

Vers une représentation fonctionnelle des images: formes, alignement et apprentissage

Cédric Beaulac

Université du Québec à Montréal

19 février 2026

Des visages bien connus ici:



**Issam-Ali
Moindjié**
Perpignan



**Marie-Hélène
Descary**
UQAM



**Mélanie
Raymond**
UQAM



**Louis-Pierre
Ménard**
UQAM

Vers une représentation fonctionnelle des images

Formes, alignement et apprentissage

Images

Données fonctionnelles

Alignement des contours

Analyse de formes

Extensions

Travaux en cours

Images

Une image, c'est quoi ?

On fait référence à une surface colorée bidimensionnelle (2D).

Les termes image et photo seront utilisés de manière interchangeable.

Image en couleur



Figure: Image en couleur d'une pizza.

Image en *teinte de gris*

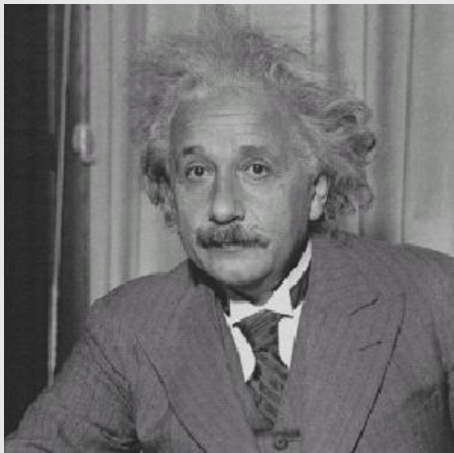


Figure: Image en noir et blanc (teinte de gris) d'Albert Einstein.

Image: objet mathématique

Depuis la numérisation de la photo, une image de résolution (h, l) est représentée par :

- ▶ une matrice de taille $h \times l$,
- ▶ notée P ,
- ▶ où le terme $P_{i,j}$ représente la coloration du pixel (i,j) .

Image: objet mathématique

Pour les images en teintes de gris, la coloration est identifiée par une valeur dans l'intervalle $[0, 255]$.

Avec blanc = 0, noir = 255 et un spectre de couleurs intermédiaires.

Il y a donc $256 = 2^8$ valeurs possibles pour la coloration d'un pixel.

Image: exemple simple



*Image en noir et blanc du chiffre
'1'.*

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

*Représentation matricielle (pixels
d'intensité de couleur)*

Image: exemple simple

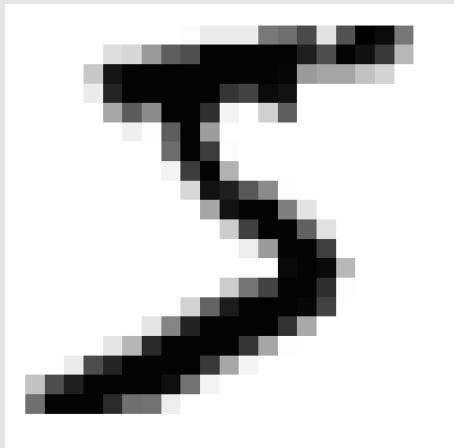


Figure: Image d'un '5' provenant de la base de données MNIST.

Image: exemple simple

	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]	[1,10]	[1,11]	[1,12]	[1,13]	[1,14]	[1,15]	[1,16]	[1,17]	[1,18]	[1,19]	[1,20]	[1,21]	[1,22]	[1,23]	[1,24]	[1,25]	[1,26]	[1,27]	[1,28]
[1,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
[2,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
[3,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
[4,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
[5,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
[6,1]	0	0	0	0	0	0	0	0	0	0	0	0	3	18	18	18	126	136	175	26	166	255	247	127	0	0	0	0
[7,1]	0	0	0	0	0	0	0	0	30	36	94	154	170	253	253	253	253	253	253	225	172	253	242	195	64	0	0	0
[8,1]	0	0	0	0	0	0	0	49	238	253	253	253	253	253	253	253	253	253	251	91	82	82	56	39	0	0	0	0
[9,1]	0	0	0	0	0	0	0	18	219	253	253	253	253	253	198	182	247	241	0	0	0	0	0	0	0	0	0	0
[10,1]	0	0	0	0	0	0	0	80	156	107	253	253	205	11	0	43	154	0	0	0	0	0	0	0	0	0	0	0
[11,1]	0	0	0	0	0	0	0	0	34	1	154	253	90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[12,1]	0	0	0	0	0	0	0	0	0	0	139	253	190	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[13,1]	0	0	0	0	0	0	0	0	0	11	190	253	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[14,1]	0	0	0	0	0	0	0	0	0	0	35	241	225	180	108	1	0	0	0	0	0	0	0	0	0	0	0	0
[15,1]	0	0	0	0	0	0	0	0	0	0	0	81	140	253	253	119	25	0	0	0	0	0	0	0	0	0	0	0
[16,1]	0	0	0	0	0	0	0	0	0	0	0	45	186	253	253	150	27	0	0	0	0	0	0	0	0	0	0	0
[17,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	91	252	253	187	0	0	0	0	0	0	0	0
[18,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	249	253	249	64	0	0	0	0	0	0	0	0
[19,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	130	183	253	253	207	2	0	0	0	0	0	0	0
[20,1]	0	0	0	0	0	0	0	0	0	0	0	39	148	229	253	253	253	253	254	182	0	0	0	0	0	0	0	0
[21,1]	0	0	0	0	0	0	0	0	0	24	114	221	253	253	253	253	201	70	0	0	0	0	0	0	0	0	0	0
[22,1]	0	0	0	0	0	0	0	0	23	66	213	253	253	253	253	198	81	2	0	0	0	0	0	0	0	0	0	0
[23,1]	0	0	0	0	0	0	0	18	171	219	253	253	253	253	195	80	9	0	0	0	0	0	0	0	0	0	0	0
[24,1]	0	0	0	0	55	172	226	253	253	253	253	244	133	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[25,1]	0	0	0	0	136	253	253	253	253	212	135	132	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[26,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[27,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[28,1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure: Représentation matricielle de l'image du '5'.

Image: surface

On peut aussi représenter les images comme une surface où l'intensité de coloration varie en fonction de la position (i, j) .

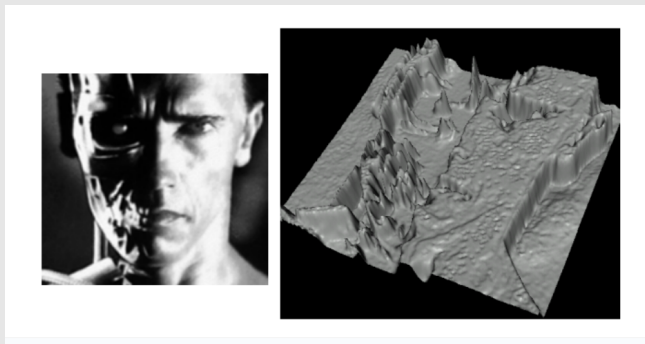


Figure: Une image à gauche et la surface qui en découle à droite.

Image en couleurs

Il s'agit d'une combinaison de rouge (R), de vert (G) et de bleu (B).

L'intensité de chaque couleur est identifiée à l'aide d'un entier dans l'intervalle $[0, 255]$.

La couleur d'un pixel est représentée par une combinaison de R, G et B.

Donc, une image couleur est représentée par une combinaison des matrices R , G et B .

Représentation des couleurs

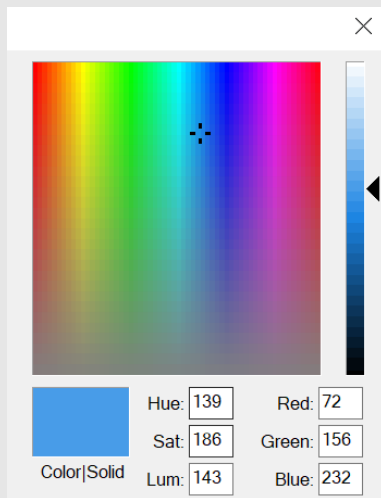


Figure: Représentation d'une couleur comme une combinaison *red*, *green* et *blue*

Image en couleurs

Conséquemment, l'objet mathématique représentant une image couleur de l pixels de large et h pixels de haut est un tenseur tridimensionnel de taille $(3, h, l)$.

Nous avons essentiellement trois matrices de taille (h, l) , une pour chacune des trois couleurs.

Image en couleurs



Figure: Une combinaison de trois teintes forme une image en couleur.

État des choses en analyse d'image

Les approches standards d'analyse d'images analysent ces matrices directement.

Les outils pour **modifier**, **transformer** et **analyser** les images considèrent celles-ci exclusivement comme une **collection de pixels**, enregistrée sous forme matricielle.

Analyse d'image

Par exemple, la filtration et la convolution sont des opérateurs matriciels correspondant à des combinaisons linéaires de pixels voisins.

Des modèles de classification performants sont mis au point en apprenant les poids de ces combinaisons dans des réseaux de neurones complexes.

Analyse d'image : des problèmes

Malgré les récentes avancées permises par les réseaux de neurones convolutifs, les approches fondées sur l'analyse de matrices de pixels souffrent de nombreux problèmes :

- ▶ L'interprétation est extrêmement difficile.
- ▶ Les données sont trop volumineuses.
- ▶ La généralisation est difficile (sensibilité aux changements technologiques).

Analyse d'image : notre solution

Dans ma recherche, je propose d'arrêter de penser aux images comme une simple collection de pixels.

À la place, nous voulons paramétriser les images comme une collection d'objets définis par leurs formes, leurs textures et leurs couleurs.

Dans cette présentation, nous discuterons, entre autres, de l'analyse de formes.

Détection et extraction de contour

Détection de contour

Un contour, c'est quoi ?

Intuitivement, c'est le tracé qui démarque un objet d'un autre.

Détection de contour



Photo de chauve-souris.



Image *masque*.

Détection de contour

Comment paramétrer ce contour ?

On peut utiliser des fonctions coordonnées (x, y) pour définir un parcours le long du contour.

Animation ici.

Paramétrisation du contour

En supposant qu'on peut obtenir des images masques.

On extrait une fonction contour (courbe planaire fermée), représentée par :

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

où $t \in [0, 1]$ représente la proportion de la courbe parcourue du départ ($t = 0$) jusqu'à la fin ($t = 1$).

(Courbe fermée : $C(0) = C(1)$).

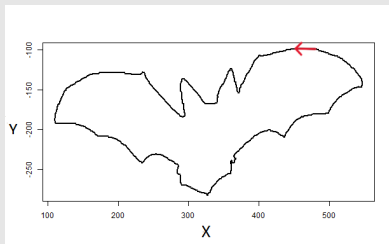
Paramétrisation du contour

Heureusement pour nous, il existe un algorithme qui parcourt le contour de manière ordonnée.

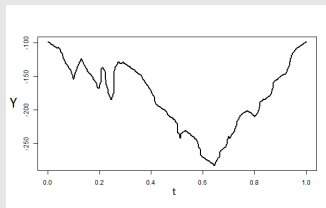
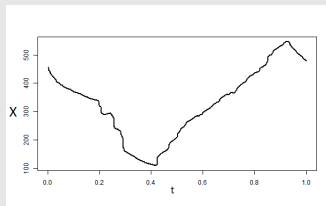
L'algorithme *marching squares* (Mantz et al., 2008) nous retourne une liste de la forme :

$$[(x[1], y[1]), (x[2], y[2]), \dots, (x[T], y[T])] \quad (1)$$

Parcours le long du contour



Le contour de la chauve-souris.



Un contour : une donnée fonctionnelle multivariée

Notre objectif est de modéliser le contour des objets dans les images comme des données fonctionnelles multivariées.

L'avantage principal est de pouvoir utiliser la grande boîte à outils disponible pour de telles données.

Données fonctionnelles

Données fonctionnelles

L'analyse de données fonctionnelles est un domaine de la statistique intéressé par l'analyse de fonctions $x(t)$, de courbes, ou bien de fonctions de plus haute dimension telles que $x(s, t)$, des surfaces.

Données fonctionnelles

En analyse de données fonctionnelles, une observation $x_i(t)$, $t \in T$, est une fonction.

L'espace sur lequel la fonction est définie, T , peut être le temps, par exemple, ou n'importe quel espace multidimensionnel.

Un ensemble de données est donc une collection de ce type d'observations : $S = \{x_i(t) \mid i \in (1, \dots, n)\}$ définies sur le même espace T .

Données fonctionnelles

Il est impossible de collecter et d'entreposer $x(t)$ pour tout $t \in T$.

Les données sont observées et enregistrées de manière discrète.

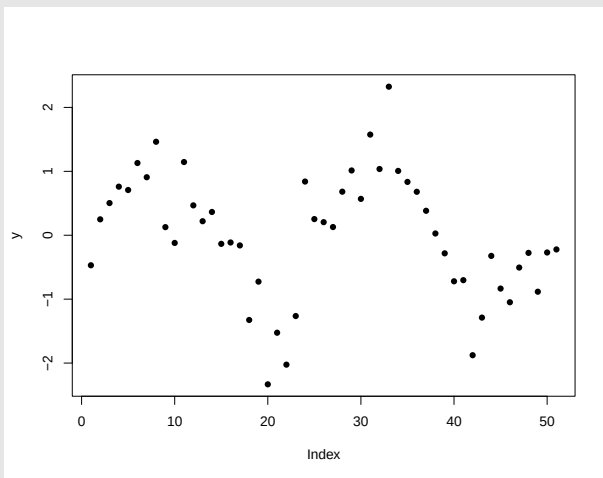


Figure: L'observation d'une variable fonctionnelle.

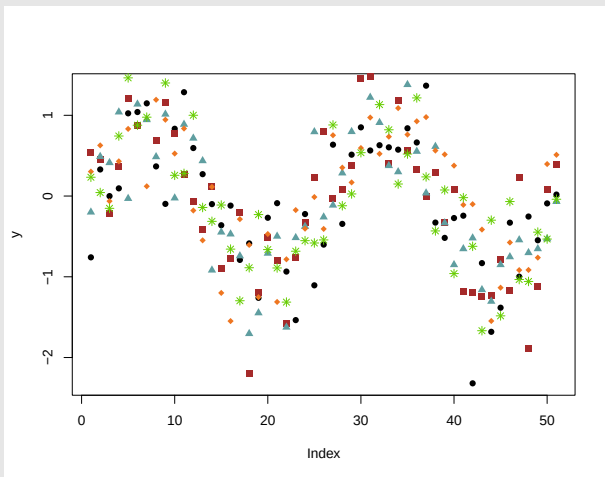


Figure: Un ensemble de données fonctionnelles.

Lissage et représentation fonctionnelle

- ▶ Nous supposons que les données fonctionnelles observées sont un échantillon provenant d'un processus lisse sous-jacent.
- ▶ Afin de modéliser le réel processus sous-jacent et de réduire l'effet de l'erreur de mesure, il est commun de lisser les données fonctionnelles observées.
- ▶ La représentation lisse est par la suite utilisée lors de l'analyse.

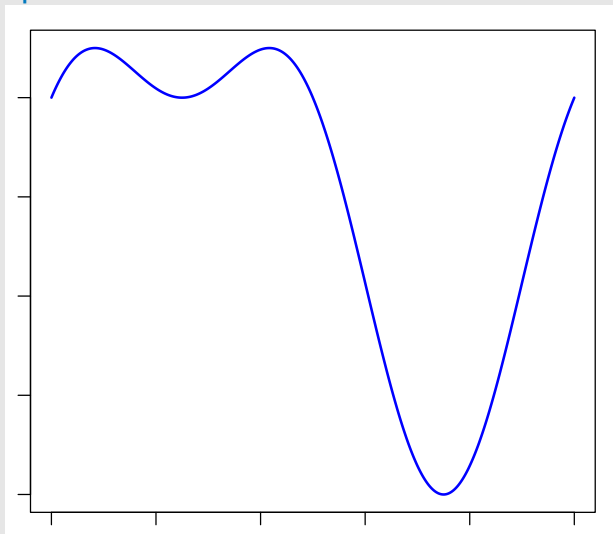
Représentation fonctionnelle

- ▶ L'approche standard est d'utiliser une base fonctionnelle : une collection de fonctions de base qui couvre le domaine T et des coefficients attachés à ces fonctions.
- ▶ La représentation fonctionnelle, lisse et continue est donc paramétrique.
- ▶ $x(t) = \sum_{k=1}^K c_k B_k(t)$
- ▶ Exemples : séries de Fourier, B-splines

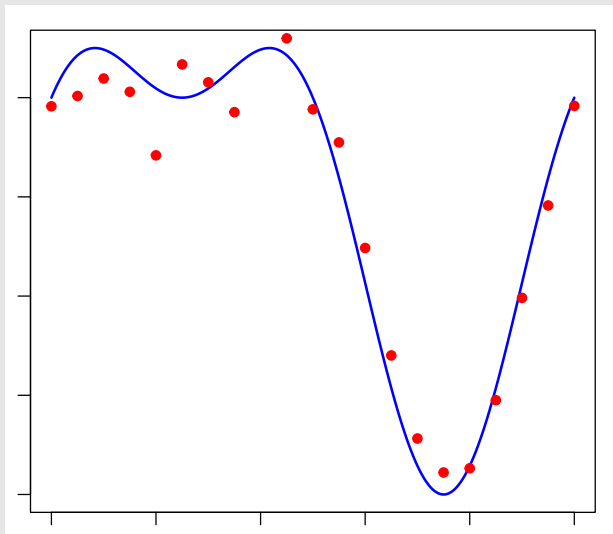
Représentation fonctionnelle

- ▶ $x(t) = \sum_{k=1}^K c_k B_k(t)$
- ▶ C'est bien puisqu'on peut évaluer $x(t)$ partout ($t \in [0, T]$).
- ▶ On a seulement besoin *d'apprendre* un petit nombre (discret) de paramètres pour produire cette représentation lisse et continue pour une observation fonctionnelle.
- ▶ On peut souvent le faire facilement en minimisant la fonction d'erreur quadratique suivante : $\sum_j [x(t_j) - \sum_{k=1}^K c_k B_k(t_j)]^2$

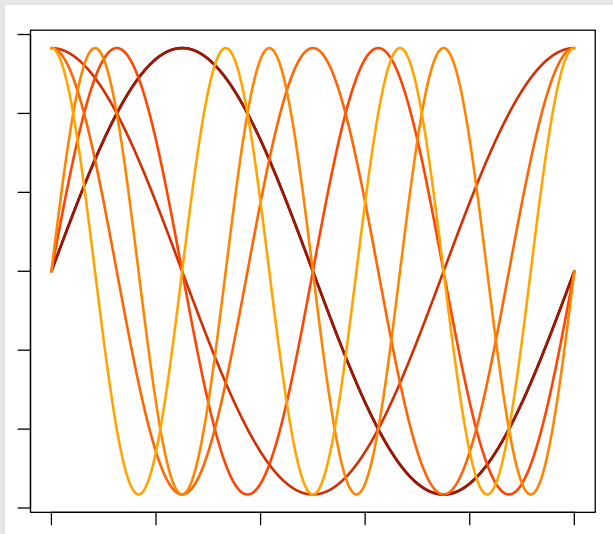
Exemple : processus lisse non observé



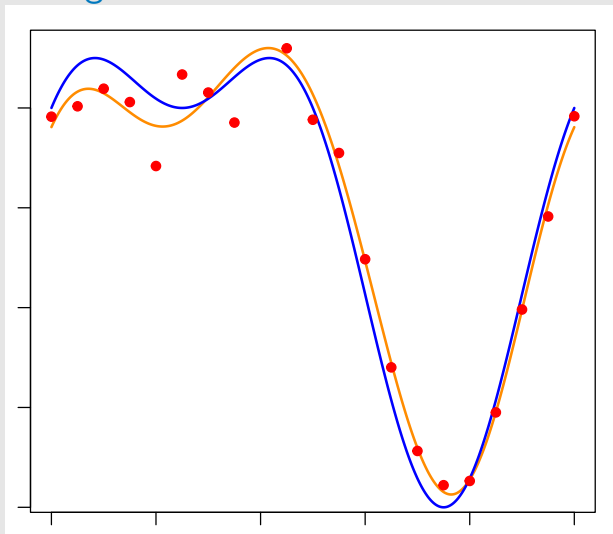
Exemple : un échantillon



Exemple : séries de Fourier



Exemple : lissage



Représentation fonctionnelle

Le lissage offre plusieurs avantages :

- ▶ Amointrit la variabilité indésirée.
- ▶ Une réduction de la dimension.
- ▶ Facilite le calcul des dérivées.
- ▶ Permet d'analyser des données collectées irrégulièrement.

Analyse de données fonctionnelles

Les fonctions lisses sont ensuite utilisées comme variables aléatoires dans des problèmes de statistique classique.

▶ En régression où les fonctions servent de prédicteurs :

▶ $y_i = \alpha + \int_T \beta(t)x_i(t) dt + \varepsilon_i$

▶ ou comme réponse :

▶ $y_i(t) = \mu(t) + \alpha_i(t) + \varepsilon_i(t)$

Analyse de données fonctionnelles

En apprentissage non supervisé :

- ▶ Analyse en composantes principales fonctionnelles (FPCA) :
- ▶ projette les fonctions vers un espace de plus petites dimensions,
- ▶ et identifie les régions (t) de grandes variabilités entre les fonctions.

Plusieurs outils classiques de la statistique furent adaptés pour accommoder les variables fonctionnelles.

Analyse en composantes principales fonctionnelles

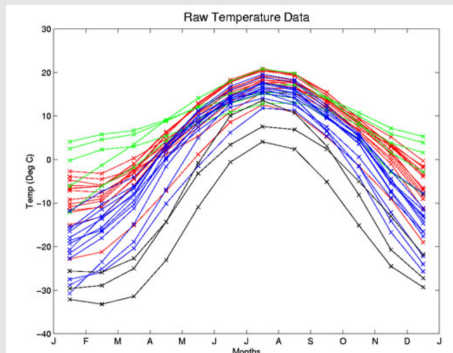


Figure: Échantillon de température sur une année collecté à plusieurs stations météo au pays.

Analyse en composantes principales fonctionnelles

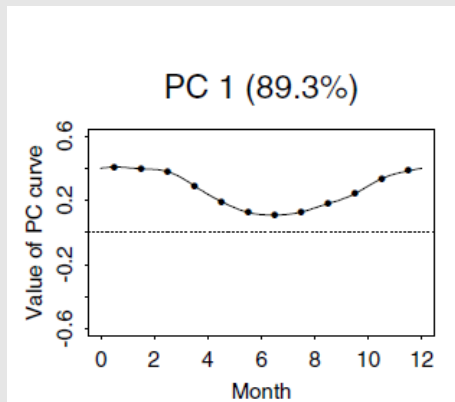


Figure: Première composante principale fonctionnelle associée.

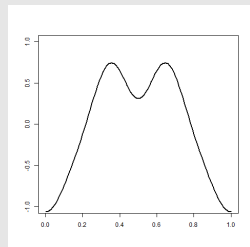
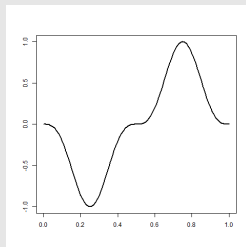
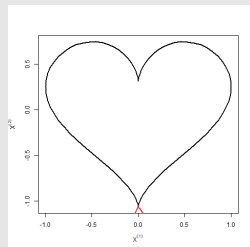
Un contour : une donnée fonctionnelle multivariée

On utilise des fonctions de base pour lisser les chemins coordonnés obtenus par *marching squares*.

Cela nous donne une représentation lisse, continue et paramétrique du contour.

On peut maintenant utiliser les approches d'analyse de données fonctionnelles multivariées pour répondre à une multitude de problèmes statistiques en lien avec l'analyse de formes.

Un contour : une donnée fonctionnelle multivariée



Exprimer les contours par une série de Fourier

Étant donné la périodicité des courbes planaires fermées, nous avons décidé d'utiliser une expansion en série de Fourier pour lisser le contour $C(t)$.

$$C(t) = \sum_{k=0}^M c_k \beta_k(t), \quad t \in [0, 1]$$

avec

$$\beta_0(t) = 1, \quad \beta_k(t) = \begin{cases} \sqrt{2} \sin((k+1)\pi t) & \text{pour } k \text{ impair,} \\ \sqrt{2} \cos(k\pi t) & \text{pour } k \text{ pair.} \end{cases}$$

Données et statistiques

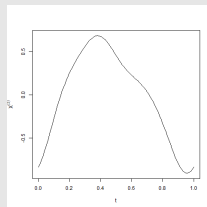
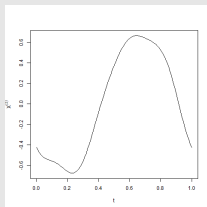
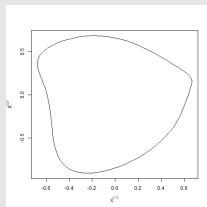
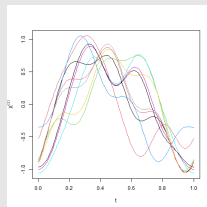
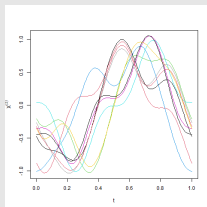
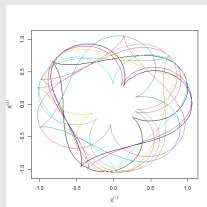
On a un contour lisse maintenant. . . on fait quoi avec une base de données ?

Avec différentes images, on observe les objets sujets à différentes orientations, positions ou échelles dans la photo.

Le point de départ ($t = 0$) du chemin est aléatoire dans ce contexte.

Un contour : une donnée fonctionnelle multivariée

Pour que les statistiques de formes soient intéressantes, il faut tout d'abord aligner les courbes.



Alignement fonctionnel des contours

Formes ou courbes ?

Comment différencier les deux ?

En bref, la courbe ou le contour, c'est ce qu'on observe dans l'image. La définition est un peu moins claire, mais ce sont nos données, notées $C(t)$.

Une forme, notée ($\tilde{C}(t)$), est une courbe plane fermée invariante sous des déformations établies (définition formelle de Kendall).

Alignement fonctionnel des contours

Pour obtenir un échantillon de **formes** ($\tilde{C}(t)$), il faut aligner les contours ($C(t)$), c'est-à-dire retirer l'effet des variables de déformation.

Celles-ci sont :

- ▶ Translation
- ▶ Échelle
- ▶ Rotation
- ▶ Le point de départ ($t = 0$) (paramétrisation)

Alignement fonctionnel des contours

Il existe quelques travaux en analyse de forme (Srivastava & Klassen, 2016) qui projettent les courbes sur *l'espace tangent*.

Cette projection retire complètement l'effet des déformations sur la courbe.

Par contre, cela nous empêche aussi d'en faire l'analyse par le fait même...

Si l'échelle d'un objet est d'intérêt, on perd cette information.

Alignement fonctionnel des contours

La procédure d'alignement proposée est d'estimer les déformations.

- ▶ Cela permet donc de les analyser comme variables d'intérêt,
- ▶ ainsi que de retirer leurs effets sur les courbes pour en obtenir la forme.

Modélisation du contour

Nous modélisons alors le contour observé par :

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

où $(\rho, \mathbf{O}, \gamma, \mathbf{T})$ sont les paramètres associés aux déformations et où $\tilde{\mathbf{C}}(t)$ est la forme.

Estimer les paramètres d'échelle et de translation

Estimer $\mathbf{T}_i = (T_x, T_y)$ et ρ_i est plutôt simple.

En supposant que nous voulons centrer nos formes à $(0, 0)$ et avoir une norme unitaire, nous voulons :

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

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

Estimer les paramètres d'échelle et de translation

Conséquemment, pour estimer $\mathbf{T}_i$ et ρ_i , nous calculons :

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

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

Ce calcul est rapide et peut être fait en parallèle sur notre échantillon de courbes.

Estimer les paramètres d'échelle et de translation : série de Fourier

Il est facile de déterminer la valeur de ces déformations étant donnée notre représentation par série de Fourier.

$$\mathbf{T}_i = \int_0^1 \mathbf{C}_i(t) dt = c_0$$

$$\rho_i = \|\mathbf{C}_i - \mathbf{T}_i\|_{\mathcal{H}} = \sqrt{\sum_{k=1}^M \|c_k\|_2^2}$$

Estimer les paramètres d'échelle et de translation

Après avoir estimé ces déformations, on les retire et on obtient $\mathbf{C}^*$, ce que nous appelons la pré-forme :

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

Estimer les paramètres de paramétrisation et de rotation

Il s'agit du plus grand défi méthodologique de l'approche proposée.

Le problème est qu'il faut estimer la paramétrisation et la rotation en même temps.

Paramétrisation : le point de départ

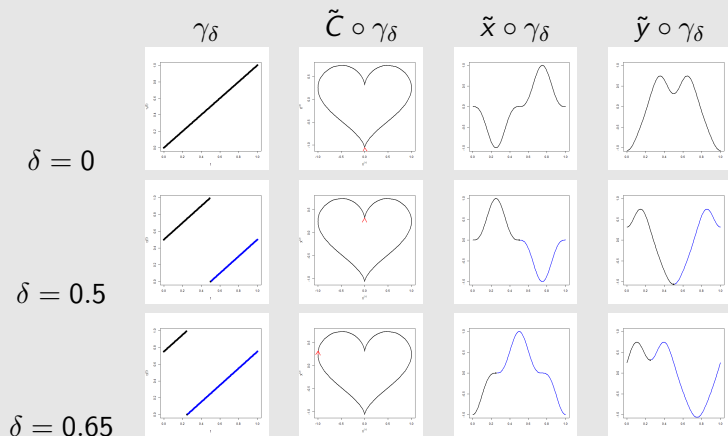
La paramétrisation γ est une simple fonction enveloppe (*wrapping function*) qui vise à modéliser différents points de départ possibles autour du contour.

On définit $\gamma \in \Gamma$ avec :

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

Paramétrisation : le point de départ

Plus simplement, on peut visualiser son effet ici :



Paramétrisation : le point de départ

Bref, comme nous analysons des courbes fermées, les fonctions coordonnées sont cycliques.

Le paramètre δ capture le point de départ sur le contour.

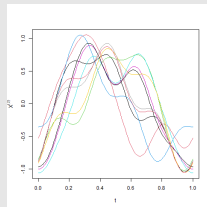
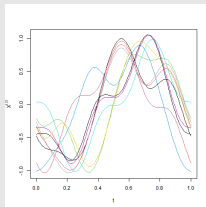
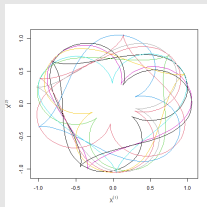
Paramétrisation : le point de départ

On pourrait peut-être continuellement ajuster δ jusqu'à aligner les points de départ.

Par contre, dépendamment de la rotation, les fonctions coordonnées sont entièrement différentes.

Il faut donc estimer ces deux déformations en tandem.

Effet de la rotation sur les fonctions coordonnées



Rotation

La rotation $\mathbf{O}$ de la pré-forme est modélisée par une matrice de rotation bien standard :

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

Conséquemment, il faut estimer θ .

Estimer les paramètres de paramétrisation et de rotation

Pour aligner les pré-formes $\mathbf{C}^*$, il nous faut un modèle (*template*) μ .

Ça peut être :

- ▶ Une observation aléatoire $\mathbf{C}_i^*$.
- ▶ Une moyenne quelconque.
- ▶ Une observation spécifique pour laquelle on aime l'alignement.

Estimer les paramètres de paramétrisation et de rotation

Étant donné un modèle μ , on cherche à aligner les pré-formes $\mathbf{C}_i^*$ en déterminant les valeurs des paramètres δ et θ qui alignent le mieux $\mathbf{C}_i^*$ avec μ :

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

Estimer les paramètres de paramétrisation et de rotation

Résoudre l'équation 3 est difficile. Il n'existe pas de solution analytique pour obtenir les deux paramètres.

Par contre, grâce à la représentation de Fourier, nous avons trouvé une solution pour δ étant donné θ et vice-versa.

Nous avons développé un algorithme itératif (similaire à l'ICP) qui met à jour les paramètres en alternance.

Estimer les paramètres de paramétrisation et de rotation

Finalement, on peut ainsi retirer les déformations et obtenir la forme de l'objet $\tilde{\mathbf{C}}$.

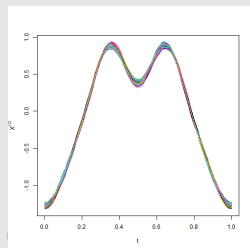
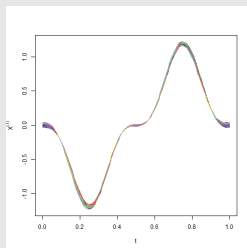
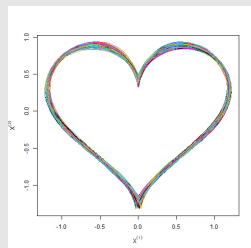
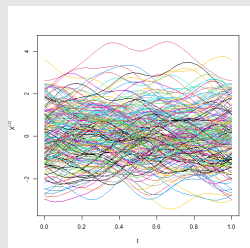
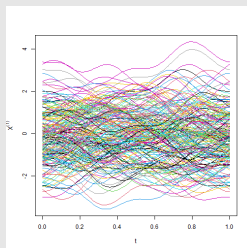
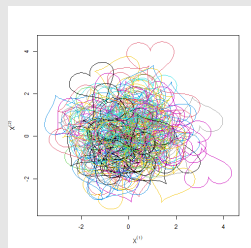
$$\mathbf{C}^*(t) = \frac{1}{\rho}(\mathbf{C}(t) - \mathbf{T})$$
$$\tilde{\mathbf{C}}(t) = \mathbf{O}_\theta \mathbf{C}^*(t) \circ \gamma_{(\delta)}$$

Résultats de l'alignement

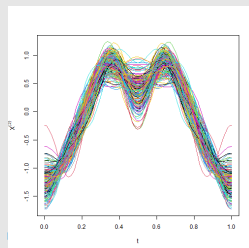
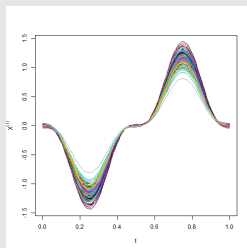
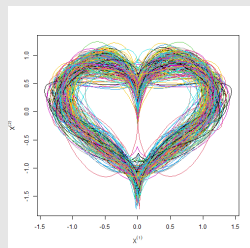
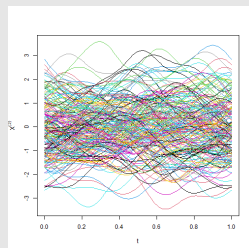
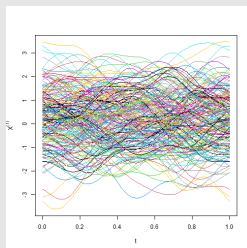
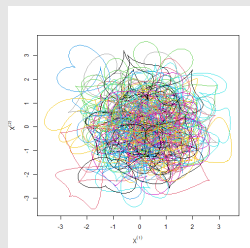
Avant d'analyser les formes obtenues, nous mettons à l'épreuve notre approche d'alignement,

à l'aide de simulations et de vraies données.

Résultats de l'alignement : données simulées



Résultats de l'alignement : données simulées



Résultats de l'alignement : données simulées

σ	EQM_{δ}	EQM_{θ}	$EQM_{\mathbf{T}}$	EQM_{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}

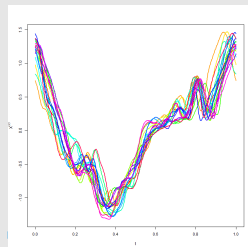
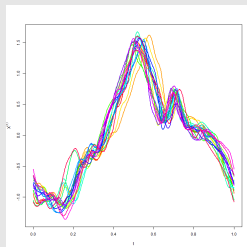
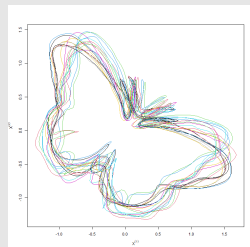
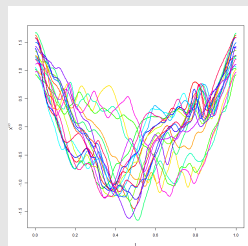
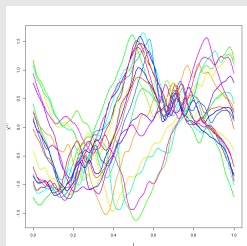
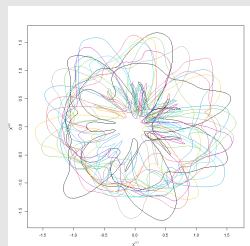
Résultats de l'alignement : vraies données

Base de données *MPEG-7* :



Figure: Exemples d'images provenant de la base de données *MPEG-7*.

Résultats de l'alignement : vraies données



Analyse statistique de formes

Analyse statistique de formes

- ▶ Jusqu'à présent, on n'a pas fait de *statistique*.
- ▶ On avait besoin de préparer les données afin de pouvoir les analyser.
- ▶ Mais c'était l'essence du problème. Maintenant que la modélisation est faite, on peut appliquer ces idées à toutes sortes de problèmes.
- ▶ On peut considérer plusieurs analyses statistiques de formes et analyser conjointement les déformations et les formes elles-mêmes.

Modéliser $\mathbf{C}$ à travers une analyse en composantes principales

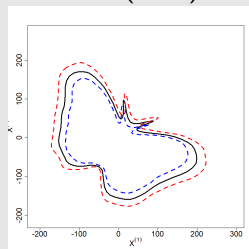
On propose donc une ACP conjointe (fonctionnelle et discrète).

On peut ainsi en extraire une projection intéressante pour des analyses supervisées et non supervisées.

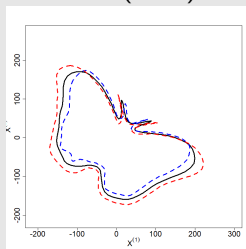
Voici quelques exemples :

Composantes principales fonctionnelles

PC1 (57%)



PC2 (22%)



PC3 (9%)

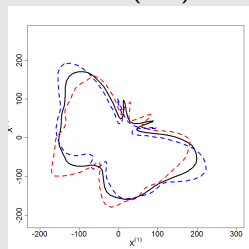


Figure: Figure des moyennes estimées $\bar{\mathbf{z}} = \sum_i \mathbf{z}_{i1}$ en noir, des $\bar{\mathbf{z}} - 20\hat{\phi}_k$ en bleu et des $\bar{\mathbf{z}} + 20\hat{\phi}_k$ en rouge.

Génération

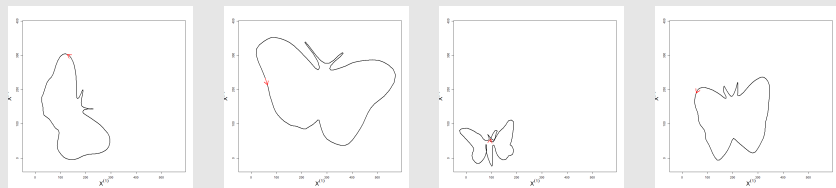


Figure: Courbes représentant des papillons générés par notre procédure.

Classification : Détection du mélanome

- ▶ Les données HAM10000 contiennent des photos de grains de beauté.
- ▶ On entraîne des modèles sur un échantillon de 8000 images de deux classes : mélanome et en santé.
- ▶ Le but est de comparer les approches fondées sur le pixel avec celles fondées sur la forme.

Classification : Détection du mélanome

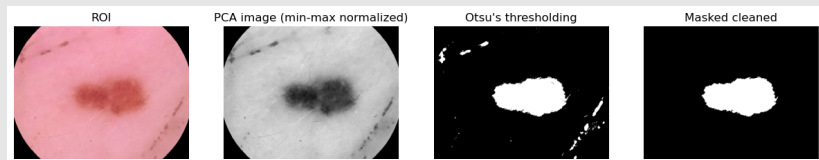


Figure: Extraction du contour.

Classification : Détection du mélanome

Modèle	avg-AUC	avg-Bal.Accuracy	avg-F1
Forme	0.640 $\pm$ 0.031	0.572 $\pm$ 0.036	0.837 $\pm$ 0.020
Pixel	0.697 $\pm$ 0.069	0.544 $\pm$ 0.049	0.814 $\pm$ 0.016

Table: Résultats des expériences (VC).

Classification : Détection du mélanome

Modèle	Mémoire (GB)	Temps d'entraînement (s)
Forme	0.7	31.7 ± 6.92
Pixel	66.7	174 ± 82.9

Table: Résultats des expériences (VC).

Extensions

Plusieurs contours en simultané

La suite de ce projet est d'étendre ce modèle à des objets plus complexes représentés par plusieurs contours.

Cela nous a forcés à revoir comment définir et estimer les déformations.

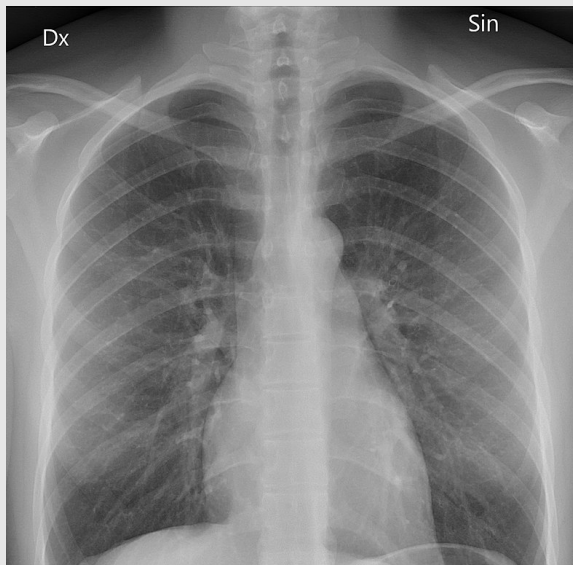
Nous avons des déformations globales comme la rotation, l'échelle et la translation, et une déformation locale : la paramétrisation.

Plusieurs contours en simultané

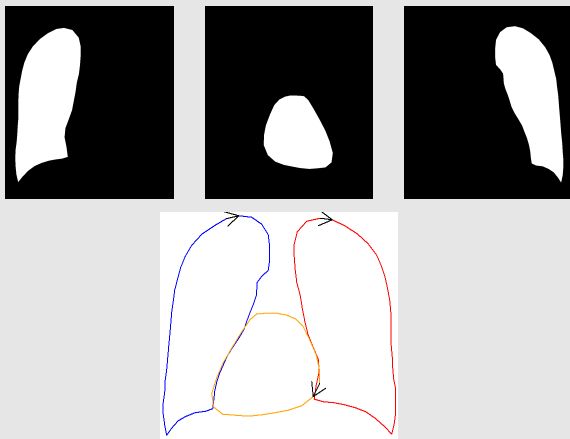
Une application de ce projet est la prédiction de la cardiomégalie ; une condition définie par un cœur trop gros proportionnellement aux poumons.

La forme des objets et leurs déformations sont donc des facteurs importants pour la classification.

Plusieurs contours en simultané



Plusieurs contours en simultané

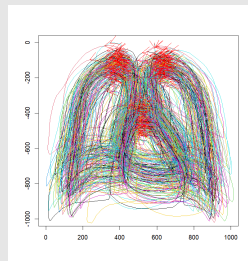
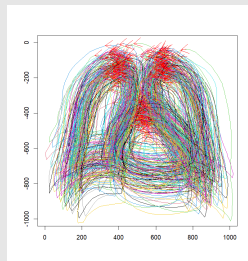


Plusieurs contours en simultan  : alignement

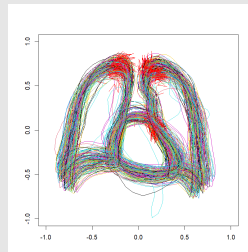
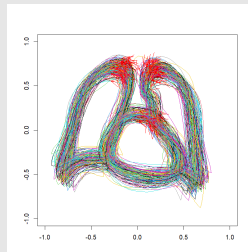
$Y = 1$

$Y = 0$

Contours
d'origine



Contours
align s



Plusieurs contours en simultané : comparaison avec d'autres alignements

		PLS	PCR	GL1	GL2
Scenario 1	ICF	85.96	83.35	77.21	79.24
	ICP	56.58	56.50	51.44	51.48
	ICPMO	65.05	62.81	60.85	60.03
Scenario 2	ICF	83.26	83.08	81.65	81.55
	ICP	52.55	52.87	52.33	52.55
	ICPMO	58.77	57.69	55.08	54.57

Table: Précision (%) à partir de la VC.

Intégration aux réseaux de neurones

On peut tout simplement intégrer les coefficients de la forme et les paramètres de déformation comme entrée ou sortie de réseaux de neurones.

On peut aussi mettre à profit les couches spécifiquement conçues pour les données fonctionnelles.

Finalement, on peut mettre au point des modèles conjoints qui utilisent la représentation de forme et de pixel en tandem.

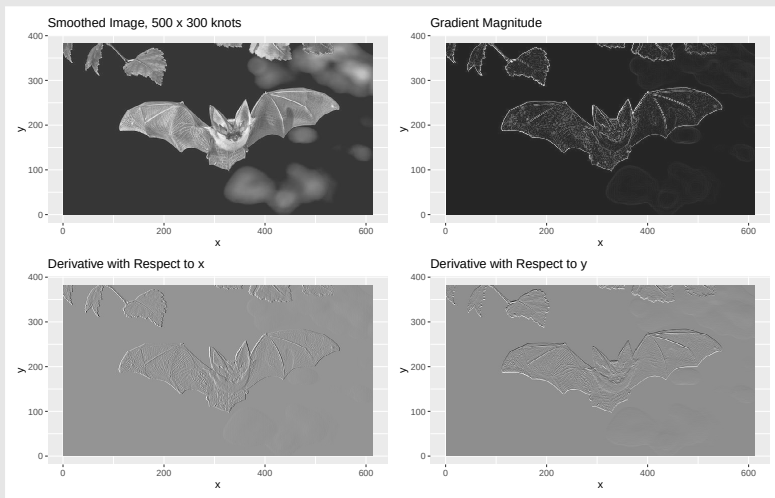
Travaux en cours

Travaux en cours : Images représentées comme surfaces

On vise maintenant à modéliser les images comme des surfaces, et donc les pixels comme une grille échantillonnée sur cette surface.

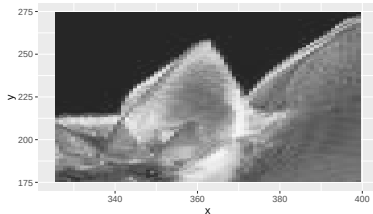
- ▶ On peut estimer une surface lisse à l'aide de *P-Splines* multidimensionnelles.
- ▶ Cela réduit dramatiquement la dimension de l'image.
- ▶ Permet des changements de résolution (non linéaires) instantanés.
- ▶ Offre un calcul rapide des dérivées ; essentiel pour la détection de contours.

Travaux en cours : Images représentées comme surfaces

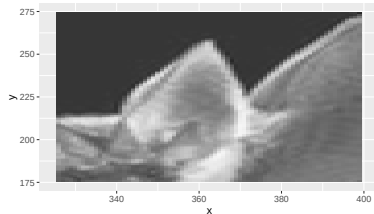


Travaux en cours : Images représentées comme surfaces

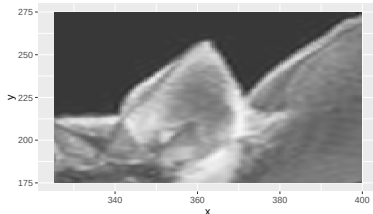
Original Image, 612 x 382



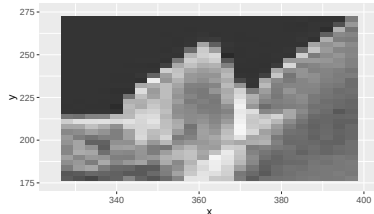
Smoothed Image (500 x 300 knots), 612 x 382



Smoothed Image, 1223 x 763



Smoothed Image, 204 x 128



Conclusion

L'analyse statistique de formes offre une nouvelle perspective pour l'analyse supervisée et non supervisée d'images.

À elle seule, la forme nous fournit une information interprétable souvent négligée dans les approches par pixels.

Afin d'analyser les formes dans les images, nous avons développé un algorithme d'alignement des contours.

Cet algorithme nous permet aussi d'estimer les paramètres de déformation.

Conclusion

On travaille à la mise au point d'une librairie R pour rendre toutes ces méthodes accessibles.

En traitant les images comme des surfaces, on espère améliorer notre procédure d'extraction de contours, mais aussi développer une nouvelle perspective d'analyse d'images.

En produisant une représentation parcimonieuse qui offre de multiples bénéfices sans inconvénients.

Michaël, je suis prêt pour les questions!

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

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

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.

θ en fonction de δ :

$$\tan(\theta_k) = \left(\sum_{j=1}^p \langle C_j^*, \bar{C}_j \circ \gamma_{\delta_j} \rangle_{\mathcal{H}} \right)^{-1} \sum_{j=1}^p \{ \langle X_j^*, \bar{X}_j \circ \gamma_{\delta_j} \rangle_{L_2} - \langle Y_j^*, \bar{Y}_j \circ \gamma_{\delta_j} \rangle_{L_2} \}.$$

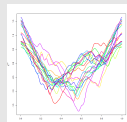
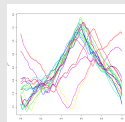
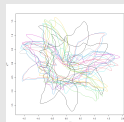
δ en fonction de θ :

$$\sum_{k \in \{1, 3, \dots, M-1\}} w_{j,k}^{1,\theta} \sin((k+1)\pi\hat{\delta}_j) = \sum_{k \in \{1, 3, \dots, M-1\}} w_{j,k}^{2,\theta} \cos((k+1)\pi\hat{\delta}_j), \quad (4)$$

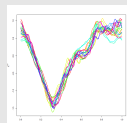
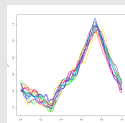
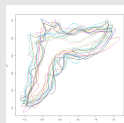
μ



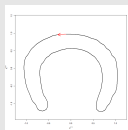
$\hat{\mathbf{x}}^*$



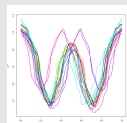
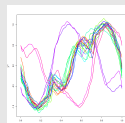
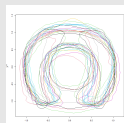
$\hat{\mathbf{x}}$



μ



$\hat{\mathbf{x}}^*$



$\hat{\mathbf{x}}$

