate coherent and realistic contours.

Figures 13 and 14 illustrate examples of generated butterfly and fork curves, respectively. Each figure consists of three rows, each displaying five curves: row (a) shows curves generated using our approach without deformations, row (b) shows curves generated using our approach with deformations, and row (c) displays curves generated using the MFPCA baseline.

To quantitatively assess how well the generated curves resemble those in the original datasets, we introduce two similarity metrics based on the distance d defined in Section 2.4:

$$D_1(\mathbf{y}^s) = \frac{1}{n_s} \sum_{i=1}^{n_s} d(\mathbf{y}^{*s}, \hat{\mathbf{c}}_i^{*s}) \quad \text{and} \quad D_2(\mathbf{y}^s) = \min_{1 \leq i \leq n_s} d(\mathbf{y}^{*s}, \hat{\mathbf{c}}_i^{*s}),$$

where $\mathbf{y}^s$ denotes a generated curve and $\mathbf{y}^{*s}$ its standardized shape, with $s \in \{b, f\}$.

A low value of $D_1(\mathbf{y}^s)$ indicates that the generated curve is, on average, close to the shapes in the dataset, while a low value of $D_2(\mathbf{y}^s)$ indicates that the generated curve is close to at least one observed shape. The values of these two metrics are reported above each generated curve in Figures 13 and 14.

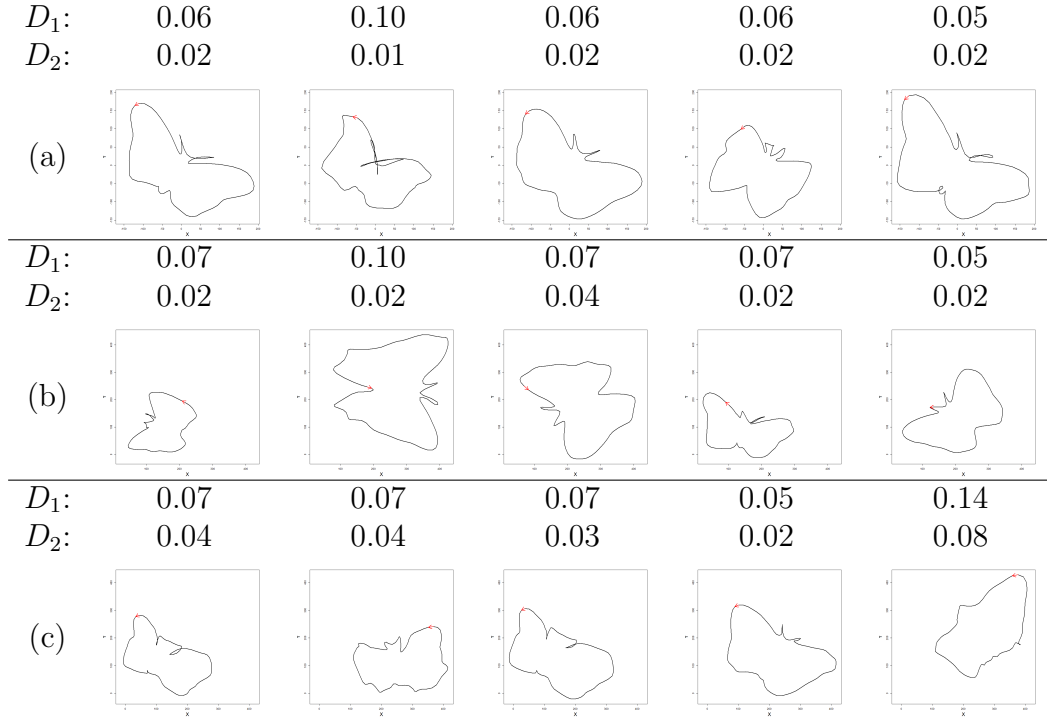


Figure 13: Butterfly contours generated with (a) our approach without the deformation parameters, (b) our approach with the deformation parameters, and (c) with MFPCA.

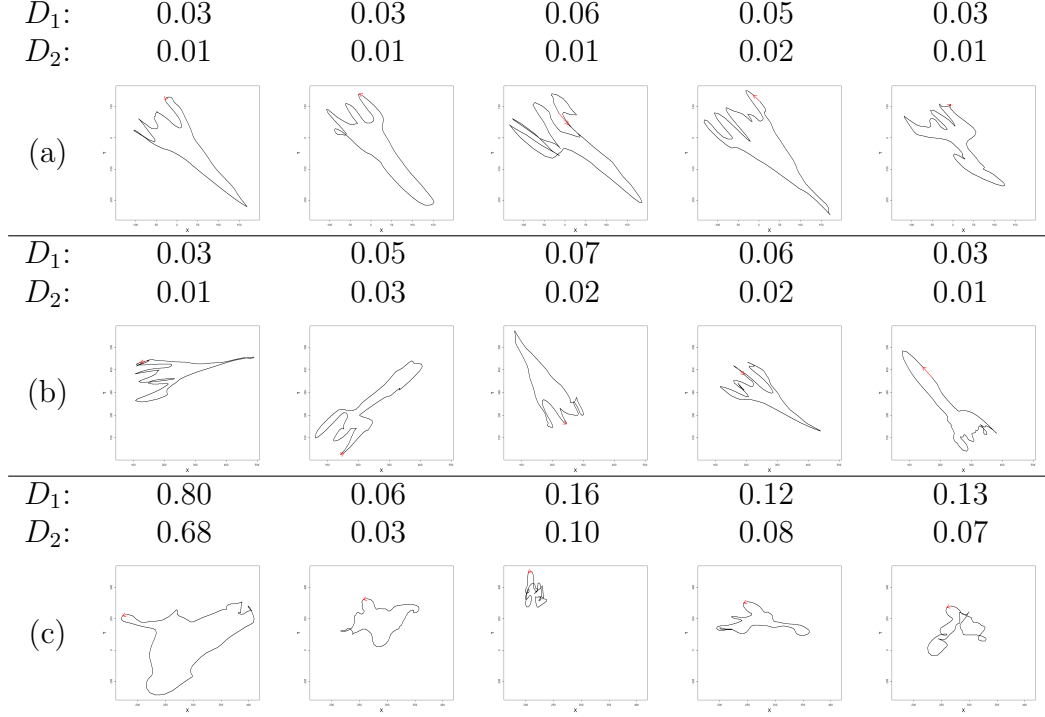


Figure 14: Fork contours generated with (a) our approach without the deformation parameters, (b) our approach with the deformation parameters, and (c) with MFPCA

For the butterfly dataset, illustrated in Figure 13, both the proposed approach and the MFPCA baseline yield comparable performance according to the proposed metrics. However, visual inspection favors our approach. In particular, the generated shapes better preserve fine structural features such as the head and antennae, which are often poorly represented or entirely absent in curves generated using MFPCA.

For the fork dataset, illustrated in Figure 14, the metrics clearly favor the proposed approach. The shapes generated using MFPCA show little resemblance to actual forks, reflecting the limitations of modeling unaligned curves. In contrast, the proposed method produces shapes that remain consistent with the overall fork geometry. Nevertheless, some generated curves still exhibit irregular or distorted features. This highlights the intrinsic difficulty of modeling complex shapes from a limited sample size ($n_f = 20$).

6 Discussion

This paper studies random planar curves, with a particular focus on contours automatically extracted from images. The main contribution of this work lies in the proposed alignment framework. We introduce a functional approach that explicitly models deformation variables and focuses on the dominant source of phase variability in contour data, namely the arbitrary choice of the starting point. This leads to a tractable and fully functional alignment procedure, referred to as the ICF algorithm, which avoids additional discretization and yields interpretable estimates of the deformation parameters.

In contrast to fully elastic alignment frameworks, which rely on highly flexible reparametrizations, our approach deliberately restricts the class of admissible transformations. This choice is motivated by the nature of image-derived contours, where the primary source of misalignment arises from the parametrization rather than from complex local deformations. By focusing on this setting, the proposed method provides a simple, identifiable, and computationally efficient alternative that is particularly well suited for automatically extracted shapes.

Beyond alignment, we propose a generative model for random planar curves within a functional data analysis framework. This model is based on two separate principal component analyses: one performed on the aligned functional variable and the other on the estimated deformation parameters. A key advantage of this formulation is that it enables the analysis of shape variability and deformation effects, rather than treating the latter as nuisance quantities.

The effectiveness of the proposed framework is demonstrated through both simulation studies and real data experiments. The results of the simulation study highlight the accuracy of the alignment procedure and its robustness to discretization. The real data analysis further emphasizes the importance of alignment: the estimated shapes are coherent, and the proposed model generates realistic curves that better reflect the geometric structure of the data than approaches based on unaligned functional representations.

A key feature of our approach is the explicit inclusion of deformation variables in the modeling framework, in particular the scaling parameter. This allows us to work within a linear functional space of closed curves. In contrast, treating scaling as a nuisance parameter would require working on the unit sphere $\mathbf{S}^\infty$, which is a non-linear manifold, and adapting tools from shape analysis such as tangent space approximations (Dryden and Mardia, 1998; Dai and Müller, 2018).

Overall, this work highlights the importance of tailoring alignment procedures to the structure of the data. By focusing on the specific characteristics of contours extracted from images, we obtain a simple yet effective framework that bridges functional data analysis, shape analysis and image analysis.

Our work also has some limitations. First, we focus on single-object contours, which may be restrictive in applications involving multiple objects. Extending the framework to handle multiple interacting curves is a natural direction for future research. Second, the class of reparametrization functions considered in this paper is restricted to shifts of the starting point. As already mentioned, our framework is intentionally designed for contours extracted from images, where the dominant source of phase variability arises from the arbitrary choice of the starting point. While this setting motivates the use of simple reparametrization functions, the proposed approach could be extended to incorporate more flexible transformations when required. In particular, integrating richer classes of reparametrizations, such as diffeomorphic transformations, constitutes a natural direction for future work.

References

- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Carlier, A., Leonard, K., Hahmann, S., Morin, G., and Collins, M. (2016). The 2d shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30.
- Dai, X. (2022). Statistical inference on the hilbert sphere with application to random densities. *Electronic Journal of Statistics*, 16(1):700–736.
- Dai, X. and Müller, H.-G. (2018). Principal component analysis for functional data on riemannian manifolds and spheres. *The Annals of Statistics*, 46(6B):3334–3362.
- Dryden, I. L. (2023). *shapes: Statistical Shape Analysis*. R package version 1.2.7.
- Dryden, I. L. and Mardia, K. V. (1998). *Statistical analysis of shape*. Wiley.
- Happ, C. and Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522):649–659.
- Happ, C., Scheipl, F., Gabriel, A.-A., and Greven, S. (2019). A general framework for multivariate functional principal component analysis of amplitude and phase variation. *Stat*, 8(1):e220.
- Jacques, J. and Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.
- Kendall, D. G. (1989). A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99.
- Marron, J. S., Ramsay, J. O., Sangalli, L. M., and Srivastava, A. (2015). Functional data analysis of amplitude and phase variation. *Statistical Science*, pages 468–484.
- nl, C. (2020). Flying grey long eared bat isolated on white background stock photo. <https://www.istockphoto.com/photo/>.
- R Core Team (2023). *The R Stats Package*. R package version 4.3.0.
- Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer New York, 2nd edition.
- Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.
- Srivastava, A., Klassen, E., Joshi, S. H., and Jermyn, I. H. (2010). Shape analysis of elastic curves in euclidean spaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(7):1415–1428.

- Srivastava, A. and Klassen, E. P. (2016). *Functional and shape data analysis*, volume 1. Springer.
- Stöcker, A., Steyer, L., and Greven, S. (2023). Functional additive models on manifolds of planar shapes and forms. *Journal of Computational and Graphical Statistics*, 32(4):1600–1612.
- Tucker, J. D. (2026). *fdasrvf: Elastic Functional Data Analysis*. R package version 2.4.3.
- Tucker, J. D., Wu, W., and Srivastava, A. (2013). Generative models for functional data using phase and amplitude separation. *Computational Statistics & Data Analysis*, 61:50–66.
- Weisstein, E. W. (2003). Heart curve. <https://mathworld.wolfram.com/>.
- Younes, L. (1998). Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics*, 58(2):565–586.
- Zhang, Z. (2021). Iterative closest point (icp). In *Computer vision: a reference guide*, pages 718–720. Springer.

A Additional figures

This section presents additional results obtained from the analysis of the bat, horseshoe, and spoon datasets introduced in Section 5. The same analysis pipeline as for the butterfly and fork datasets was applied, following the alignment and modeling procedures described in Sections 5.1 and 5.2.

The alignment results are displayed in Figure 15. Overall, the estimated shapes and their associated coordinate functions appear to be well aligned with the reference function $\hat{\mu}$. The spoon dataset is particularly interesting, as the bowl of the spoon does not always appear on the same side of the contour. This illustrates a potential identifiability issue for certain shapes. Nevertheless, the corresponding coordinate functions remain visually well aligned.

The modeling results for $\mathbf{C}$ are presented in Figures 16, 17, and 18. Compared with the standard multivariate functional PCA (MFPCA) approach, the proposed method generally generates more realistic shapes, particularly for the bat and horseshoe datasets.

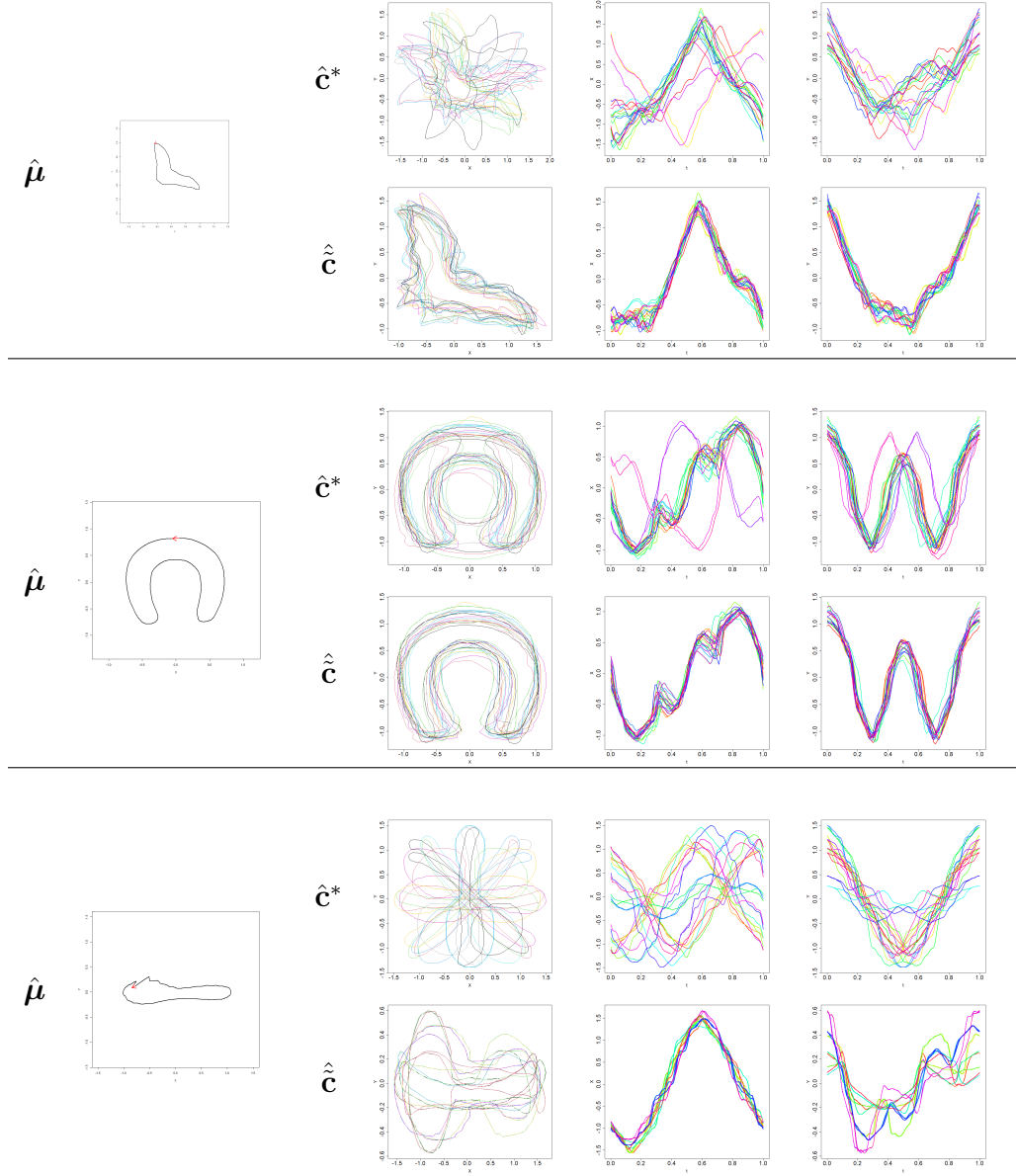


Figure 15: Illustration of the alignment procedure for the bat dataset (top block), the horseshoe dataset (second block) and the spoon dataset (bottom block). In each block, the left column displays the estimated Fréchet mean. The first row shows estimated pre-shapes together with their coordinate functions, while the second row presents the corresponding aligned shapes.

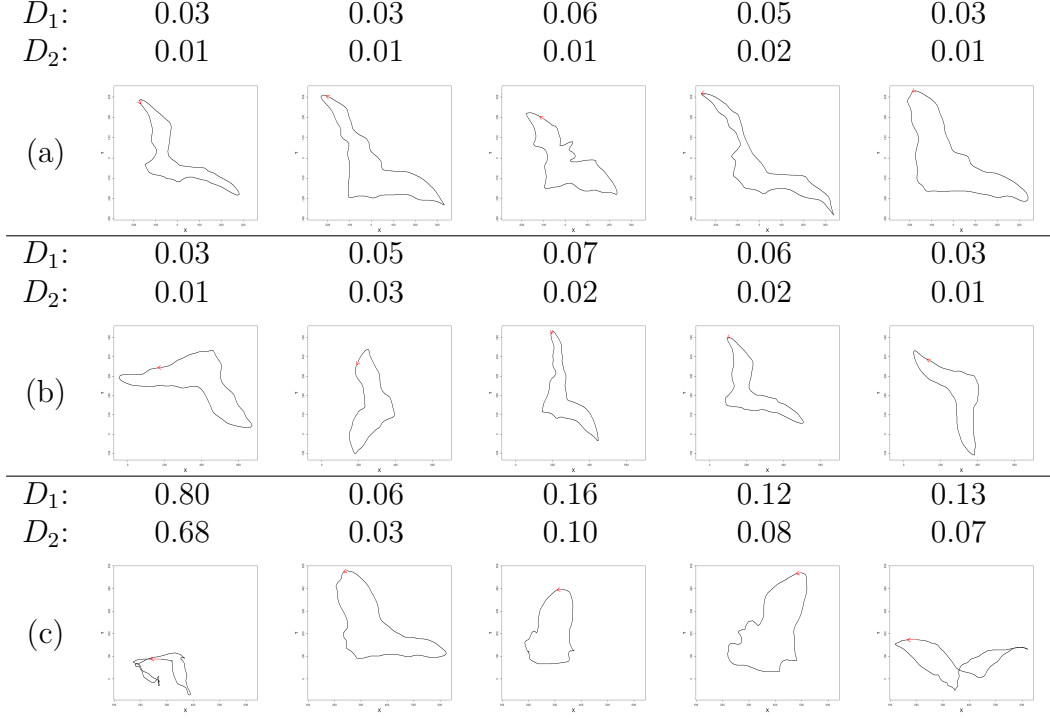


Figure 16: Bat contours generated with (a) our approach without the deformation parameters, (b) our approach with the deformation parameters, and (c) with MFPCA.

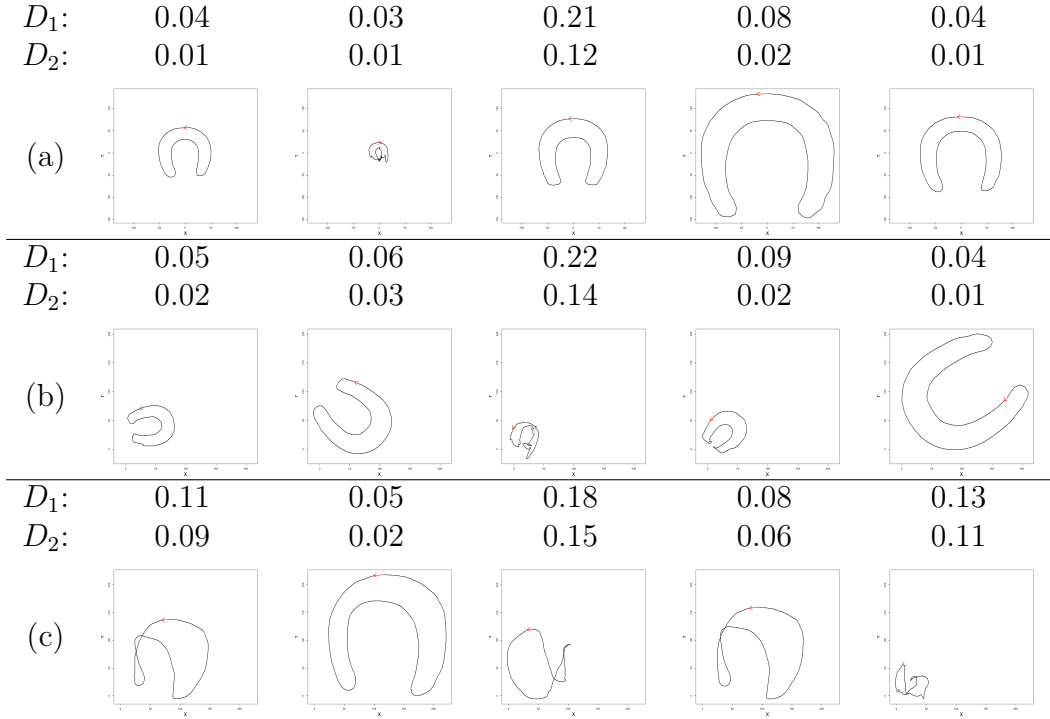


Figure 17: Horseshoe contours generated with (a) our approach without the deformation parameters, (b) our approach with the deformation parameters, and (c) with MFPCA.

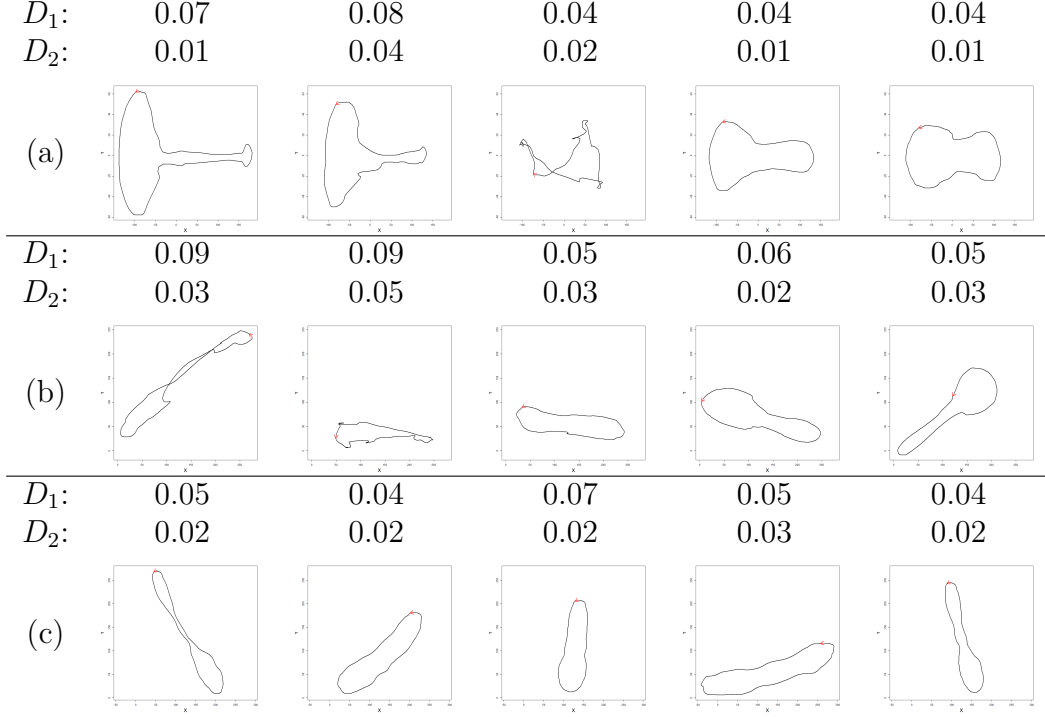


Figure 18: Spoon contours generated with (a) our approach without the deformation parameters, (b) our approach with the deformation parameters, and (c) with MFPCA.

B Proofs of the formal results

Proof of Proposition (2.1). The proposition is composed of two parts: (i) $(\Gamma, \circ)$ is a group, and (ii) the isometry propriety holds.

Proof of (i): First recall that for $r \in \mathbb{R}$, we have that $\text{mod}(r, 1) = r - \lfloor r \rfloor \equiv \{r\}$, i.e., that $\{r\}$ is the fractional part of r . We use the following properties of the fractional part in the proof: (a) $\{\{r\}\} = \{r\}$, (b) $\{\{r_1\} + r_2\} = \{r_1 + r_2\}$, (c) $\{r + 1\} = \{r\}$ for $r, r_1, r_2 \in \mathbb{R}$, (d) $\{r\} = r - 1$ for $1 \leq r < 2$ and (e) $\{r\} = r + 1$ for $-1 \leq r < 0$.

The group structure is a consequence of the following lemma.

Lemma B.1. *Let $\gamma_{\delta_1}, \gamma_{\delta_2} \in \Gamma$, then*

$$\gamma_{\delta_1} \circ \gamma_{\delta_2} = \gamma_{\tilde{\delta}} \in \Gamma, \tag{20}$$

with $\tilde{\delta} = \text{mod}(\delta_1 + \delta_2, 1)$.

Proof. First note that by definition $\gamma_0 = \text{mod}(t, 1) = \text{mod}(t - 1, 1) = \gamma_1$.

- **Case where $\delta_1 \in \{0, 1\}$ and/or $\delta_2 \in \{0, 1\}$.** If $\delta_1 = 0$ and $\delta_2 \in [0, 1]$, then for any

$t \in [0, 1]$, we have

$$\begin{aligned}
\gamma_{\delta_1} \circ \gamma_{\delta_2}(t) &= \gamma_0 \circ \gamma_{\delta_2}(t) = \text{mod}(\gamma_{\delta_2}(t), 1) = \text{mod}(\text{mod}(t - \delta_2, 1), 1) \\
&= \text{mod}(t - \delta_2, 1) \\
&= \begin{cases} \gamma_{\delta_2} & \text{if } \delta_2 < 1 \\ \gamma_0 & \text{if } \delta_2 = 1 \end{cases} \\
&= \gamma_{\text{mod}(0+\delta_2, 1)}.
\end{aligned}$$

All the other cases are shown with a similar argument.

- **Case where $\delta_1 \in (0, 1)$ and $\delta_2 \in (0, 1)$.** For any $t \in [0, 1]$ we have

$$\gamma_{\delta_2}(t) = \begin{cases} 1 + (t - \delta_2) & \text{if } t - \delta_2 < 0, \\ (t - \delta_2) & \text{if } t - \delta_2 \geq 0, \end{cases}$$

which leads to

$$\begin{aligned}
\gamma_{\delta_1} \circ \gamma_{\delta_2}(t) &= \text{mod}(\gamma_{\delta_2}(t) - \delta_1, 1) \\
&= \begin{cases} \text{mod}(1 + (t - \delta_2) - \delta_1, 1) & \text{if } t - \delta_2 < 0 \\ \text{mod}((t - \delta_2) - \delta_1, 1) & \text{if } t - \delta_2 \geq 0 \end{cases} \\
&= \text{mod}(t - (\delta_2 + \delta_1), 1) \\
&= \text{mod}(t - \tilde{\delta}, 1).
\end{aligned}$$

The last equality comes from the fact that

$$\delta_1 + \delta_2 = \begin{cases} \tilde{\delta} & \text{if } \delta_1 + \delta_2 < 1 \\ \tilde{\delta} + 1 & \text{if } \delta_1 + \delta_2 \geq 1 \end{cases}, \quad \text{and} \quad \text{mod}(t - \tilde{\delta}, 1) = \text{mod}(t - (\tilde{\delta} + 1), 1).$$

Since by definition $\tilde{\delta} \in [0, 1]$, we finally have that $\gamma_{\delta_1} \circ \gamma_{\delta_2}(t) = \text{mod}(t - \tilde{\delta}, 1) = \gamma_{\tilde{\delta}}(t)$, with $\gamma_{\tilde{\delta}} \in \Gamma$.

□

The proof that $(\Gamma, \circ)$ is a group follows directly from Lemma B.1 :

- **Associativity:** Let $\gamma_{\delta_1}, \gamma_{\delta_2}, \gamma_{\delta_3} \in \Gamma$, then we have

$$\begin{aligned}
(\gamma_{\delta_1} \circ \gamma_{\delta_2}) \circ \gamma_{\delta_3} &= \gamma_{\text{mod}(\delta_1+\delta_2, 1)} \circ \gamma_{\delta_3} \\
&= \gamma_{\text{mod}(\text{mod}(\delta_1+\delta_2, 1)+\delta_3, 1)} \\
&= \gamma_{\text{mod}(\delta_1+\text{mod}(\delta_2+\delta_3, 1), 1)} \\
&= \gamma_{\delta_1} \circ (\gamma_{\delta_2} \circ \gamma_{\delta_3}),
\end{aligned}$$

where we used property (b) of fractional parts to obtain the third equality.

- **Identity element:** The function $\gamma_0 \in \Gamma$ is the identity element since for any $\gamma_{\delta} \in \Gamma$, we have

$$\gamma_{\delta} \circ \gamma_0 = \gamma_0 \circ \gamma_{\delta} = \gamma_{\text{mod}(\delta+0, 1)} = \gamma_{\delta}.$$

- **Inverse element:** Let $\gamma_{\delta} \in \Gamma$ then $\gamma_{1-\delta} \in \Gamma$ is the inverse of γ_{δ} since

$$\gamma_{\delta} \circ \gamma_{1-\delta} = \gamma_{\text{mod}(\delta+(1-\delta), 1)} = \gamma_0 \quad \text{and} \quad \gamma_{1-\delta} \circ \gamma_{\delta} = \gamma_{\text{mod}(1-\delta)+\delta, 1)} = \gamma_0.$$

This concludes the proof of (i).

Proof of (ii): Let $\mathbf{f} = (f^{(1)}, f^{(2)})^\top \in \mathcal{H}$ and $\gamma_\delta \in \Gamma$, we have that

$$\|\mathbf{f} \circ \gamma_\delta\|_2^2 = \int_0^1 (f^{(1)} \circ \gamma_\delta(t))^2 dt + \int_0^1 (f^{(2)} \circ \gamma_\delta(t))^2 dt,$$

where for $j \in \{1, 2\}$ and $t \in (0, 1)$, we have

$$f^{(j)} \circ \gamma_\delta(t) = \begin{cases} f^{(j)}(1+t-\delta) & \text{if } t-\delta < 0, \\ f^{(j)}(t-\delta) & \text{if } t-\delta \geq 0. \end{cases}$$

Then

$$\begin{aligned} \int_0^1 (f^{(j)} \circ \gamma_\delta(t))^2 dt &= \int_0^\delta (f^{(j)}(1+t-\delta))^2 dt + \int_\delta^1 (f^{(j)}(t-\delta))^2 dt \\ &= \int_{1-\delta}^1 (f^{(j)}(u))^2 du + \int_0^{1-\delta} (f^{(j)}(u))^2 du \\ &= \int_0^1 (f^{(j)}(u))^2 du, \end{aligned}$$

which leads to

$$\|\mathbf{f} \circ \gamma_\delta\|_2^2 = \int_0^1 (f^{(1)}(u))^2 du + \int_0^1 (f^{(2)}(u))^2 du = \|\mathbf{f}\|_2^2.$$

□

Lemma B.2. $d(\cdot, \cdot)$ is a pseudo-distance on $\mathcal{H}$.

Proof of Lemma B.2. Let $\mathbf{f}, \mathbf{g} \in \mathcal{H}$.

- **Positivity:** We have by definition

$$d(\mathbf{f}, \mathbf{g}) = \min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{f} \circ \gamma_\delta - \mathbf{O}_\theta \mathbf{g}\|_{\mathcal{H}} \geq 0,$$

and

$$d(\mathbf{f}, \mathbf{f}) = \min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{f} \circ \gamma_\delta - \mathbf{O}_\theta \mathbf{f}\|_{\mathcal{H}} = \|\mathbf{f} \circ \gamma_0 - \mathbf{O}_0 \mathbf{f}\|_{\mathcal{H}} = 0.$$

- **Symmetry:** First note that

$$d(\mathbf{f}, \mathbf{g}) = \min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{f} \circ \gamma_\delta - \mathbf{O}_\theta \mathbf{g}\|_{\mathcal{H}} = \min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{O}_{2\pi-\theta} \mathbf{f} - \mathbf{g} \circ \gamma_{1-\delta}\|_{\mathcal{H}}.$$

Letting $\theta' = 2\pi - \theta$ and $\delta' = 1 - \delta$, we have that

$$d(\mathbf{f}, \mathbf{g}) = \min_{\delta' \in [0,1], \theta' \in [0,2\pi]} \|\mathbf{O}_{\theta'} \mathbf{f} - \mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}} = d(\mathbf{g}, \mathbf{f}).$$

- **Triangle inequality:**

Since $\|\cdot\|_{\mathcal{H}}$ induce a proper distance, we have that

$$\|\mathbf{f}_1 - \mathbf{f}_2\|_{\mathcal{H}} + \|\mathbf{f}_2 - \mathbf{f}_3\|_{\mathcal{H}} \geq \|\mathbf{f}_1 - \mathbf{f}_3\|_{\mathcal{H}}$$

for $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3 \in \mathcal{H}$.

For any $\mathbf{f}, \mathbf{r}, \mathbf{g} \in \mathcal{H}$, by letting $\mathbf{f}_1 = \mathbf{f} \circ \gamma_{\delta}$, $\mathbf{f}_2 = \mathbf{O}_{\theta}\mathbf{r}$ and $\mathbf{f}_3 = \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}$ in the previous inequality, we obtain

$$\begin{aligned} & \|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}} + \|\mathbf{O}_{\theta}\mathbf{r} - \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}} \geq \|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}} \\ & \Rightarrow \min_{\delta, \delta' \in [0,1], \theta, \theta' \in [0,2\pi]} (\|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}} + \|\mathbf{O}_{\theta}\mathbf{r} - \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}}) \\ & \geq \min_{\delta, \delta' \in [0,1], \theta, \theta' \in [0,2\pi]} \|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}}. \end{aligned} \quad (21)$$

The left-hand side of the inequality (21) admits the following simplification

$$\begin{aligned} & \min_{\delta, \delta' \in [0,1], \theta, \theta' \in [0,2\pi]} (\|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}} + \|\mathbf{O}_{\theta}\mathbf{r} - \mathbf{O}_{\theta'}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}}) \\ & = \min_{\delta, \delta' \in [0,1], \theta, \theta' \in [0,2\pi]} (\|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}} + \|\mathbf{r} - \mathbf{O}_{\theta'-\theta}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}}) \\ & = \min_{\delta \in [0,1], \theta \in [0,2\pi]} (\|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}}) + \min_{\delta' \in [0,1], \theta^* \in [0,2\pi]} (\|\mathbf{r} - \mathbf{O}_{\theta^*}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}}), \end{aligned}$$

where

$$\min_{\delta \in [0,1], \theta \in [0,2\pi]} \|\mathbf{f} \circ \gamma_{\delta} - \mathbf{O}_{\theta}\mathbf{r}\|_{\mathcal{H}} = d(\mathbf{f}, \mathbf{r}),$$

and

$$\min_{\delta' \in [0,1], \theta^* \in [0,2\pi]} \|\mathbf{r} - \mathbf{O}_{\theta^*}\mathbf{g} \circ \gamma_{\delta'}\|_{\mathcal{H}} = \min_{\delta' \in [0,1], \theta^* \in [0,2\pi]} \|\mathbf{r} \circ \gamma_{1-\delta'} - \mathbf{O}_{\theta^*}\mathbf{g}\|_{\mathcal{H}} = d(\mathbf{r}, \mathbf{g}).$$

Using a similar reasoning, we can show that the right-hand side of (21) is equal to $d(\mathbf{f}, \mathbf{g})$, and thus that $d(\mathbf{f}, \mathbf{r}) + d(\mathbf{r}, \mathbf{g}) \geq d(\mathbf{f}, \mathbf{g})$.

□

Proof of Lemma (2.2). Due to the block structure of the matrix $\beta(\delta)$, the lemma follows directly from the fact that

$$\begin{pmatrix} \sin(2m\pi\gamma_{\delta}(t)) \\ \cos(2m\pi\gamma_{\delta}(t)) \end{pmatrix} = \mathbf{O}_{-2m\pi\delta} \begin{pmatrix} \sin(2m\pi t) \\ \cos(2m\pi t) \end{pmatrix}, \quad (22)$$

for $m \in \mathbb{N}^*$ and $\gamma_{\delta} \in \Gamma$, which we prove below.

First note that for $t \in [0, 1]$,

$$\sin(2m\pi\gamma_{\delta}(t)) = \begin{cases} \sin(2m\pi(1 + (t - \delta))) & t \leq \delta \\ \sin(2m\pi((t - \delta))) & t > \delta, \end{cases}$$

where $\sin(2m\pi(1 + (t - \delta))) = \sin(2\pi(t - \delta) + 2m\pi) = \sin(2\pi(t - \delta))$. Hence

$$\sin(2m\pi\gamma_\delta(t)) = \sin(2m\pi(t - \delta)).$$

The same reasoning shows that

$$\cos(2m\pi\gamma_\delta(t)) = \cos(2m\pi(t - \delta)).$$

Finally,

$$\begin{pmatrix} \sin(2m\pi\gamma_\delta(t)) \\ \cos(2m\pi\gamma_\delta(t)) \end{pmatrix} = \begin{pmatrix} \sin(2m\pi t - 2m\pi\delta) \\ \cos(2m\pi t - 2m\pi\delta) \end{pmatrix} = \mathbf{O}_{-2m\pi\delta} \begin{pmatrix} \sin(2m\pi t) \\ \cos(2m\pi t) \end{pmatrix},$$

which concludes the proof. $\square$

Proof of Proposition (2.3). Let $\mathbf{f}_{\theta,\delta} = \mathbf{C} - \mathbf{O}_\theta \boldsymbol{\mu} \circ \gamma_\delta$, this function admits the following representation

$$\mathbf{f}_{\theta,\delta} = \begin{pmatrix} \boldsymbol{\psi}^\top & 0 \\ 0 & \boldsymbol{\psi}^\top \end{pmatrix} \vec{\boldsymbol{\alpha}}_{\theta,\delta},$$

with

$$\vec{\boldsymbol{\alpha}}_{\theta,\delta} = \text{Vec} \left((\boldsymbol{\alpha}\boldsymbol{\beta}(\delta) - \mathbf{O}_\theta \mathbf{u})^\top \right),$$

and where $\text{Vec}(\cdot)$ is the vectorization operator. Hence the norm of $\mathbf{f}_{\theta,\delta}$ is given by

$$\begin{aligned} \|\mathbf{f}_{\theta,\delta}\|_{\mathcal{H}}^2 &= \int_0^1 \mathbf{f}_{\theta,\delta}^\top(t) \mathbf{f}_{\theta,\delta}(t) dt \\ &= \vec{\boldsymbol{\alpha}}_{\theta,\delta}^\top \underbrace{\left[\int_0^1 \begin{pmatrix} \boldsymbol{\psi}(t) & 0 \\ 0 & \boldsymbol{\psi}(t) \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi}^\top(t) & 0 \\ 0 & \boldsymbol{\psi}^\top(t) \end{pmatrix} dt \right]}_{\mathbf{I}} \vec{\boldsymbol{\alpha}}_{\theta,\delta} \\ &= \vec{\boldsymbol{\alpha}}_{\theta,\delta}^\top \vec{\boldsymbol{\alpha}}_{\theta,\delta} \\ &= \|\vec{\boldsymbol{\alpha}}_{\theta,\delta}\|_2^2 \\ &= \left\| \text{Vec} \left((\boldsymbol{\alpha}\boldsymbol{\beta}(\delta) - \mathbf{O}_\theta \mathbf{u})^\top \right) \right\|_2^2 \\ &= \|\boldsymbol{\alpha}\boldsymbol{\beta}(\delta) - \mathbf{O}_\theta \mathbf{u}\|_F^2, \end{aligned}$$

which concludes the proof. $\square$

Proof. The equation (14) can be seen as a constrained Procrustes problem, since the orthogonal matrix $\boldsymbol{\beta}(-\delta)$ has a fixed form. Let note $\mathbf{u}^\theta = \mathbf{O}_\theta \mathbf{u}$, we have

$$\hat{\delta} = \arg \min_{\delta \in [0,1]} \|\boldsymbol{\alpha}\boldsymbol{\beta}(\delta) - \mathbf{u}^\theta\|_F^2 = \arg \min_{\delta \in [0,1]} \|\boldsymbol{\beta}(\delta) - \boldsymbol{\alpha}^\top \mathbf{u}^\theta\|_F^2. \quad (23)$$

Since $\beta(\delta)$ is a sparse orthogonal matrix, Equation (23) reduces to

$$\hat{\delta} = \arg \min_{\delta \in [0,1]} \left(\sum_{k \in \{1,3,\dots,M-1\}} \|\Sigma_{k,\theta} - \mathbf{O}_{\pi\delta(k+1)}\|^2 \right), \quad (24)$$

where

$$\alpha^\top \mathbf{u}^\theta \odot \begin{pmatrix} \mathbf{I}_2 & 0 & \dots & 0 \\ 0 & \mathbf{I}_2 & \dots & 0 \\ \vdots & \dots & & \vdots \\ 0 & 0 & \dots & \mathbf{I}_2 \end{pmatrix} = \begin{pmatrix} \Sigma_{1,\theta} & 0 & \dots & 0 \\ 0 & \Sigma_{3,\theta} & \dots & 0 \\ \vdots & \dots & & \vdots \\ 0 & 0 & \dots & \Sigma_{M-1,\theta} \end{pmatrix},$$

and $\odot$ denotes the Hadamard product.

Regarding the resolution of (24), note that

$$\begin{aligned} \frac{\partial}{\partial \delta} \|\Sigma_{k,\theta} - \mathbf{O}_{\pi\delta(k+1)}\|^2 &= \pi(k+1) (\text{Tr}(\Sigma_{k,\theta}) \sin(\pi(k+1)\delta) \\ &\quad + \text{Tr}(\Sigma_{k,\theta} \mathbf{O}_{\pi/2}) \cos(\pi(k+1)\delta)) \end{aligned}$$

which leads to

$$\begin{aligned} \sum_{k \in \{1,3,\dots,M-1\}} \frac{\partial}{\partial \delta} \|\Sigma_{j,k,\theta} - \mathbf{O}_{\pi\delta(k+1)}\|^2 &= \pi \sum_{k \in \{1,3,\dots,M-1\}} \left(w_{j,k}^{1,\theta} \sin(\pi(k+1)\delta) \right. \\ &\quad \left. - w_{j,k}^{2,\theta} \cos(\pi(k+1)\delta) \right), \end{aligned}$$

where $w_{j,k}^{1,\theta} = \text{Tr}((k+1)\Sigma_{k,\theta})$, $w_{j,k}^{2,\theta} = \text{Tr}(-(k+1)\Sigma_{k,\theta} \mathbf{O}_{\pi/2})$.

Therefore, the solution $\hat{\delta}_j$ of (14) belongs to $\mathcal{S}_M$, where

$$\mathcal{S}_M = \left\{ \delta \in [0, 1], \sum_{k \in \{1,3,\dots,M-1\}} w_{j,k}^{1,\theta} \sin(\pi(k+1)\delta) = \sum_{k \in \{1,3,\dots,M-1\}} w_{j,k}^{2,\theta} \cos(\pi(k+1)\delta) \right\}.$$

This concludes the proof. $\square$