From Pixels to Shapes: A Functional Framework for Image Analysis

Cédric Beaulac

Université du Québec à Montréal

June the 23th 2025

The work presented here is the result of a collaboration with postodoctoral fellow Issam-Ali Moindjié and professor Marie-Hélène Descary, both at Université du Québec à Montréal.

From Pixels to Shapes
A Functional Framework for Image Analysis

Image data
Edge and contour detection
Functional basis expansion and smoothing
Functional contour alignement
Functional shape analysis
Applications

Image data

Image data

Images are natively captured and stored in a matrix format since cameras went digital.

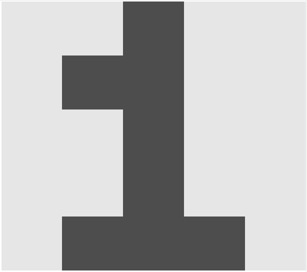
The element (i, j) represents the color intensity at pixel $[i, j]$.

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

For a color image, it is represented with 3 matrices of integer elements in $[0, 255]$.

Image data

For black and white or grayscale images, it 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 color intensity)

Image data analysis

Typical approaches for image analysis are designed to analyze these matrices.

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

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

Image data analysis

Even though these approaches can perform extremely well in predictive tasks, pixel-based approaches have several issues.

- Uninterpretable.
- Multiple hyperparameters that are difficult to adjust.
- Generalization issues.
- Require a large amount of data.

Our solution

We want to stop looking at images as a collection of pixels.

Instead, images are analyzed as a collection of objects, defined by their shapes, textures, and colors.

Our journey begins with shapes: how to extract them from images and how to analyze them.

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

Photo of a bat.

Flood fill image.

Edge and contour detection

Starting with *flood fill* images.

We seek to extract a functional form for the contour (planar closed curve), represented via coordinates:

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

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

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

Traveling along the contour

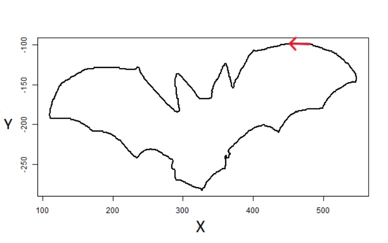
We need to travel along those pixels in an orderly way.

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

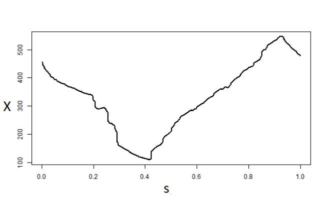
$$[(x[1], y[1]), (x[2], y[2]), \dots, (x[T], y[T])] \tag{1}$$

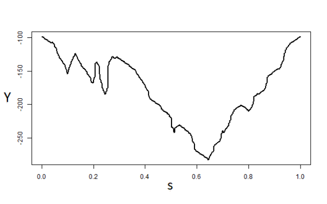
Starting from the top-right of the contour (this is important).

Traveling along the contour



Contour of the bat.





Functional data analysis

Functional data analysis: An introduction

Functional data analysis (FDA) (Ramsay & Silverman, 2005) is a field of statistics focused on studying data sets of functions.

Supervised learning problem examples:

- In regression problems, functions can be predictors:
- $y_i = \alpha + \int_T \beta(t) x_i(t)\, dt + \varepsilon_i$
- or responses:
- $y_i(t) = \mu(t) + \alpha_i(t) + \varepsilon_i(t)$

Functional data analysis: An introduction

Unsupervised learning problem examples:

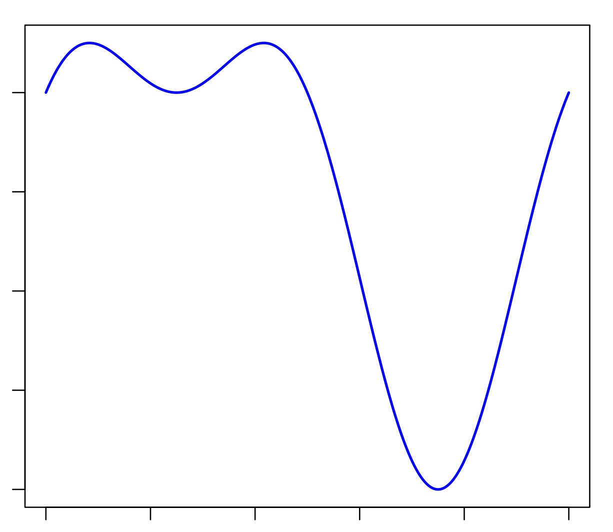
- Functional principal component analysis (FPCA) allows us to:
- project functions to a low-dimensional representation,
- and identify regions of high variability across data points.

In summary, many statistical analyses defined for continuous and categorical variables can be applied to functional variables.

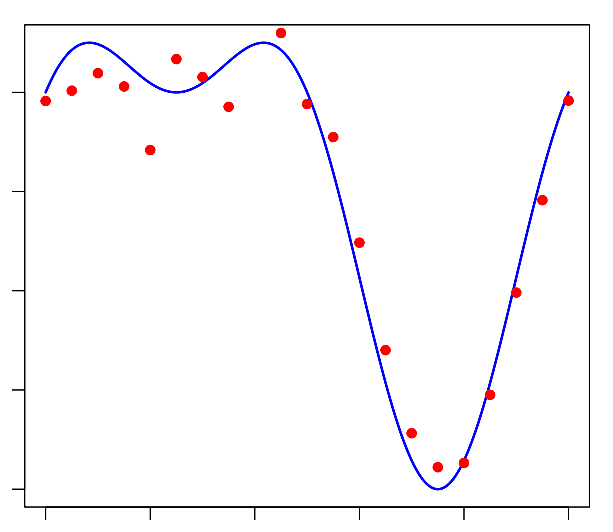
Smoothing observed functional data

- Data are naturally collected and stored in a discrete manner: $x[t] = x(t) + \varepsilon$.
- A common approach is to reconstruct the function before analysis.
- To estimate the smooth function $x(t),\ t \in (0, 1)$, we smooth the discrete data $x[t]$ using a basis expansion.
- Examples include B-spline expansions and Fourier expansions.

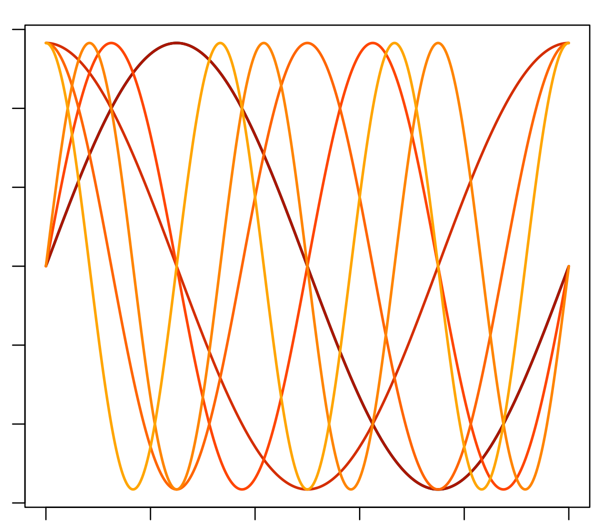
Example



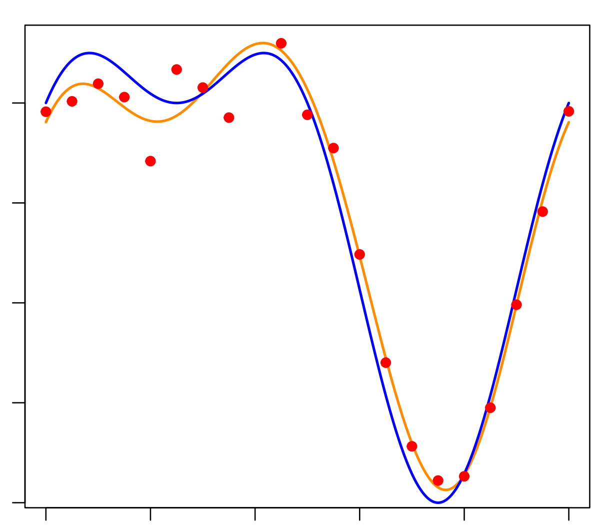
Example



Example



Example



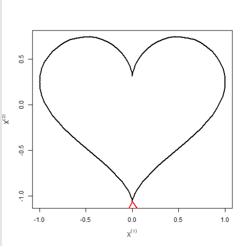
Functional representation of the contour

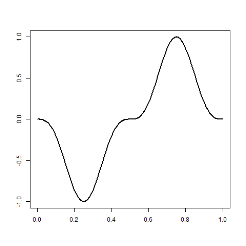
We use basis expansion to smooth the coordinate paths obtained with marching square.

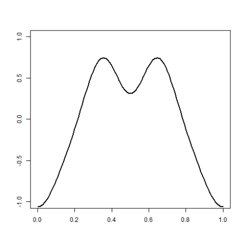
It gives us a smooth, continuous and parametric representation of the contour.

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

Functional representation of the contour







Shapes and statitics

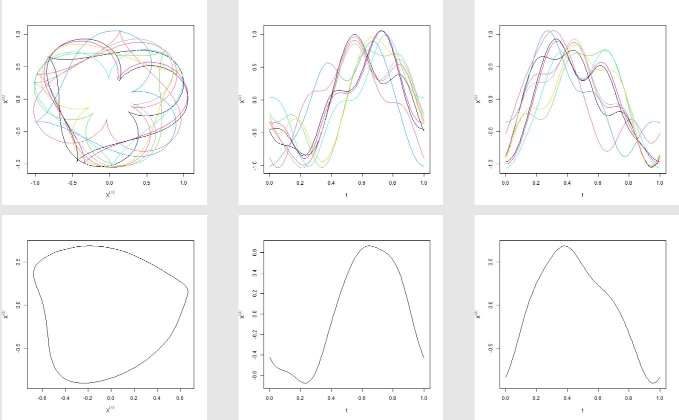
So what about repetition and data sets?

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 estimate and remove the effects of deformation variables.

These are:

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

Functional contour alignment

Existing work in shape analysis (Srivastava & Klassen, 2016) projects shapes onto a tangent space.

This removes the effect of the deformation variables.

However, this would prevent the statistical analysis of these variables.

What if the size of the object has statistical meaning?

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.

Functional contour alignement

The resulting contour we observed is parametrized as:

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

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

Estimating the scale and translation

Estimating $\mathbf{T}_i = (T_x, T_y)$ and ρ_i is rather simple.

If we want shapes to be centered at $(0,0)$ and to have unitary norm, this means that:

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

$$||\tilde{C}||_{\mathcal{H}} = \int_0^1 \tilde{X}^2(t)dt + \int_0^1 \tilde{Y}^2(t)dt = 1$$

Estimating the scale and translation

It makes estimation $\mathbf{T}_i$ and ρ_i easy:

$$\mathbf{T}_i = \int_0^1 \mathbf{C}_i(t)dt$$

$$\rho_i = ||\mathbf{C}_i - \mathbf{T}_i||_{\mathcal{H}}$$

This is extremely quick 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.

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.

Starting point (reparametrization)

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

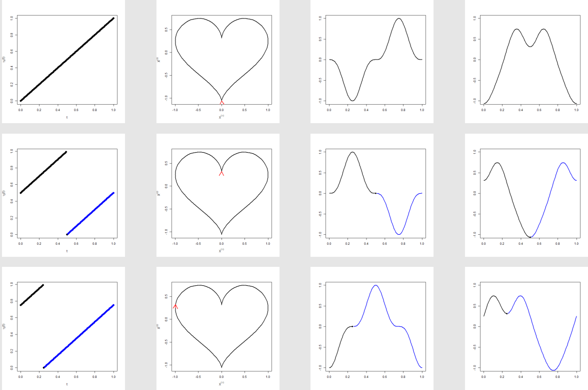
We define $\gamma \in \Gamma$, with

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

Starting point (reparametrization)

we can visualise the effect of this function here:

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



$\delta = 0$

$\delta = 0.5$

$\delta = 0.155$

Reparametrization

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

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

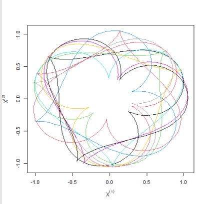
Reparametrization

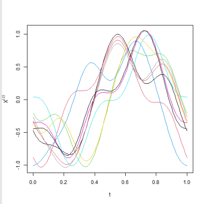
One might think that we can wrap the functions until they are aligned.

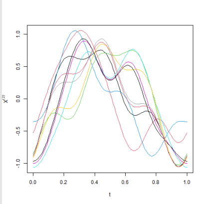
But the coordinate functions are entirely different for different rotations.

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

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 $\boldsymbol{\mu}$.

$$(\hat{\theta}, \hat{\delta}) = \underset{(\theta,\delta)\in[0,2\pi]\times[0,1]}{\arg\min} ||\mathbf{O}_\theta \mathbf{C}_i^* \circ \gamma_\delta - \boldsymbol{\mu}||_{\mathcal{H}}^2. \quad (3)$$

Estimating the reparametrization and the rotation

Solving equation 3 is difficult.

However, representing the pre-shape $\mathbf{C}^*$ and the template $\boldsymbol{\mu}$ using the Fourrier basis expension has multiple benefits.

- Leads to a nice solution for the estimation of $\mathbf{T}$ and ρ
- Leads to a solution for the rotation/reparametrization of the form $\hat{\theta} = f(\hat{\delta})$ (and inverse)

The use of Fourrier basis expension was fundamental in solving equation 3.

Estimating the reparametrization and the rotation

Having a way to express $\hat{\delta}$ as a function of $\hat{\theta}$ means that we can solve the alignement issues by

- Searching on a grid (for δ) for an optimum value
- Developping an iterative algorithm (ICP-like) 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 $\tilde{\mathbf{C}}$.

$$\mathbf{C}^*(t) = \frac{1}{\rho}(\mathbf{C}(t) - \mathbf{T})$$
$$\tilde{\mathbf{C}}(t) = \mathbf{O}_\theta \mathbf{C}^*(t) \circ \gamma_{(\delta)}$$

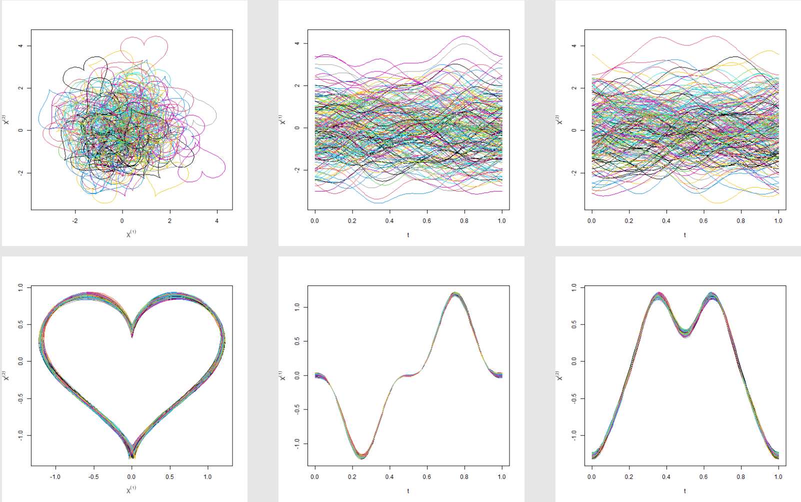
Alignement results

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

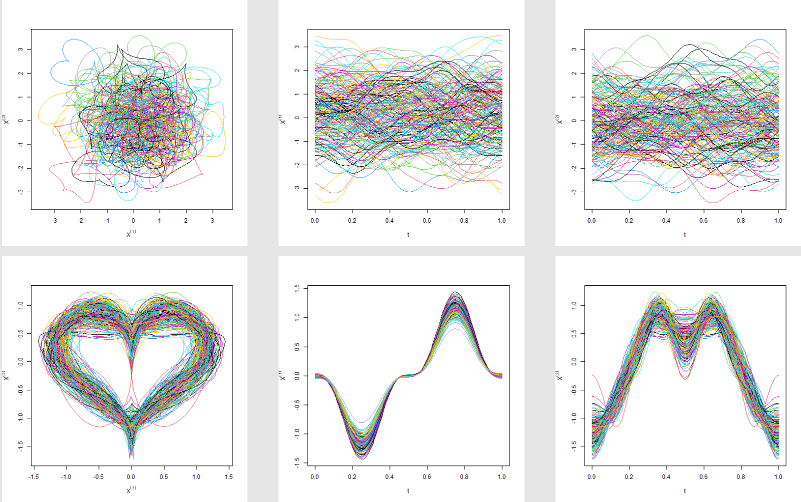
We deformed shapes and realigned them (simulations).

We also aligned real data.

Alignement on simulated data



Alignement on simulated data



Alignement on 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}

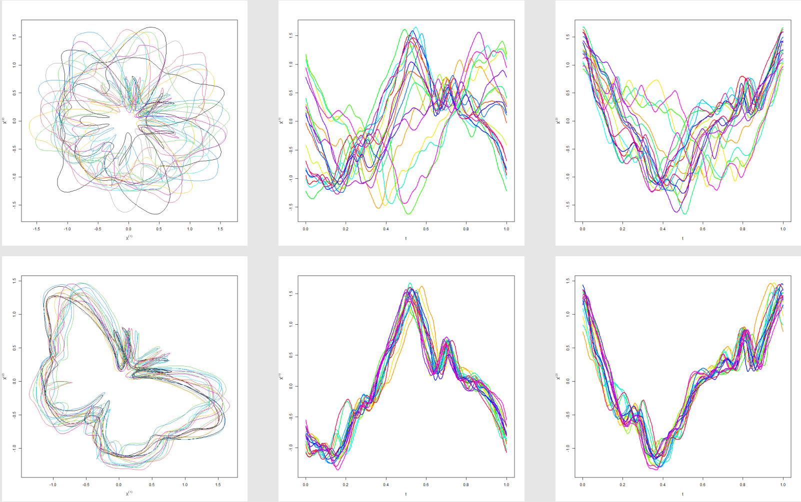
Table: MSE of the estimated parameters for the different scenarios and values of σ

Alignement on real data

MPEG-7 database:

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

Alignement on real data



Functional shape analysis

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 $\mathbf{X}$ 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).

Breaking down the procedure

- Edge/contour detection for a collection of images.
- Extract an ordered list of pixels.
- Learn a functional representation for both coordinates.
- Estimate deformation parameters.
- Remove the deformations to obtain the shape as two univariate functions.
- Statistical analysis of the deformation variables and shapes.

Functional Principal component

PC1 (57%) PC2 (22%) PC3 (9%)

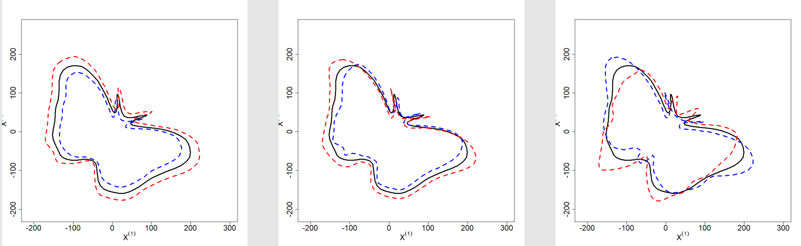


Figure: Plots of the estimated mean function $\bar{\mathbf{z}} = \sum_i \mathbf{z}_{i1}$ in black, of $\bar{\mathbf{z}} - 20\hat{\phi}_k$ in blue and of $\bar{\mathbf{z}} + 20\hat{\phi}_k$ in red, for $k = 1$ (first column), $k = 2$ (second column) and $k = 3$ (third column).

Shape generation

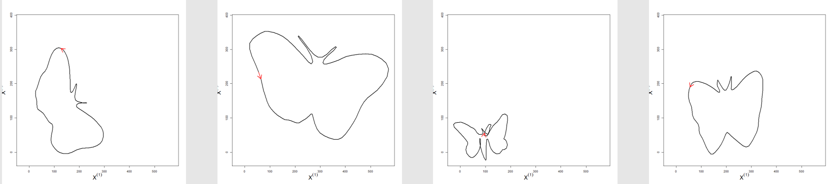


Figure: Butterfly curves generated with our approach with the deformation parameters.

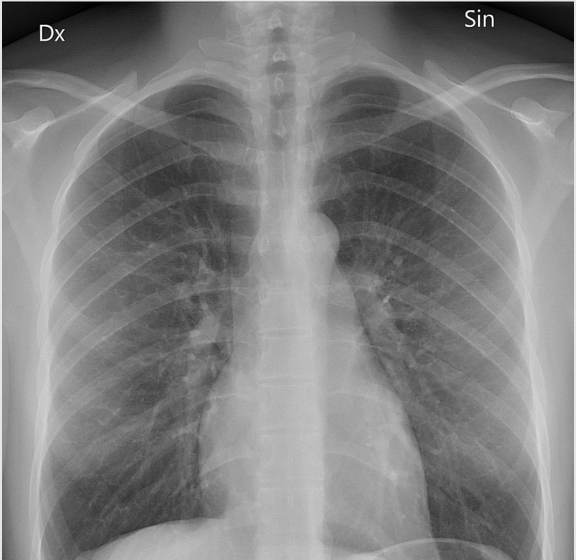
Multiple shapes

A second project extends our approach to consider multiple shapes at once.

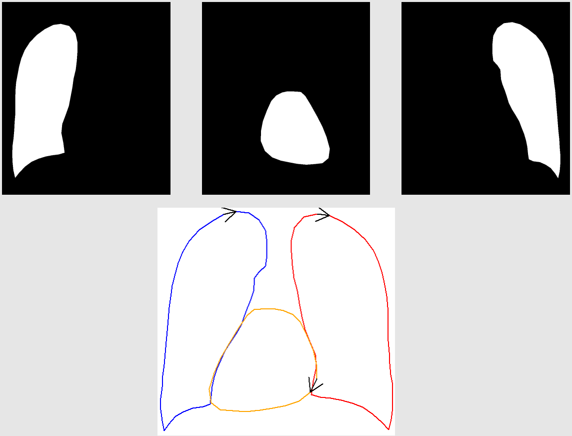
Applications on X-rays used to identify patients with cardiomegaly.

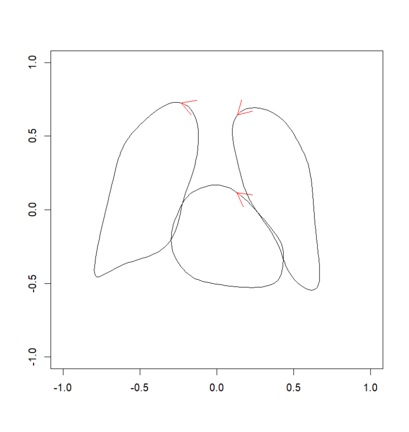
This forced us to question many aspects of the alignment procedure.

Multiple shapes

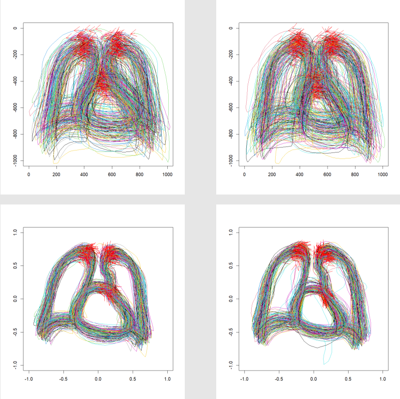


Multiple shapes





$Y = 1$ $Y = 0$



Multiple shapes: classification results

Classification accuracy with linear functional model:

LA	GL	GFUL	PCR	PLS
82.7	82.8	82.8	83.1	85.9

Conclusion

We are developing new approaches to analyze images by considering the shapes within them.

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.

Conclusion

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

Shapes and deformation parameters can be input or output of neural networks for non-linear learning.

Can be added in current pipeline has an additionnal representation of images.

Current development

New edge detection procedure that directly provides us with a Fourrier representation of the contour:

- Insipred by snakes.
- Trying to redefine the energy function.

I would love to answer your questions.

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.

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 shapes and applications for (segmented) images classification, writting

Alignement on real data: deformations

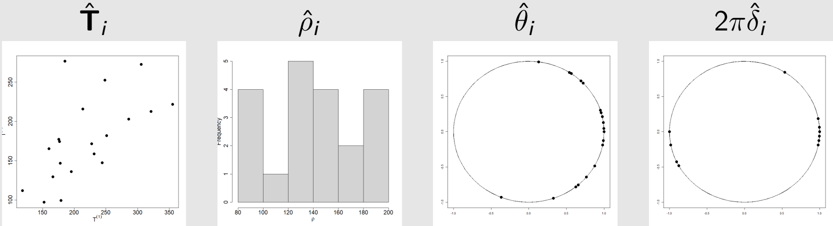


Figure: Plots of the estimated deformation parameters of each curve in both datasets