Rethinking Image Data through Functional Representations of Shapes and Surfaces

Cédric Beaulac

Université du Québec à Montréal

November 25, 2025

Joint work in collaboration with:

Issam-Ali Moindjié
Perpignan

Marie-Hélène Descary
UQAM

Mélanie Raymond
UQAM

Rethinking Image Data through Functional Representations of Shapes and Surfaces

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.

- Problem with interpretation.
- Large data (high-resolution videos).
- Generalization issues (sensitivity to resolution and technology).

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.

Mask 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

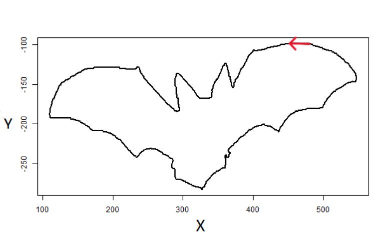
We need to travel along those pixels in an orderly way.

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

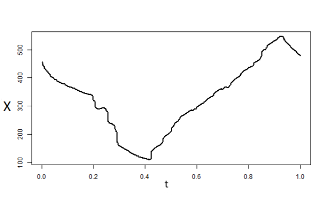
$$[(x[1], y[1]), (x[2], y[2]), \ldots, (x[T], y[T])] \tag{1}$$

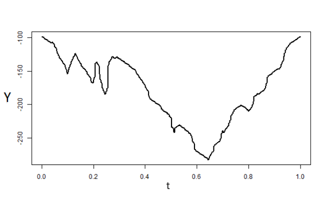
Starting from the top-right of the contour (this is important).

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:

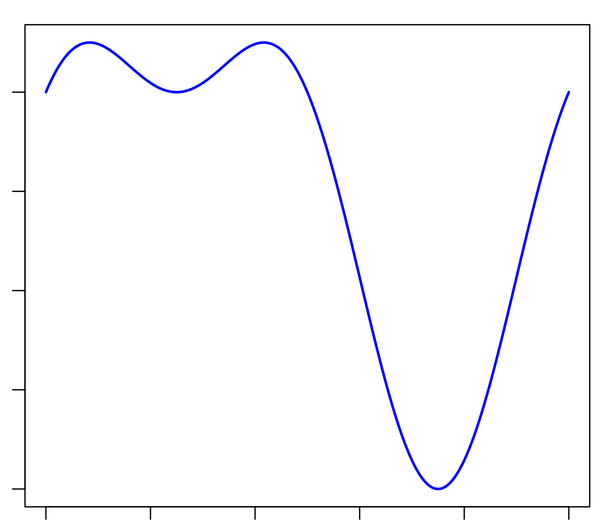
- Functional principal component analysis (FPCA) allows us to:
- project functions to a low-dimensional representation,
- and identify regions of high variability across data points.

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

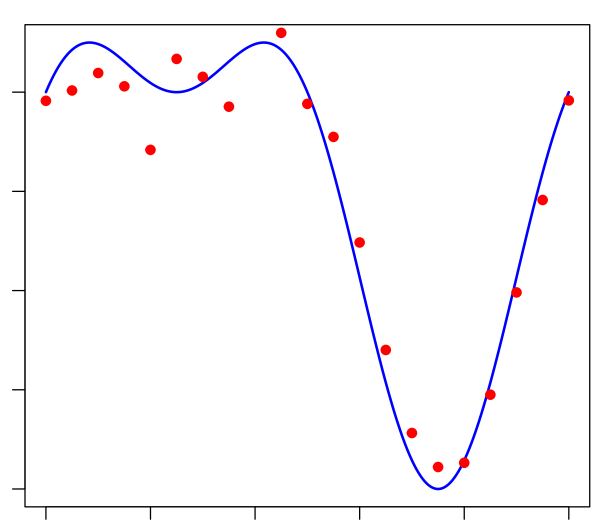
Smoothing observed functional data

- 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.

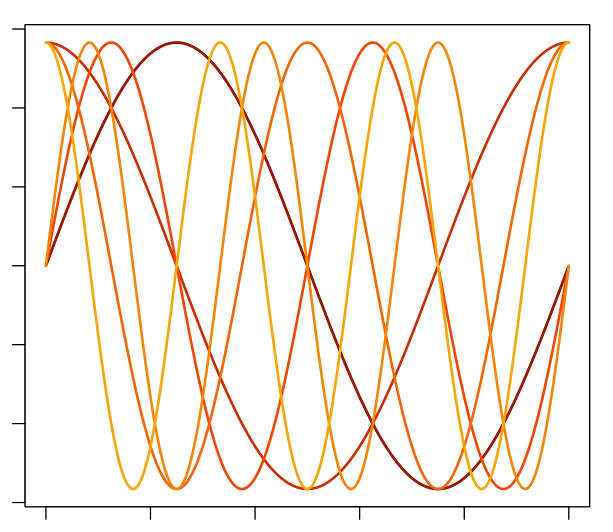
Example



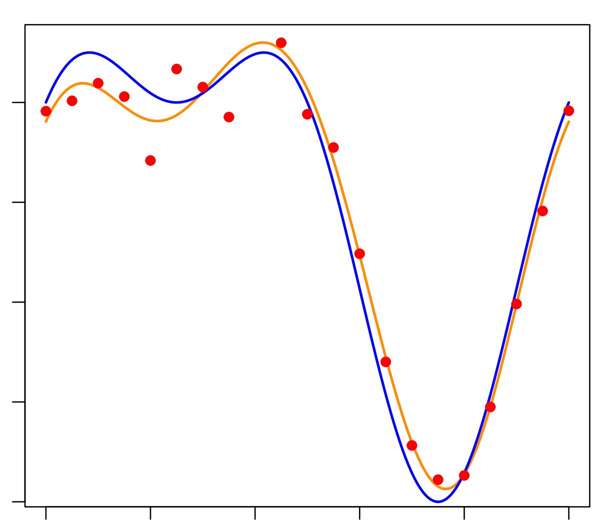
Example



Example



Example



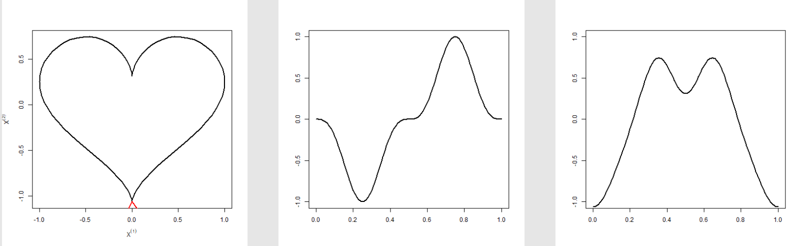
Functional representation of the contour

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

Functional representation of the contour



Shapes and statistics

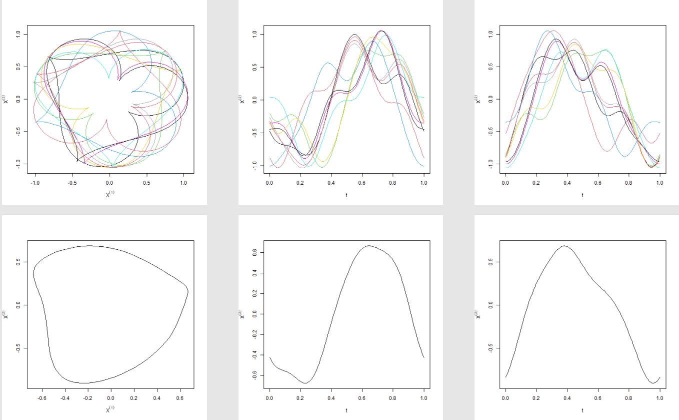
So what about repetition and data sets?

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 statistics

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



Functional contour alignment

Functional contour alignment

A shape is invariant with respect to translation, scaling and rotation.

We extrated the contour not the shape.

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 contours 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 alignment

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 alignment

The resulting contour we observed is parameterized 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.

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 unit 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 independently.

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 reparameterization 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 (reparameterization)

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

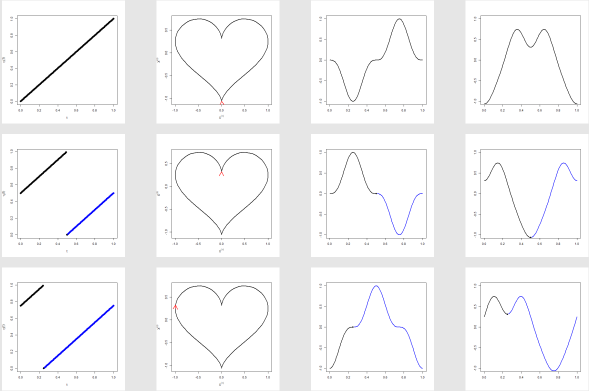
We define $\gamma \in \Gamma$, with

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

Starting point (reparameterization)

we can visualize the effect of this function here:

	γ_δ	$\tilde{C} \circ \gamma_\delta$	$\tilde{x} \circ \gamma_\delta$	$\tilde{y} \circ \gamma_\delta$
$\delta = 0$				
$\delta = 0.5$				
$\delta = 0.65$				



Reparameterization

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

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

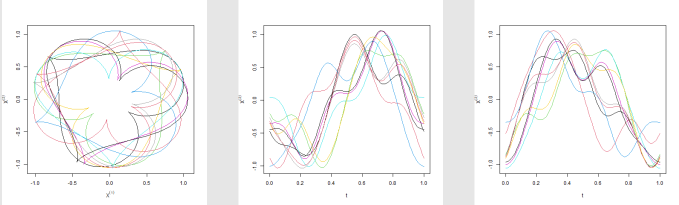
Reparameterization

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 parameterized 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 reparameterization 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 reparameterization 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 $\boldsymbol{\mu}$.

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

Estimating the reparameterization and the rotation

Solving equation 3 is difficult.

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

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

The use of Fourier basis expansion was fundamental in solving equation 3.

Estimating the reparameterization and the rotation

Having a way to express $\hat{\delta}$ as a function of $\hat{\theta}$ means that we can solve the alignment 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 reparameterization and the rotation

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

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

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

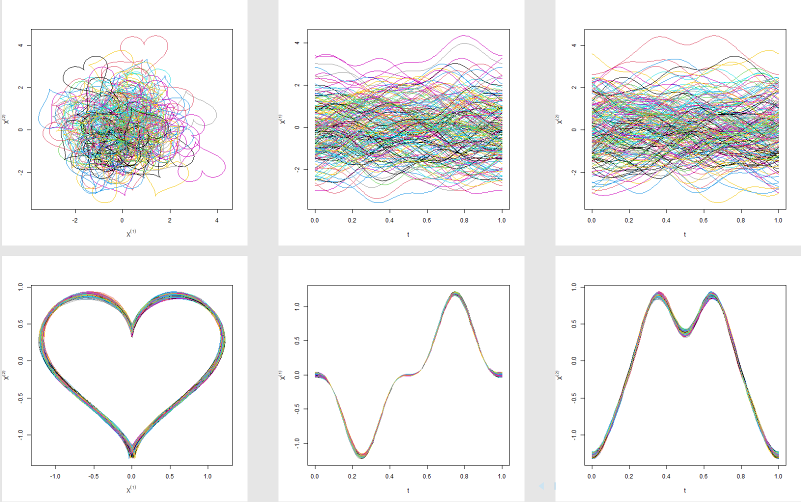
Alignment results

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

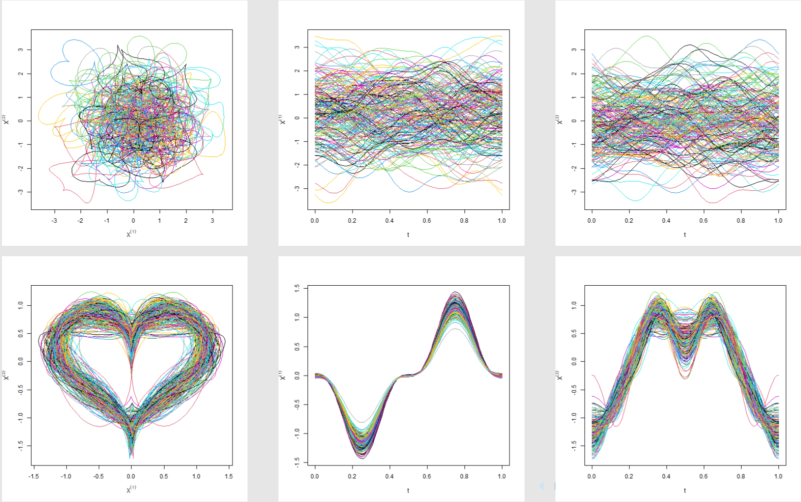
We deformed shapes and realigned them (simulations).

We also aligned real data.

Alignment on simulated data



Alignment on simulated data



Alignment 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}

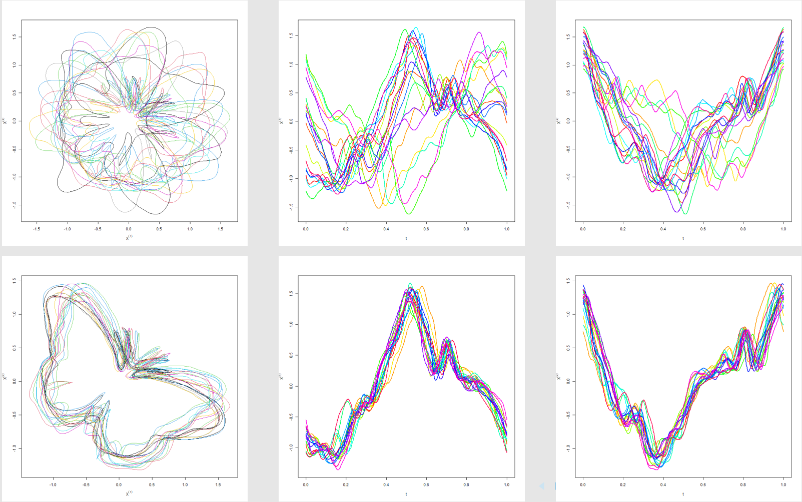
Table: MSE of the estimated parameters for the different scenarios and values of σ

Alignment on real data

MPEG-7 database:

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

Alignment 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

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

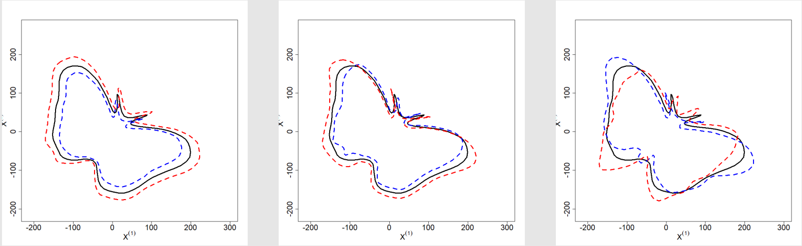


Figure: Plots of the estimated mean function $\bar{\mathbf{z}} = \sum_i \mathbf{z}_{i1}$ in black, of $\bar{\mathbf{z}} - 20\hat{\phi}_k$ in blue and of $\bar{\mathbf{z}} + 20\hat{\phi}_k$ in red, for $k = 1$ (first column), $k = 2$ (second column) and $k = 3$ (third column).

Generation

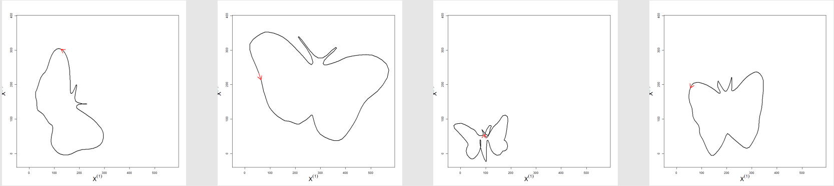


Figure: Butterfly curves generated with our approach with the deformation parameters.

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 benign.
- Our goal was to compare the performance of a shape-based model against a pixel-based one.

Classification: Melanoma detection based on mole shapes

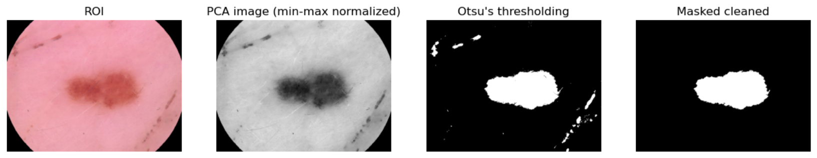


Figure: Segmentation pipeline.

Classification: Melanoma detection based on mole shapes

Model	avg-AUC	avg-Bal.Accuracy	avg-F1
Shape	0.640 ± 0.031	$\mathbf{0.572 \pm 0.036}$	$\mathbf{0.837 \pm 0.020}$
Pixel	$\mathbf{0.697 \pm 0.069}$	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)
Shape	**0.7**	$\mathbf{31.7 \pm 6.92}$
Pixel	66.7	174 ± 82.9

Table: Stratified-Nested-CV Results overview.

Extensions

Multiple shapes

A second project extends our approach to consider multiple shapes in a single image.

This is to better represent realistic objects.

This forced us to question the alignment procedure and how to consider deformation variables.

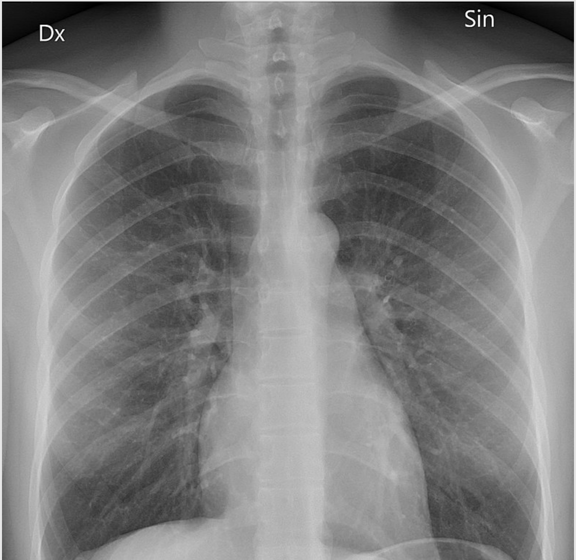
Multiple shapes

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

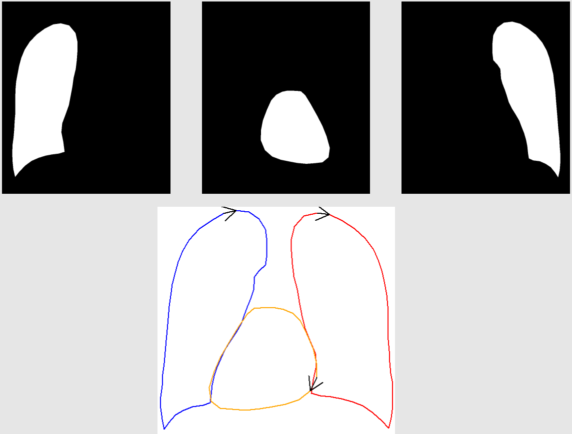
Overall the idea is to consider deformation variable to be global and affect all shapes the same way.

This then capture relative differences in the shapes $\tilde{C}$

Multiple shapes



Multiple shapes

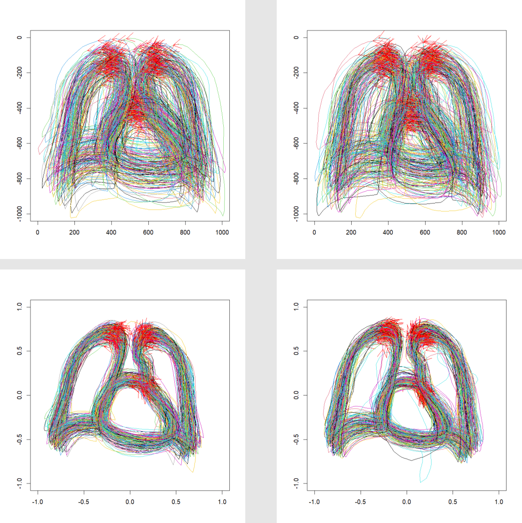


Multiple shapes: alignment results

$Y = 1$ $\qquad$ $Y = 0$

Original data

Aligned data



Neural network integration

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

Can also be input in recently developed functional layers.

Can be added in current pipeline has an additional representation of images in tandem of pixel representation.

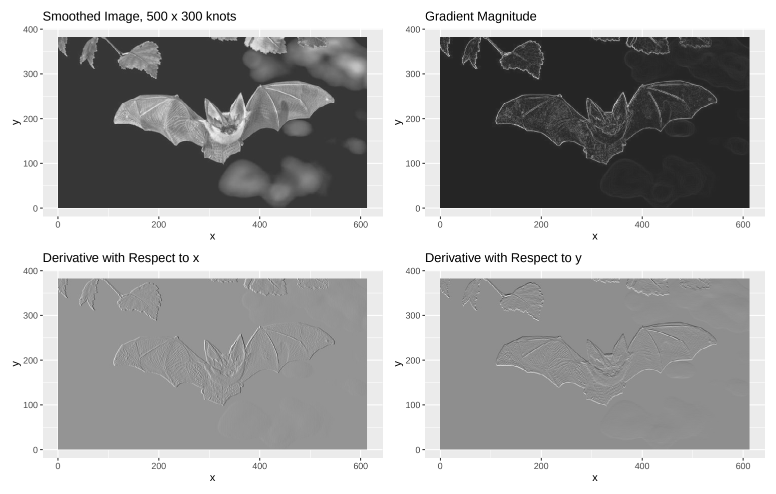
Current development

Current development: Images as surface

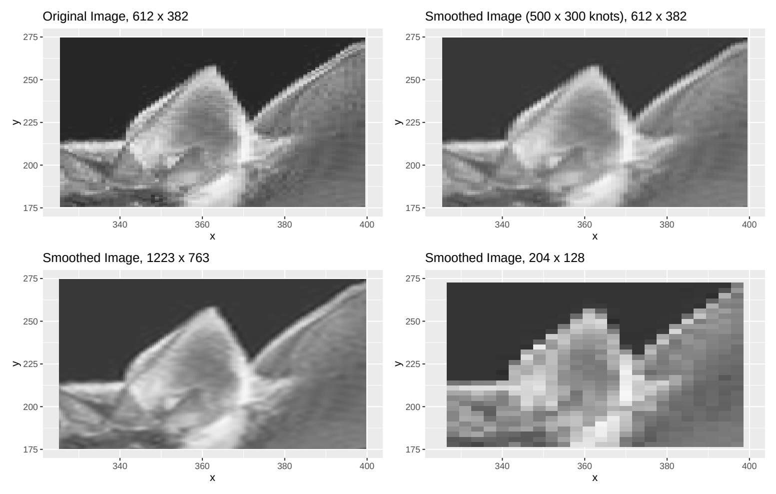
We model images as surfaces and pixels as a grid of samples over that surface.

- We can recover a smooth surface with tensor-product P-splines.
- Dramatically reduce the dimension of the image.
- Can quickly change the resolution using non-linear interpolation.
- Quick computation of derivatives needed for edge-detection.

Current development: Images as surface



Current development: Images as surface



Conclusion

Statistical shape analysis can provide new information about images useful 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.

Conclusion

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

Treating images as surfaces can improve the shape extracted from images but can also bring a new perspective on image analysis.

Provide a parsimonious representation with multiple benefits and almost no cons.

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.

Alignment on real data: deformations

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

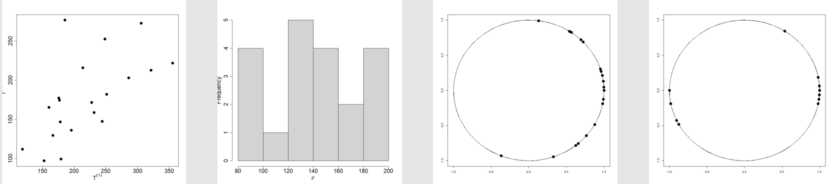


Figure: Plots of the estimated deformation parameters of each curve in both datasets

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.

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