Functional Representations of Images for Statistical Learning: Shapes, Surfaces, and Beyond

Cédric Beaulac

Université du Québec à Montréal

August the 9th 2026

Collaborators involved:

Issam-Ali Moindjié
Perpignan

Marie-Hélène Descary
UQAM

Mélanie Raymond
UQAM

Louis-Pierre Ménard
UQAM

Functional Representations of Images for Statistical Learning: Shapes, Surfaces, and Beyond

Image data
Edge and contour detection
Functional data analysis
Functional contour alignement
Shape Analysis
Extension to multivariate planar curves

Image data

Image Data

Images have been natively captured and stored in matrix format since cameras became digital.

The element (i, j) represents the color intensity at pixel $[i, j]$.

For black-and-white or grayscale images, the color intensity is an integer in the range $[0, 255]$.

A color image is typically represented using three matrices with integer entries in $[0, 255]$.

Image Data

For black-and-white or grayscale images, the pixel intensity is an integer in the range $[0, 255]$.

Black-and-white image of the digit '1'

$$\begin{bmatrix} 255 & 255 & 0 & 255 & 255 \\ 255 & 0 & 0 & 255 & 255 \\ 255 & 255 & 0 & 255 & 255 \\ 255 & 255 & 0 & 255 & 255 \\ 255 & 0 & 0 & 0 & 255 \end{bmatrix}$$

Matrix representation (pixel intensities)

Images as Surfaces

We can also think of an image as a surface, where the height at location (i, j) corresponds to the color intensity.

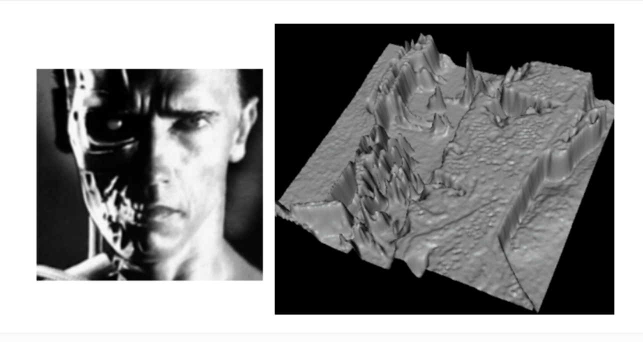


Figure: An image on the left and its associated surface representation on the right.

Image Data Analysis

Typical approaches for image analysis are designed to analyze matrices.

Filtering and convolution are matrix operations that perform linear combinations of neighboring pixels.

Powerful predictive models can then be built by learning convolution weights within broader machine learning frameworks.

Recently, there has been interest in developing **transformer models** for image analysis.

Image Data Analysis

Pixel-based approaches still present several challenges:

- Difficult interpretation,
- Large data objects (high-resolution images and videos),
- Generalization issues (sensitivity to resolution and acquisition technology).

Our Solution

We want to stop looking at images as simple collections of pixels.

Instead, images can be analyzed as collections of objects defined by their shapes, textures, and colors.

Over the last few years, we developed a framework for the analysis of shapes extracted from images.

Edge and contour detection

Edge and contour detection

The first step to analyse shapes in images is to extract them.

For this, we use contour detection techniques.

Edge and contour detection can provide us with flood fill images (or masks).

This is not an easy task.

Edge and contour detection

Assume we have a mask obtained through edge detection.

Photo of a bat.

Flood fill image.

Planar Closed Curves

We extract a functional representation for the contour (a planar closed curve), represented through coordinates:

$$C(t) = (X(t), Y(t)),$$

where $t \in [0, 1]$ represents the proportion of the curve traveled from the start ($t = 0$) to the end ($t = 1$).

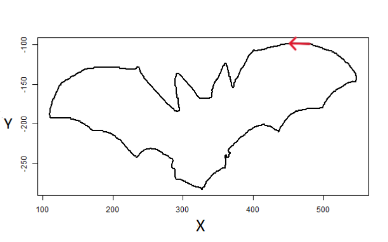
For closed curves, $C(0) = C(1)$.

Traveling along the contour

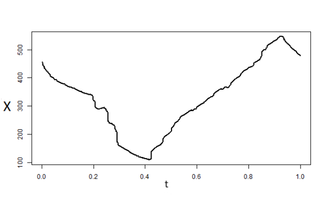
The marching square algorithm (Mantz et al. 2008) will provide us with an ordered sequence of pixels:

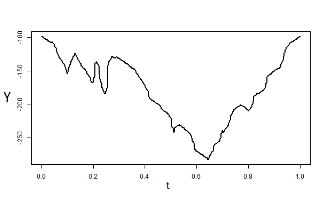
$$[(x[1], y[1]), (x[2], y[2]), ..., (x[T], y[T])] \tag{1}$$

Traveling along the contour



Contour of the bat.





Functional Data Analysis

Functional Data

Functional data analysis (FDA) is a field of statistics concerned with studying data assumed to be realizations of smooth functions,

$$x(t),$$

or higher-dimensional objects such as surfaces,

$$x(s, t).$$

The domain may represent time, space, or more general multidimensional structures.

Functional Data

In FDA, one observation $x_i(t)$ is therefore a function.

In FDA, a dataset consists of a collection of functions

$$S = \{x_i(t) : i = 1, \ldots, n\},$$

where all observations are defined on a common domain T.

In practice, the underlying function cannot be observed continuously and is **only recorded at a finite number of sampling locations**.

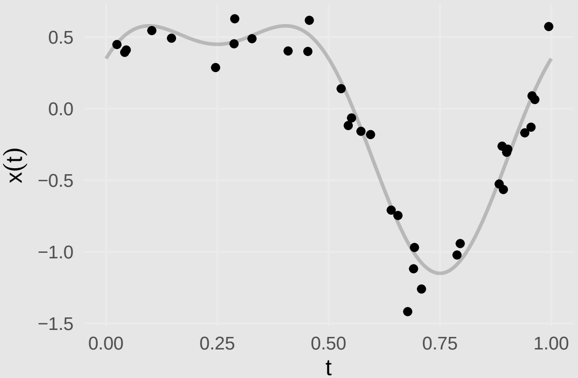


Figure: Discrete observations from an underlying smooth functional process.

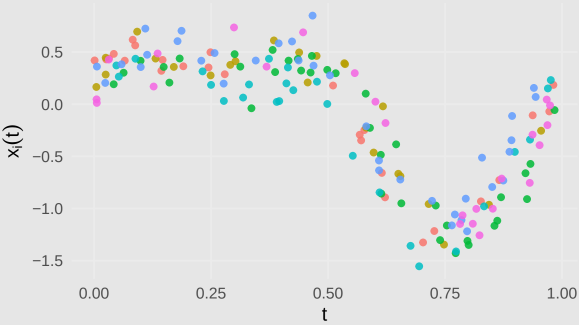


Figure: A sample of functional observations.

Note: The data may be discretely observed on different grids and sampled at a different number of locations.

Smoothing and Functional Representation

FDA commonly assumes that the observed data arise from an underlying smooth process contaminated by measurement error.

Observed measurements are therefore often smoothed before analysis.

Basis Expansions

A standard approach consists of representing functions using basis expansions,

$$\tilde{x}(t) = \sum_{k=1}^{K} c_k B_k(t),$$

where $\{B_k(t)\}$ is a collection of basis functions.

Common choices include:

- Fourier bases,
- B-splines,
- wavelets.

Why Functional Representations?

Smoothing the data to recover a functional representation offers several advantages:

- Noise reduction through smoothing,
- Dimension reduction via basis coefficients,
- Continuous evaluation over the entire domain,
- Natural computation of derivatives,
- Analysis of irregularly sampled data.

Basis Expansions

The representation

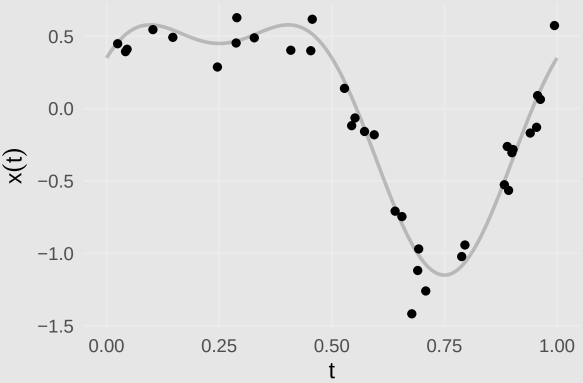
$$\tilde{x}(t) = \sum_{k=1}^{K} c_k B_k(t),$$

provides us with all of those advantages.

Learning such representations is also straightforward and boils down to minimizing a MSE criterion:

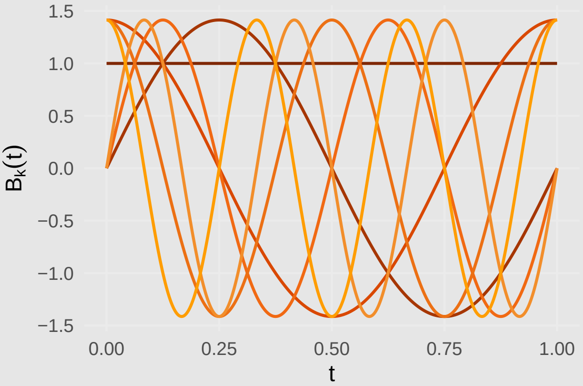
$$\sum_{j} \left[x(t_j) - \sum_{k=1}^{K} c_k B_k(t_j) \right]^2 .$$

Example: Functional Smoothing



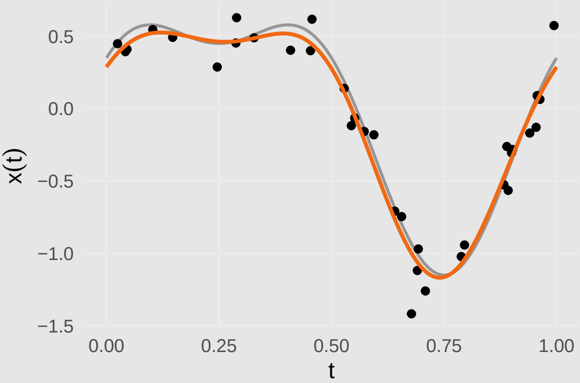
Underlying smooth process.

Example: Fourier Basis



Fourier basis functions.

Example: Smoothed Reconstruction



Recovered smooth functional representation.

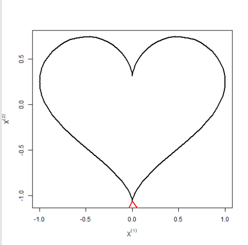
Functional representation of the contour

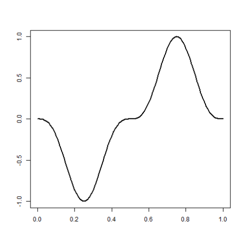
We use basis expansion to smooth the coordinate paths obtained with marching square.

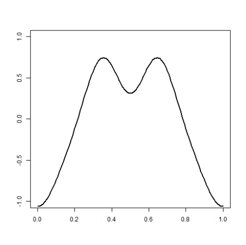
It gives us a smooth, continuous and parametric representation of the contour.

We can use multivariate FDA approaches to solve statistical questions about the shapes.

Functional representation of the contour







Shapes and statitics

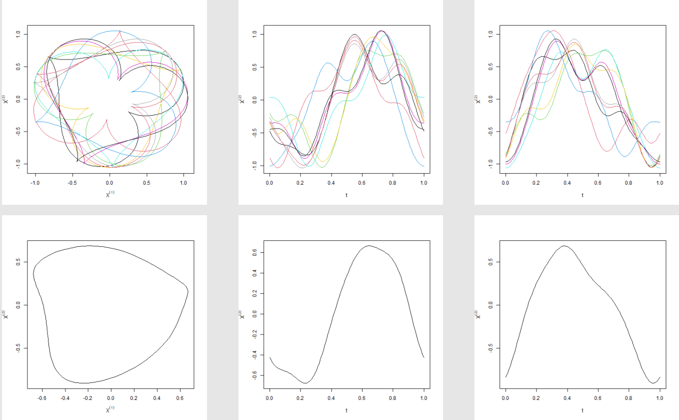
So what about repetition and data sets?

With repetition, different rotations and scales of the same object lead to completely different coordinate functions.

This starting point of the coordinate paths is arbitrary with respect to shape features.

Shapes and statitics

Contours need to be aligned first in order for the statistics to be meaningful.



Functional contour alignement

Functional contour alignment

In order to obtain a sample of shapes, we must first align the contour.
This is because shape is an geometric structure invariante with respect to some deformations.

These are:

- Translation
- Scale
- Rotation
- Path starting point (parameterization)

Our proposed model

The resulting contour we observed is parametrized as:

$$\mathsf{C}(t) = \rho \mathbf{O} \tilde{\mathsf{C}} \circ \gamma(t) + \mathbf{T} \tag{2}$$

where $(\rho, \mathbf{O}, \gamma, \mathbf{T})$ are the deformation parameters and $\tilde{\mathsf{C}}(t)$ the shape.

Functional contour alignement

The alignment procedure we propose estimates these deformations, allowing for their analysis.

It also allows the removal of their effects to analyze what remains: the shape.

Shape as Functional Data

We model closed curves as bivariate functional data.

Smoothing the data solves the problem of images and objects having different resolutions (irregular sampling).

Fourier bases are a natural choice given the cyclical nature of closed curves.

Representationusing Fourrier Basis

First, we represent contours using a Fourrier basis.

$$\mathbf{C}(t) = \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} = \sum_{k=0}^{M} \begin{pmatrix} a_{k1} \\ a_{k2} \end{pmatrix} \psi_k(t), \quad t \in [0,1], \tag{3}$$

with

$$\psi_0(t) = 1, \; \psi_k(t) = \begin{cases} \sqrt{2}\sin((k+1)\pi t) & \text{if } k \text{ odd}, \\ \sqrt{2}\cos(k\pi t) & \text{if } k \text{ even}, \end{cases} \tag{4}$$

for $t \in [0,1]$, $k = 1, \ldots, M$, with M even.

Estimating the scale and translation

Estimating $\mathbf{T}_i = (T_x, T_y)$ and ρ_i is rather simple by assuming the shapes to be centered at $(0, 0)$ and to have unitary norm.

$$\mathbf{T} = \int_0^1 \mathbf{C}(t)dt = \int_0^1 \sum_{k=0}^{M} \boldsymbol{a}_k \psi_k(t)dt = \boldsymbol{a}_0,$$

$$\rho = ||\mathbf{C} - \mathbf{T}||_{\mathcal{H}} = \left\| \sum_{k=1}^{M} \boldsymbol{a}_k \psi_k \right\|_{\mathcal{H}} = \sqrt{\sum_{k=1}^{M} \boldsymbol{a}_k{}_2^2},$$

This is extremely quick since it is given by the functional representation and can be done shape by shape independantly.

Estimating the scale and translation

After we estimate the translation and scale deformation, we obtain $\mathbf{C}^*$, which we call the pre-shape:

$$\mathbf{C}^*(t) = \frac{1}{\rho}(\mathbf{C}(t) - \mathbf{T})$$

Estimating the reparametrization and the rotation

The biggest challenge when developing our proposed approach.

The effects of these deformations are entangled.

We need to estimate both at the same time.

Starting point (reparametrization)

The reparametrization γ is a simple wrapping function that parametrize the effect of different starting point when traveling along the contour.

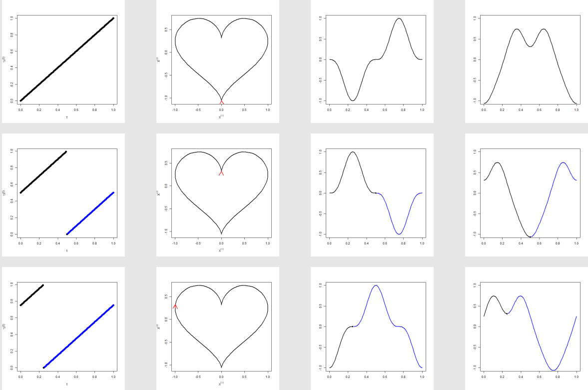
We define $\gamma \in \Gamma$, with

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

Starting point (reparametrization)

we can visualise the effect of this function here:

γ_δ | $\tilde{C} \circ \gamma_\delta$ | $\tilde{x} \circ \gamma_\delta$ | $\tilde{y} \circ \gamma_\delta$



$\delta = 0$

$\delta = 0.5$

$\delta = 0.65$

Reparametrization

Because we analyses closed curves, the coordinate functions are cyclical.

The δ parameter dictate where on the contour did we begin travelling.

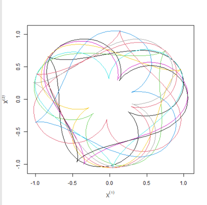
Reparametrization

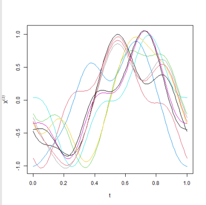
One might think that we can wrap the functions until they are aligned.

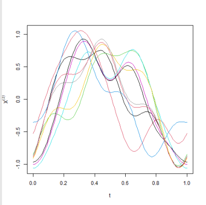
But the coordinate functions are entirely different for different rotations.

These deformations must be estimated jointly.

Effect of the rotation on the coordinate functions







Rotation

The rotation $\mathbf{O}$ of the pre-shape is parametrized with a standard rotational matrix:

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

The problem boils down to estimating θ.

Estimating the reparametrization and the rotation

The first step to align the pre-shapes $\mathbf{C}^*$ is to define a template $\boldsymbol{\mu}$. The reparameterization or rotation do not really matter as long as they are the same for all pre-shapes.

The template can be:

- A random observation $\mathbf{C}_i^*$.
- Some version of the Karcher/Frechet mean.
- A specific observation aligned as desired.

Estimating the reparametrization and the rotation

Given a template $\boldsymbol{\mu}$, we seek to align the pre-shape $\mathbf{C}_i^*$ by finding the parameters δ and θ that aligns the best $\mathbf{C}_i^*$ to $\boldsymbol{\mu}$.

$$(\hat{\theta}, \hat{\delta}) = \underset{(\theta,\delta)\in[0,2\pi]\times[0,1]}{\arg\min} ||\mathbf{O}_\theta \mathbf{C}_i^* \circ \gamma_\delta - \boldsymbol{\mu}||_{\mathcal{H}}^2. \tag{5}$$

Estimating the reparametrization and the rotation

There are no closed-form solutions.

However, we can find a solution for θ given a fixed δ,

and, we can find a solution for δ given a fixed θ.

Estimating the reparametrization and the rotation

We can therefore solve the alignement issues by

- Searching on a grid (for δ) for an optimum value
- **Developping an iterative algorithm (ICF) that updates both parameters every other steps**.

After removing all of the deformation variables; we are left with the shape $\tilde{\mathbf{C}}$.

Estimating the reparametrization and the rotation

After removing all of the deformation variables; we are left with the shape $\tilde{\mathbf{C}}$.

$$\mathbf{C}^*(t) = \frac{1}{\rho}(\mathbf{C}(t) - \mathbf{T})$$

$$\tilde{\mathbf{C}}(t) = \mathbf{O}_\theta \mathbf{C}^*(t) \circ \gamma_{(\delta)}$$

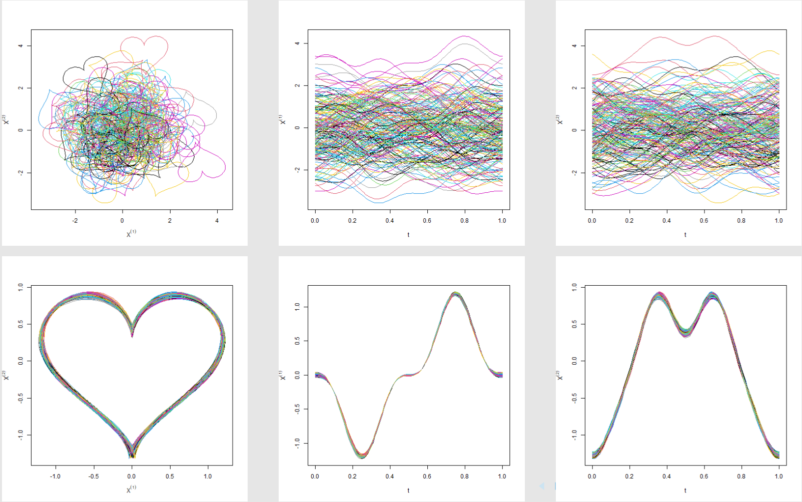
Alignement results

Before we go over the statistical analysis we conduct on shapes; let us make sure the alignement procedure works.

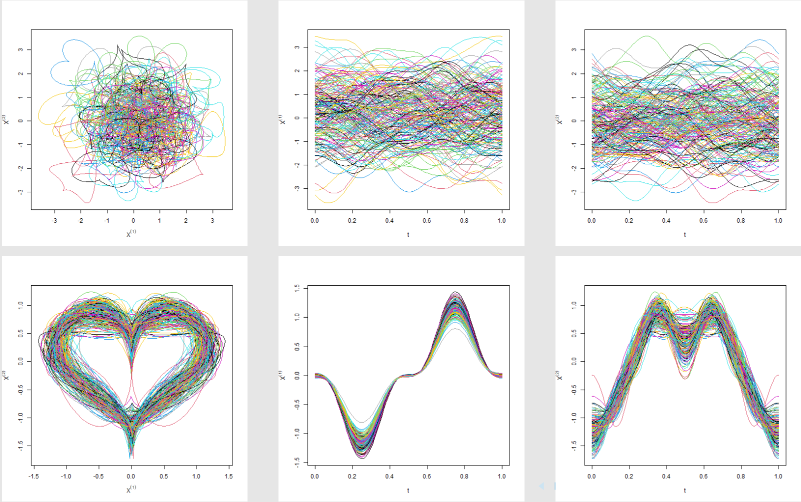
We deformed shapes and realigned them (simulations).

We also aligned real data.

Alignement on simulated data



Alignement on simulated data

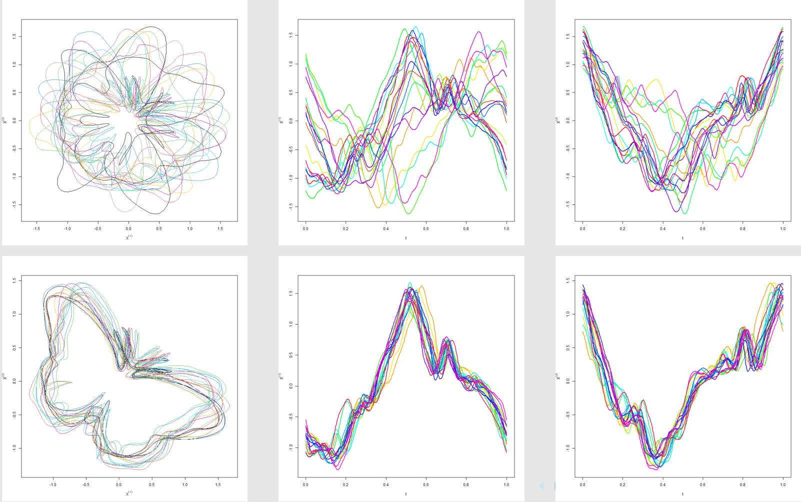


Alignement on real data

MPEG-7 database:

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

Alignement on real data



Shape analysis : applications

Functional shape analysis

- So far, we have not done any *statistics*.
- We needed to prepare the data so that the statistical analysis is meaningful.
- At this point, we can consider multiple statistical problems related to shape and analyze both deformation variables (scalar) and shapes (functional) jointly.

Modeling **X** through a joint PCA approach

We propose a joint Principal Component Analysis (PCA) approach.

We extract features that can be used for both unsupervised and supervised learning problems.

We can consider multiple statistical problems related to shape and analyze both the deformation variables (scalar) and the shape (functional).

Functional Principal component

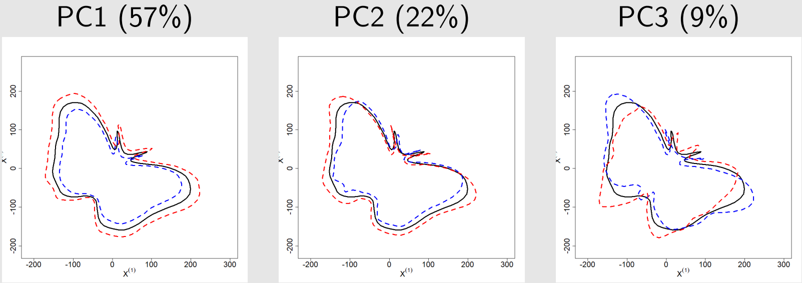


Figure: Plots of the estimated mean function $\bar{\mathbf{z}} = \sum_i \mathbf{z}_{i1}$ in black, of $\bar{\mathbf{z}} - 20\hat{\phi}_k$ in blue and of $\bar{\mathbf{z}} + 20\hat{\phi}_k$ in red, for $k = 1$ (first column), $k = 2$ (second column) and $k = 3$ (third column).

Generation

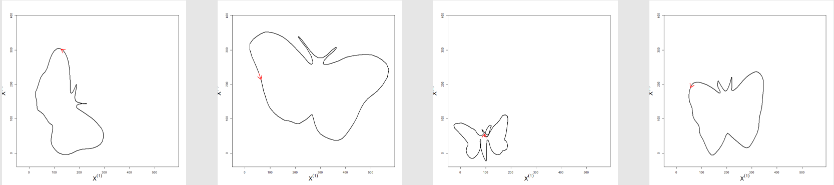


Figure: Butterfly curves generated with our approach with the deformation parameters.

Classification: Brain tumor classification

- BraTS 2020 contains segmentationg masks of brain tumors in multimodal magnetic resonance imaging (MRI) scans.
- We trained shaped-based and pixel-based model on 400 contours of tumors with two labels: high-grade or low-grade glioma.

Classification: exemples

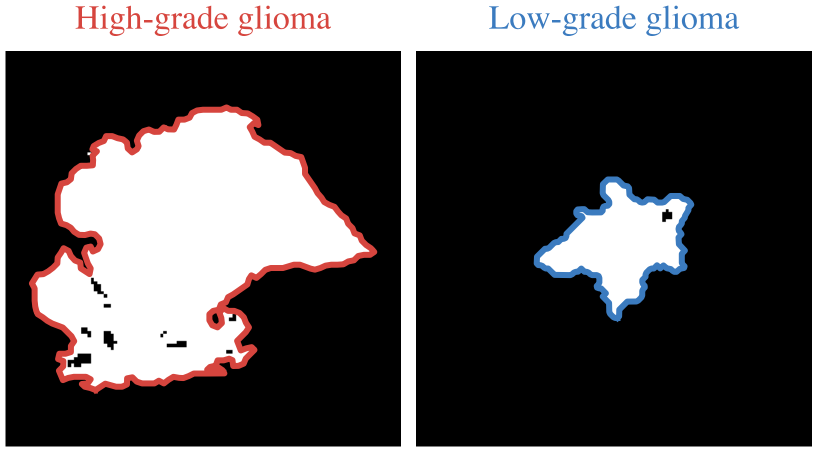


Figure: Exemple of masks.

Classification: Brain tumor classification

Representation	Model	Acc.	Bal.
Pixel	Convolution	76.8 (11.0)	65.9 (4.6)
	Transformer	74.6 (20.3)	63.3 (9.4)
Shape	MLP	**81.4 (6.0)**	**71.5 (6.4)**

Table: BraTS 2020 patient-disjoint cross-validation results, mean (SD) in percent.

Classification: Brain tumor classification

Representation	Model	Params	Train	Infer.
Pixel	Convolution	11.2M	547 s	349.9 ms
	Transformer	5.43M	285 s	180.2 ms
Shape	MLP	7.5k	0.75 s	0.03 ms

Table: Computational metrics over the different models.

Classification: Melanoma detection based on moles shape

- In the previous exemple, we trained models strictly using shape.
- Pixel-based model are trained on the mask and shape based model on the coordinates functions.
- It is fair to assume that a pixel-based model would perform better on coloured images.

Classification: Melanoma detection based on mole shapes

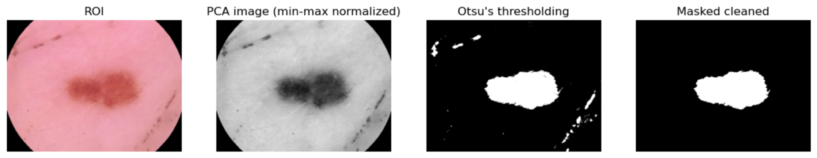


Figure: Segmentation pipeline.

Color parametrized as surface

Images as Surfaces

We can also think of an image as a surface, where the height at location (i, j) corresponds to the color intensity.

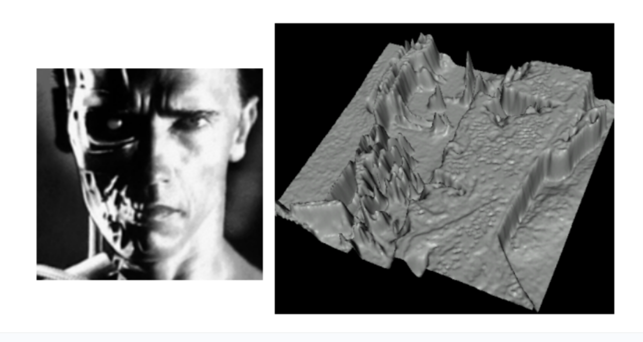


Figure: An image on the left and its associated surface representation on the right.

Motivating problem

It is hypothesised in ecology that the red hourglass on the thorax of black widows spider is varies depending on the spider environement.

We want to jointly study this hourglass shape and color.

I did not include a picture here, arachnophobia is prevelant.

From Curves to Surfaces

In one dimension, a functional observation may be smoothed and represented as

$$\tilde{x}(t) = \sum_{k=1}^{K} c_k B_k(t).$$

The coefficients $c_1, \ldots, c_K$ are learned by minimizing some criterion.

These same ideas naturally extend to higher-dimensional functions such as surfaces

$$x(s, t).$$

Tensor-Product B-Splines

A smooth surface can be represented as

$$\tilde{x}(s,t) = \sum_{k=1}^{K_s} \sum_{\ell=1}^{K_t} \theta_{k\ell} B_k(s) B_\ell(t).$$

Here, $B_k(s)$ and $B_\ell(t)$ are one-dimensional B-spline basis functions along the two spatial directions.

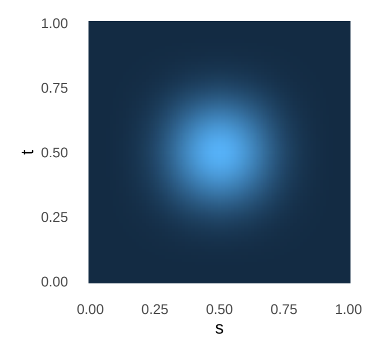
The coefficients $\theta_{k\ell}$ determine the shape of the smooth surface.

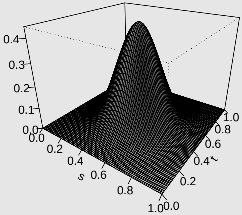
A Single Tensor-Product Basis Function

A tensor-product B-spline basis function

$$B_k(s)B_\ell(t)$$

defines a localized smooth surface over the domain.



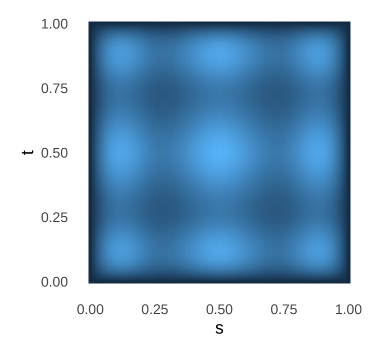


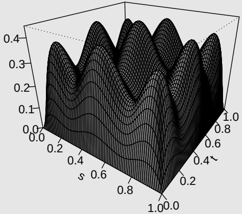
A single tensor-product basis function represented as a heatmap and as a

Combining Tensor-Product Basis Functions

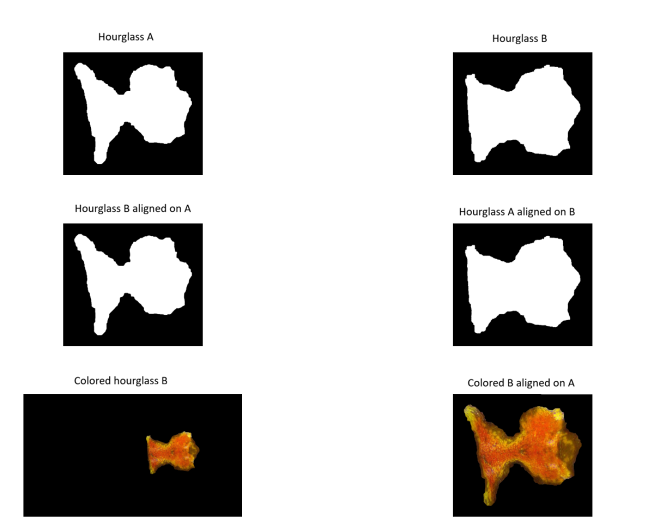
Smooth surfaces are obtained through linear combinations of many localized tensor-product basis functions.

The tensor-product structure naturally extends one-dimensional spline representations to two-dimensional domains.

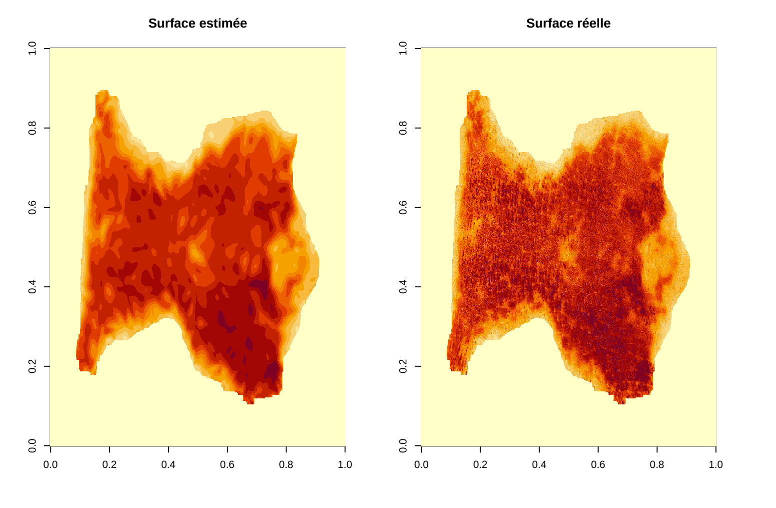




Current development: Images as surface



Current development: Images as surface



Functional Surface Representations for Images

With these tools, we can fit a smooth surface representation of an image.

The pixel grid is viewed as a discrete observation of an underlying smooth surface.

The remaining challenge is therefore to estimate the coefficient matrix Θ, typically by minimizing a mean squared reconstruction criterion over the observed pixels.

Functional Surface Representations for Images

This surface representation offers several advantages:

- continuous image representations allowing instantaneous changes in resolution,
- analytical spatial derivatives that may be useful for image segmentation and contour detection
- noise reduction through smoothing (if desired),
- resolution-independent representations,
- low-dimensional coefficient representations.

Functional Surface Representations for Images

We can leverage the extensive toolbox of functional data analysis to:

- classify or cluster surfaces,
- construct statistical tests to distinguish between populations,
- compute descriptive statistics for populations of surfaces,
- and identify regions of high variability across populations using functional principal component analysis.

Conclusion

Conclusion

Statistical shape analysis can provide new information about images usefull in unsupervised and supervised analysis.

On their own, shapes can provide interpretable information lost in pixel-based approaches.

In order to analyze the underlying shapes of objects, we developed an alignment procedure.

Thus we can also include deformation parameters in analysis.

Current work

We are developping a surface representation for the colors inside objects of interest.

We are developping statistical test for color differences based on those surfaces.

We are also working on a vast R package to cover shape and color analysis end-to-end.

I would love to answer your questions.

I.A. Moindjié, C. Beaulac and M.H. Descary. A Functional Approach for Curve Alignment and Shape Analysis, submitted

I.A. Moindjié, C. Beaulac and M.H. Descary. Statistical Analysis of Multivariate Planar Curves and Applications to X-ray Classification, submitted

J. O. Ramsay and B. W. Silverman. Functional Data Analysis (Second Edition). Springer, 2005.

A. Srivastava and E.P. Klassen, Functional and Shape Data Analysis, Springer, 2016.

H. Mantz, K. Jacobs and K. Mecke, Utilizing Minkowski functionals for image analysis: a marching square algorithm, Journal of Statistical Mechanics: Theory and Experiment, 2008.

Gaggion, Nicolás, Candelaria Mosquera, Lucas Mansilla, Julia Mariel Saidman, Martina Aineseder, Diego H. Milone, and Enzo Ferrante. "CheXmask: a large-scale dataset of anatomical segmentation masks for multi-center chest x-ray images." Scientific Data 11, no. 1 (2024): 511.

Eilers, Paul HC, and Brian D. Marx. Practical smoothing: The joys of P-splines. Cambridge University Press, 2021.

Xiao, Luo, Yingxing Li, and David Ruppert. "Fast bivariate P-splines: the sandwich smoother." Journal of the Royal Statistical Society Series B: Statistical Methodology 75, no. 3 (2013): 577-599.

Alignement on real data: deformations

$\hat{\mathbf{T}}_i$ $\hat{\rho}_i$ $\hat{\theta}_i$ $2\pi\hat{\delta}_i$

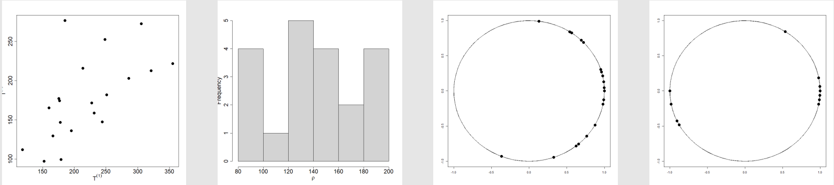


Figure: Plots of the estimated deformation parameters of each curve in both datasets

Alignement on simulated data

σ	MSE_δ	MSE_θ	$\text{MSE}_{\mathbf{T}}$	MSE_ρ
0.01	3.39×10^{-4}	3.41×10^{-4}	9.98×10^{-32}	1.80×10^{-32}
0.1	3.15×10^{-4}	3.15×10^{-4}	6.34×10^{-32}	1.81×10^{-32}

Table: MSE of the estimated parameters for the different scenarios and values of σ

Classification: Melanoma detection based on moles shape

- HAM10000 dataset contains photos of moles.
- We trained a model on 8000 images with two labels: melanoma or belign.
- Our goal was to compare the performance of a shape-based model against a pixel-based one.

Classification: Melanoma detection based on mole shapes

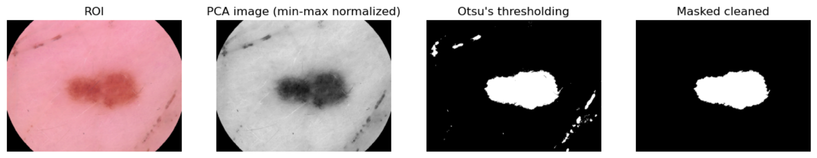


Figure: Segmentation pipeline.

Classification: Melanoma detection based on mole shapes

Model	avg-AUC	avg-Bal.Accuracy	avg-F1
ANN	0.640 ± 0.031	$\mathbf{0.572 \pm 0.036}$	$\mathbf{0.837 \pm 0.020}$
CNN	$\mathbf{0.697 \pm 0.069}$	0.544 ± 0.049	0.814 ± 0.016

Table: Stratified-Nested-CV Results overview.

Classification: Melanoma detection based on mole shapes

Model	VRAM (GB)	avg-Training time (s)
ANN	**0.7**	**31.7** $\pm$ **6.92**
CNN	66.7	174 $\pm$ 82.9

Table: Stratified-Nested-CV Results overview.