

Approche multi-échelles dans les espaces de formes.

DESWARTE Raphaël, LUU Keurcien, TRIAS Justin

29 juin 2011

Encadrants : TROUVÉ Alain, CHARON Nicolas.

Adresses électroniques des encadrants

Alain Trouvé : alain.trouve@cmla.ens-cachan.fr

Nicolas Charon : nicolas.charon@cmla.ens-cachan.fr

Adresses électroniques des auteurs

Raphaël Deswarte : raphael.deswarte@ens-cachan.fr

Keurcien Luu : keurcien.luu@ens-cachan.fr

Justin Trias : justin.trias@ens-cachan.fr

Remerciements

Nous adressons nos remerciements à nos encadrants Alain Trouvé et Nicolas Charon pour le temps qu'ils nous ont consacré, leur aide et leur disponibilité. Nous remercions aussi Yohann Simon qui nous a fourni nos données sur les souris et a accepté de venir nous exposer la problématique de ses recherches. Enfin, nous remercions les membres du laboratoire, en particulier son directeur et ses secrétaires, pour leur accueil.

Résumé Ce rapport présente différents appariements possibles (linéaire exact ou inexact, non linéaire), en utilisant des espaces à noyau reproduisant et des landmarks, et développe l'importance du paramètre d'échelle dans ces méthodes. Il présente également une approche nouvelle utilisant plusieurs échelles dans un même appariement. Enfin, une étude est menée sur des données de souris dont certaines étaient atteintes d'une maladie génétique et une analyse statistique des résultats est effectuée.

Table des matières

1 Introduction

De par leur propriétés géométriques, les formes ont dès l'Antiquité suscité l'intérêt des mathématiciens. On peut citer Euclide qui s'intéressait déjà aux polygones réguliers. Mais c'est seulement lors des dernières décennies que l'étude des formes en tant qu'objets mathématiques s'est développée. La notion de forme, bien qu'en apparence très intuitive, est un concept difficile à définir. Leur domaines d'utilisation sont divers et en pleine expansion, en imagerie médicale et notamment en biométrie, dont le développement a permis l'usage de nouveaux passeports avec empreintes digitales, ou encore les applications à la reconnaissance faciale (appareils photos détecteurs de visages).

La comparaison de formes nécessite une théorie adaptée. La simple comparaison pixel par pixel de deux photographies ne serait pas appropriée (deux images translattées d'un même objet seraient alors considérées comme deux formes totalement différentes). On impose alors que deux objets se déduisant d'une transformation appartenant à $SO_n(\mathbb{R})$ doivent être considérés comme identiques, et c'est pourquoi la notion de forme a été introduite : "Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object." (David George Kendall, 1984).

En fait les problématiques dépassent assez largement le cas de la prise en compte d'une invariance par rotation, translation et homothétie comme le fait Kendall. Il s'agit de comprendre comment analyser et décrire des populations de formes qui sont des structures de très grandes dimensions et pour lesquelles il faut définir des relations de proximité. Les approches par espaces de formes considèrent les formes elles-mêmes comme des points dans des espaces métriques, les comparaisons entre formes se faisant en calculant des distances. Kendall considère *in fine* la distance euclidienne dans la situation assez simple où les formes sont décrites comme un famille de points caractéristiques (appelés landmarks par la suite). Le choix de la distance euclidienne correspond à une situation où les déplacements des landmarks d'une forme à l'autre sont considérés de façon indépendante et cela ne correspond pas à la corrélation naturelle des déplacements des points géométriquement proches. D'autres métriques peuvent être construites en prenant en compte une distance caractéristiques d'interaction des points entre eux selon une échelle caractéristique λ . Dans la dernière décennie, une large famille d'espaces de formes munis d'une structure riemannienne a été introduite à partir de la notion de coût optimal de déformation d'une forme à une autre par une trajectoire de difféomorphismes agissant sur des structures d'intérêts variés (configuration d'images, surfaces segmentées, lignes,...) avec de nombreuses applications à la neuro-imagerie (voir par exemple la thèse de Stanley Durrleman, "Statistical models of currents for measuring the variability of anatomical curves, surfaces and their evolution"). Là encore une échelle caractéristique intervient dans l'analyse codant la régularité des transformations à travers l'introduction d'un espace à noyau reproduisant. Une question naturelle qui se pose est de construire des approches multi-échelles permettant des analyses plus riches.

Un des buts de notre stage a été de reprendre dans cette perspective les approches précédentes en essayant de proposer une analyse multi-échelle d'une population de formes à travers un modèle permettant d'analyser la variabilité en composant des déformations à différentes échelles. Si les analyses multi-échelles sont bien développées dans le cadre de l'analyse fonctionnelle avec les décompositions en ondelettes par exemple, la situation est très différente et semble-t-il beaucoup plus ouverte dans une situation où nous perdons la structure additive pour aller vers des structures non commutatives intervenant dans la composition des déformations non linéaires.

Nous nous sommes progressivement familiarisé avec l'état de l'art en reprenant les méthodes classiques linéaires pour aller progressivement vers méthodes non linéaires. Si un certain nombre de codes ont été développés pour traiter des situations difficiles de populations d'images 3D ou de surfaces 2D, ces codes ne sont pas facilement modifiables et il nous a été demandé de faire une implémentation complète en Matlab dans le cadre relativement restreint de populations de landmarks en 2D afin de bien maîtriser les codes et de pouvoir les expérimenter plus facilement.

Nous avons pu mettre en place une première extension multi-échelles des analyses de formes par déformations avec un traitement complet de la population de formes jusqu'à l'analyse statistique.

Nous avons en effet mis nos programmes et les théories étudiées en application dans le cadre de l'étude d'une maladie génétique chez les souris (maladie qui existe aussi chez l'homme). Cette maladie entraîne des modifications de la forme du crâne, il est alors intéressant de comprendre à quelles échelles ces modifications interviennent et de savoir s'il est possible de discerner ceux qui sont atteints de la maladie de ceux qui sont sains. Nous avons réalisé une analyse statistique des résultats obtenus, en utilisant des techniques classiques (Analyse en Composantes Principales, "leave-one-out", test de permutation) ou beaucoup plus récentes (Analyse de Fisher Discriminante avec utilisation des noyaux pour 3 classes). Les résultats obtenus avec le programme d'appariement non linéaire sont meilleurs que ceux obtenus précédemment avec la méthode de l'appariement linéaire : nous avons réussi à classer les 3 groupes d'individus (sains, mutants, hétérozygotes) avec moins d'erreurs que ce qui avait été fait jusqu'alors. La classification étant satisfaisante, cela montre que la mutation agit à toutes les échelles étudiées.

2 Méthode linéaire.

2.1 Espaces à noyau reproduisant.

Dire que les méthodes linéaires sont des méthodes pour lesquelles les appariements se ramènent à un problème d'interpolation de champs de vecteurs en dehors de positions connues (landmarks) sur lesquels les valeurs du champ sont connues de façon exacte ou approchée. Ce point de vue utilise la théorie des espaces à noyau reproduisant pour construire les contrôles de régularité des champs interpolants. Les noyaux permettent alors de ramener les problèmes à des problèmes linéaires très faciles à expliciter et à résoudre et fournissant un point de départ vers des méthodes non-linéaires plus sophistiquées qui seront introduites par la suite. La théorie des espaces à noyau reproduisant nous était inconnue et nous avons dû nous familiariser avec elle au début du stage. Nous donnons ci-dessous un bref aperçu de la théorie avec les éléments principaux que nous avons dû utiliser. Notons que les espaces à noyau ont connu une intense activité à partir de la fin des années 90 dans la communauté du *machine learning* avec le développement des machines à vecteurs de support.

On souhaite travailler avec un espace de fonctions régulières qui tiennent compte des invariances par translation et rotation. Le choix d'un espace reproduisant associé à un noyau vectoriel radial s'avère alors satisfaisant. Commençons par donner quelques définitions et propriétés qui nous serviront par la suite.

Définition 1 Soit $(V, \langle \cdot, \cdot \rangle_V)$ un espace de Hilbert de champs de vecteurs définis sur \mathbb{R}^2 . On dit que V est à noyau reproduisant si : $\forall x, a \in \mathbb{R}^2$, l'application $\delta_x : f \in V \mapsto f(x) \in \mathbb{R}^2$ est linéaire continue. Il existe alors $K_V : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto M_2(\mathbb{R})$ tel que :

$$(i) \quad \forall y \in \mathbb{R}^2, K_V(\cdot, y) : x \mapsto K_V(x, y) \in V.$$

$$(ii) \quad \forall v \in V, \forall \alpha \in \mathbb{R}^2, \forall y \in \mathbb{R}^2, (v(y), \alpha)_{L^2} = \langle v, K_V(\cdot, y) \rangle_V.$$

De (i) et (ii) on déduit la propriété (iii) dite "reproduisante" du noyau k :

$$(iii) \quad \forall (x, y) \in \mathbb{R}^2, \forall (\alpha, \beta) \in \mathbb{R}^2 \times \mathbb{R}^2, \alpha^t K_V(x, y) \beta = \langle K_V(x, \cdot) \alpha, K_V(y, \cdot) \beta \rangle_V.$$

Un noyau vectoriel radial est de la forme $K_V : (x, y) \in V \times V \mapsto h(|x - y|)I_2$ où h vérifie certaines conditions que l'on suppose satisfaites. Si τ est une translation, on a bien $K_V(\tau(x), \tau(y)) = K_V(x, y)$ et de même si r est une rotation, $K_V(r(x), r(y)) = K_V(x, y)$. Choissant $h : t \mapsto \exp(-\frac{t^2}{2\lambda^2})$, on introduit le noyau vectoriel gaussien :

$$\forall (x, y) \in \mathbb{R}^2 \times \mathbb{R}^2, K_V(x, y) = \exp(-\frac{|x - y|^2}{2\lambda^2})I_2 \in M_2(\mathbb{R}).$$

Cette application n'est pas invariante par changement d'échelle et l'introduction du paramètre λ permet de choisir l'échelle d'observation souhaitée.

Proposition 1 K_V est un noyau défini positif strict i.e. :

$$\forall n \in \mathbb{N}, \forall (x_i, \epsilon_i)_{1 \leq i \leq n}, x_i, \epsilon_i \in \mathbb{R}^2, \sum_{i=1}^n \sum_{j=1}^n \epsilon_i \epsilon_j K_V(x_i, x_j) \geq 0. \quad (1)$$

Par ailleurs, si les x_i sont distincts 2 à 2 alors (1) ne s'annule que lorsque les ϵ_i sont tous nuls.

Un des résultats centraux de la théorie des espaces à noyau reproduisant est que la donnée d'un noyau positif K_V induit l'existence d'un espace de Hilbert pour lequel le noyau associé est K_V . Ainsi la donnée d'un noyau définit un espace à noyau reproduisant et la construction se fait donc du noyau vers l'espace ce qui permet de construire une très grande variété d'espaces. Ici nous nous restreindrons aux noyaux gaussiens mais pour lesquels nous avons un paramètre d'échelle qui jouera un grand rôle dans la suite.

2.2 Appariement exact.

Nous nous intéressons ici à l'appariement exact de formes. Les formes que l'on considère sont des ensembles finis de points de \mathbb{R}^p . On fixe alors à n leur nombre de points. Partant d'une forme initiale (x_1, \dots, x_n) et d'une forme cible (y_1, \dots, y_n) , on cherche à déterminer la transformation $v \in V$ du plan vérifiant :

$$v = \operatorname{argmin}_{v \in V} \{ \|v\|_V \mid \forall i, v(x_i) = y_i - x_i, 1 \leq i \leq n \}.$$

Il s'agit d'un problème de minimisation sous contraintes visant à construire un champ dense (c'est-à-dire défini partout) à partir de valeurs prescrites sur les points constituant la forme initiale. On montre facilement que v est de la forme :

$$v : y \mapsto \sum_{i=1}^n K_V(y, x_i) \alpha_i.$$

Alors $(\alpha_1, \dots, \alpha_n)$ vérifie le système d'équations (P) :

$$\sum_{i=1}^n K_V(x_j, x_i) \alpha_i = c_j, 1 \leq j \leq n.$$

$S_V(x)$ est la matrice par blocs constituée des matrices $K_V(x_i, x_j)$ de taille $p \times p$ et permet de réécrire (P) sous la forme :

$$S_V(x) \alpha = c$$

où $\alpha = (\alpha_1, \dots, \alpha_n)$ et $c = (c_1, \dots, c_n)$.

$$S_V(x_1, \dots, x_n) = \begin{pmatrix} 1 & 0 & \exp(-\frac{|x_1-x_2|^2}{2\lambda^2}) & 0 & \dots & \dots & \dots \\ 0 & 1 & 0 & \exp(-\frac{|x_1-x_2|^2}{2\lambda^2}) & \dots & \dots & \dots \\ \exp(-\frac{|x_1-x_2|^2}{2\lambda^2}) & 0 & 1 & 0 & \dots & \dots & \dots \\ 0 & \exp(-\frac{|x_1-x_2|^2}{2\lambda^2}) & 0 & 1 & & & \\ \vdots & & & & \ddots & & \\ \vdots & & & & & \ddots & 0 \\ \vdots & & & & & & 1 \end{pmatrix}.$$

On peut montrer que dans le cas d'un noyau radial comme c'est le cas ici, la matrice $S_V(x)$ est inversible. Tout d'abord on remarque que, posant $A = (k(x_i, x_j))_{1 \leq i, j \leq n}$, $S_V(x)$ est semblable à la matrice par blocs :

$$\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}.$$

Il suffit alors uniquement de montrer que A est inversible pour avoir l'inversibilité de $S_V(x)$. Soit $b \in \mathbb{R}^n$ tel que $Ab = 0$. Ceci se réécrit

$$\forall j, 1 \leq j \leq n, \sum_{i=1}^n k(x_i, x_j) b_i = 0$$

et on obtient donc

$$\sum_{j=1}^n \sum_{i=1}^n k(x_i, x_j) b_i b_j = 0$$

en multipliant par b_j chacune de ces équations et en les additionnant.

Les x_i étant tous distincts, la **Proposition 1** assure que $b = 0$, ce qui prouve que A est inversible. En conclusion, $v = \sum_{i=1}^n K_V(\cdot, x_i) \alpha_i$ où $\alpha = S_V(x)^{-1} c$.

La norme de v se calcule alors aisément à partir de cette expression :

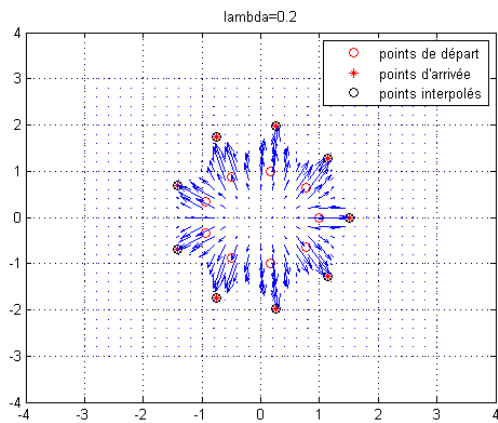
$$\begin{aligned} \|v\|_V^2 &= \left\langle \sum_{i=1}^n K_V(\cdot, x_i) \alpha_i, \sum_{j=1}^n K_V(\cdot, x_j) \alpha_j \right\rangle_V \\ &= \sum_{i=1}^n \sum_{j=1}^n \langle K_V(\cdot, x_i) \alpha_i, K_V(\cdot, x_j) \alpha_j \rangle_V \\ &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i^t K_V(x_i, x_j) \alpha_j \quad (\text{d'après la propriété (i) de } K_V) \\ &= \alpha^t S_V(x) \alpha \\ &= c^t S_V(x)^{-1} c \end{aligned}$$

On peut vérifier que v est bien de norme minimale parmi les éléments ϕ de V vérifiant : $\forall i, \phi(x_i) = y_i - x_i, 1 \leq i \leq n$. Une preuve succincte de cette affirmation est donnée en annexe pour s'en convaincre.

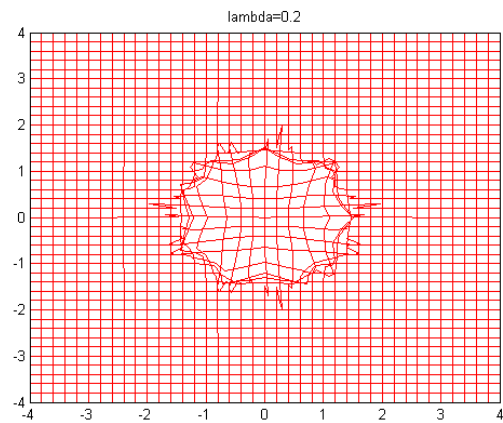
Visualisation d'un exemple et de l'influence du paramètre λ .

L'ajustement du paramètre λ donne la possibilité de contrôler les degrés d'interaction entre les différents points considérés. Observons maintenant l'exemple d'un cercle de rayon 1 que l'on transforme via une homothétie en un cercle cible de rayon 1.5. Les cercles rouges, les étoiles rouges, et les cercles noirs représentent respectivement la forme initiale x , la forme cible y et la forme $x + v(x)$. Bien entendu dans cette partie y coïncide avec $x + v(x)$ puisque c'est ce que l'on impose au champ de déplacement. La longueur des flèches bleues traduisent quant à elles la norme du déplacement de chaque point du quadrillage.

Sur la première figure, on s'aperçoit que seuls les points très proches de la forme initiale se déplacent. On peut estimer qu'à une distance de la forme initiale supérieure à 3λ , $v \simeq 0$. La figure de droite montre la déformation induite par v . On notera les irrégularités de la grille ainsi déformée, le paramètre λ n'est pas donc pas judicieux puisqu'il ne préserve pas le caractère lisse du cercle. On remarque aussi que la déformation "fait des plis" au niveau des landmarks ce qui correspond à un non inversibilité de la transformation. Ce phénomène est la conséquence de la présence de fortes valeurs des dérivées premières de la transformation qui ne permet pas de garantir le caractère inversible. Seules les approches non linéaires que nous verrons par la suite permettrons de contourner ce problème.



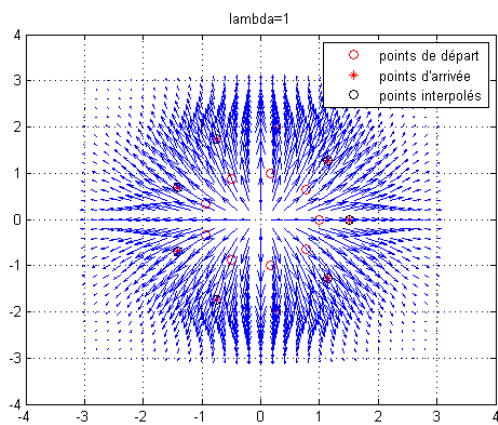
(a) Champ de vecteur



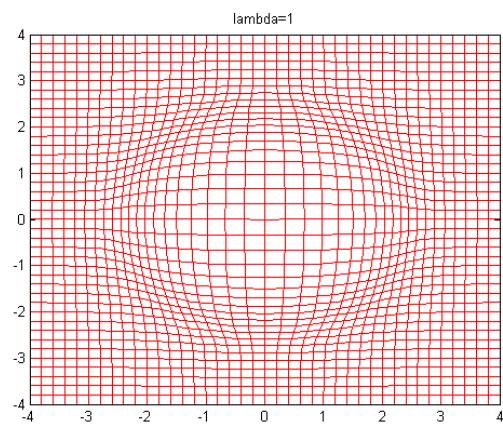
(b) Grille de déformation associée

FIGURE 1 – $\lambda = 0.2$

Dans la figure 2, on considère cette fois-ci une échelle $\lambda = 1$ plus grande que la précédente. Le champ de déplacement a ici visiblement une meilleure régularité (Figure 2), tout point situé sur le cercle de départ, landmark ou non, est envoyé via v sur le cercle cible. Ceci est dû au fait que dans ce cas, on autorise le déplacement de points qui étaient trop éloignés pour l'échelle $\lambda = 0, 2$, les landmarks ne sont plus indépendants et toute la régularité de la forme de départ est restituée.



(a) Champ de vecteur



(b) Grille de déformation associée

FIGURE 2 – $\lambda = 1$

2.3 Appariement inexact.

La méthode précédemment exposée réalise un "appariement exact" (*exact matching*), elle consiste à envoyer précisément les points de départ sur les points d'arrivée. Or dans les applications pratiques, ce n'est pas souhaitable : les données utilisées sont bruitées, elles peuvent contenir des erreurs (par exemple des erreurs de pointage sur les radiographies de souris). Ces erreurs peuvent déplacer deux points, en théorie très proches, dans des directions différentes, ce qui ajoute un coût artificiel qui n'a pas lieu d'être.

Aussi allons-nous chercher à régulariser nos données, à les "lisser". Pour ce faire, on utilise un "appariement inexact" (*inexact matching*) : on donne la possibilité au champ de déplacement de ne pas envoyer exactement les points de départ sur les points d'arrivée théoriques, mais on rajoute un coût de pénalité proportionnel à la distance euclidienne entre points d'arrivée théoriques et points d'arrivée calculés. Le coefficient de proportionnalité μ , appelé "attache aux données", est un paramètre fixé par l'utilisateur, selon la proximité qu'il souhaite obtenir par rapport aux données théoriques : en faisant croître μ , on devrait se rapprocher du cas de l'appariement exact (cas correspondant à $\mu \rightarrow \infty$).

On cherche donc à minimiser la quantité :

$$J(v) = \frac{1}{2} \|v\|_V^2 + \mu \sum_{i=1}^N |y_i - (x_i + v(x_i))|^2$$

où v est de la forme :

$$y \mapsto v(y) = \sum_{i=1}^N K_V(y, x_i) \alpha_i$$

et $\|v\|_V$ est la norme du champ de déplacement dans l'espace V calculée comme en 2.2, et $(x_i + v(x_i))$ est le point d'arrivée i calculé.

$$\text{Notons } f(\alpha) = \frac{1}{2} \alpha^t S_V(x) \alpha + \mu \sum_{i=1}^N |y_i - x_i - \sum_{j=1}^N K_V(x_i, x_j) \alpha_j|^2.$$

La différentielle de f au point α vaut alors pour tout $h \in \mathbb{R}^{2N}$:

$$Df_\alpha(h) = \alpha^t S_V(x) h - 2\mu(y - x)^t S_V(x) h + 2\mu \alpha^t S_V(x)^t S_V(x) h.$$

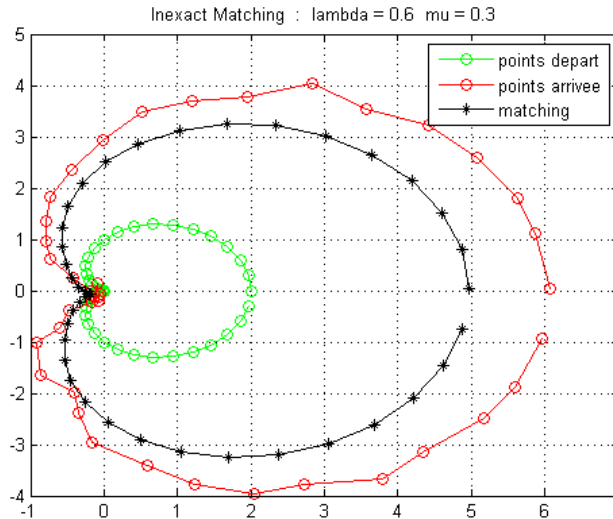
Ainsi $Df_\alpha = 0$, si, et seulement si $\alpha^t \left(\frac{I_{2N}}{2\mu} - S_V(x) \right) = (y - x)^t$, ce qui nous permet de déterminer α via la résolution du système d'équations linéaires :

$$\sum_{i=1}^N (K_V(x_j, x_i) + \frac{1}{2\mu} I_2) \alpha_i = c_j.$$

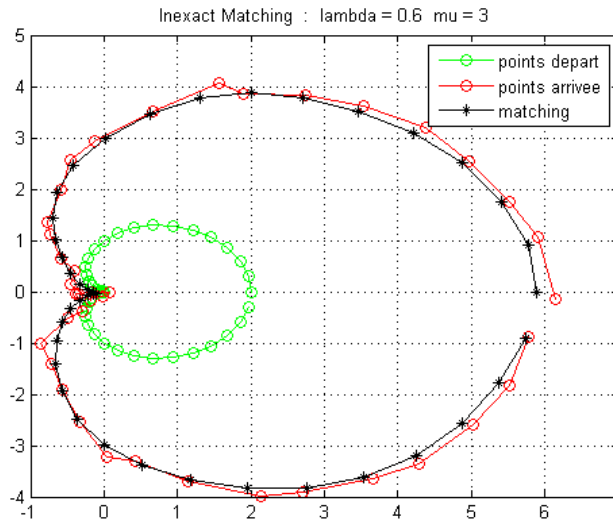
L'existence et l'unicité d'une telle solution se prouvent de la même façon qu'en 2.2. Puisque la somme d'un noyau défini positif strict et d'un noyau défini positif est un noyau défini positif strict, il suffit d'adapter la démonstration au noyau défini positif strict $K_V + \frac{1}{2\mu} I_p$. En faisant tendre μ vers l'infini, on retombe effectivement sur le système d'équations (P) qu'on a défini dans la partie précédente, ce qui est cohérent avec l'idée intuitive qu'une pénalité "infinie" correspond au cas de l'appariement exact.

Visualisation d'un exemple et de l'influence du paramètre μ .

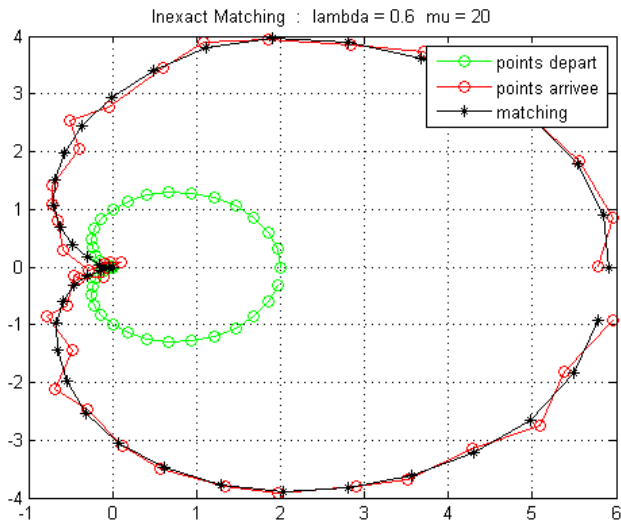
On tente ici d'apparier, avec un paramètre d'échelle raisonnable ($\lambda = 0.6$) la cardioïde verte de départ avec la cardioïde bruitée rouge. Pour un plus de clarté, on a décidé de relier les landmarks entre eux afin de donner l'allure de la cardioïde, mais les traits dessinés n'ont aucune signification théorique.



Pour une attache aux données faible ($\mu = 0.3$) on obtient un résultat noir très lisse (la déformation étant ainsi peu coûteuse en énergie) mais loin des points d'arrivée théoriques (ce qui n'est pas gênant vu que la pénalité est légère).



En augmentant la pénalité ($\mu = 3$), on observe que les points résultants se rapprochent des points d'arrivée.



Avec un μ important (ici $\mu = 20$), le résultat est très proche des points d'arrivée mais la forme obtenue est nettement plus régulière.

3 Méthode non linéaire.

3.1 Appariement non linéaire.

La méthode linéaire consistait à trouver un champ de déplacement qui appariait les points de départ et d'arrivée, le problème majeur de cette méthode provient des champs de déplacement eux-mêmes, puisque, en réalité, rien n'assure qu'ils soient inversibles, en particulier si l'on n'est pas dans le cas de petites déformations.

Pour éviter ce désagrément, nous allons chercher un champ de déplacement difféomorphe en intégrant un champ de vitesse le long d'une trajectoire.

A toute trajectoire dans l'espace des landmarks $t \rightarrow x(t)$ est associé un champ de vitesse d'énergie minimale

$$v_x(t)(y) = \sum_{i=1}^N K_V(y, x_i(t)) \alpha_i(t)$$

où les α proviennent de la résolution de l'équation $\dot{x}(t) = S_V(x(t))\alpha(t)$.

Ici, les points de départ correspondront au temps initial à $x(0)$ et les points d'arrivée que l'on vise correspondront au temps final à $x(1)$, le temps variant de 0 à 1. Il faut maintenant minimiser le coût de cette transformation, qui est donné par :

$$C(x, \alpha) = \frac{1}{2} \int_0^1 \|v_x(t)\|_V^2 dt + \mu \sum_{i=1}^N |y_i - x_i(1)|^2$$

Le problème d'appariement est en fait un problème d'optimisation sur les α , il peut se réécrire sous la forme :

$$\left\{ \begin{array}{l} \operatorname{argmin} \int_0^1 L(x, \alpha) dt + E(x(1)) \\ \dot{x} = f(x, \alpha) = S_v(x)\alpha \\ x(0) \text{ fixé} \end{array} \right.$$

Conformément à la théorie du contrôle optimal, l'introduction d'une fonction auxiliaire p , et d'un hamiltonien associé $H(x, p, \alpha) = (p|\dot{x} - f(x, \alpha)) - L(x, \alpha)$, permet de simplifier les équations grâce au théorème suivant.

L'idée est la suivante : si l'on note $J(x, \alpha) = \int_0^1 L(x, \alpha) dt + E(x(1))$ sous la condition $\dot{x} = f(x, \alpha)$ et $x(0)$ fixé, on peut, en introduisant un multiplicateur de Lagrange, noté $p(t)$, se ramener à chercher les solutions critiques de

$$\begin{aligned} J_E(x, p, \alpha) &= \int_0^1 (L(x, \alpha) + (p|\dot{x} - f(x, \alpha))) dt + E(x(1)) \\ &= \int_0^1 ((p|\dot{x}) - H(x, p, u)) dt + E(x(1)) \end{aligned}$$

Le problème se résume alors à trouver les solutions telles que :

$$\frac{\delta J_E}{\delta \alpha} = \frac{\delta J_E}{\delta p} = \frac{\delta J_E}{\delta x} = 0$$

en ajoutant une nouvelle contrainte cette fois-ci pour "éliminer" le terme d'attache aux données en imposant $p(1) + DE(q(1)) = 0$.

On peut résumer tout ceci via le théorème suivant :

"Théorème" 1 *Considérant la fonctionnelle $J(\alpha) = \int_0^1 C(x, \alpha)dt + g(x(1))$ avec $x = \Phi(\alpha)$ et p solution rétrograde de $p_1 + \frac{dg}{dx}(q(1)) = 0$ et $\dot{p} = -\frac{\partial H}{\partial x}(x, p, \alpha)$.*

Si α atteint le minimum de J alors

$$\frac{\partial H}{\partial \alpha}(x, p, \alpha) = 0.$$

Les solutions qui minimisent J vérifient

$$\left\{ \begin{array}{l} \dot{x} = \frac{\partial H}{\partial p}(x, p, \alpha) \\ \dot{p} = -\frac{\partial H}{\partial x}(x, p, \alpha) \\ \frac{\partial H}{\partial \alpha}(x, p, \alpha) = 0 \end{array} \right.$$

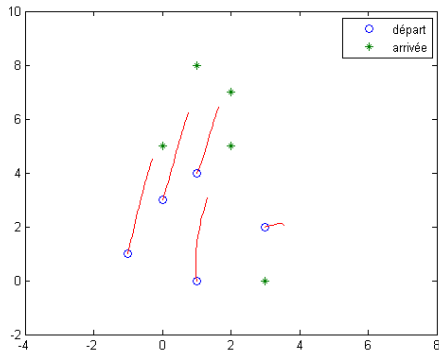
La méthode numérique utilisée dans les algorithmes met en place une descente de gradient sur le terme d'énergie C par rapport à α . Ainsi, dans un premier temps, x peut être calculé à partir de α grâce à l'équation $\dot{x} = S_V(x)\alpha$ et aux conditions initiales, qui ne sont autres que les points de départ, ce qui fixe $x(0)$. Dans un deuxième temps, p est calculé de manière rétrograde à partir de α et de x grâce à l'équation $\dot{p} = -\frac{\partial H}{\partial x}(x, p, \alpha)$ et à la condition finale $p(1) = -2\mu(y - x(1))$.

Enfin la descente de gradient s'effectue dès lors que le calcul de p est achevé, étant donné que le gradient sur C en α est donné par la différence $\alpha - p$.

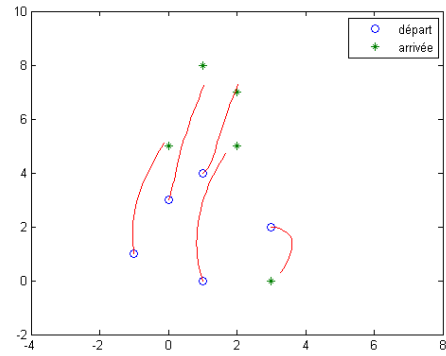
L'appariement va donc être réalisé grâce à cette méthode de descente de gradient sous une réécriture hamiltonienne du problème d'optimisation de transformation. Nous présentons ici quelques résultats obtenus pour différentes valeurs du paramètre d'échelle et d'attache aux données. Le choix de l'échelle est crucial, puisque, pour une échelle trop petite, les points ne seront pas corrélés, ils se déplaceront de manière indépendante, et le système sera donc - trop - souple. Inversement, pour une échelle trop grande, tous les points seront liés, ils se déplaceront tous ensemble, et le système sera alors - trop - rigide.

L'influence du paramètre d'attache aux données est quant à elle tout autre. Dans le terme d'énergie, plus ce paramètre est grand, plus le terme qu'il a en facteur devra être petit. Ce terme est fonction de la distance des points d'arrivée de la transformation aux points théoriques que l'on veut apparier. Attribuer une grande valeur à μ aura donc pour effet d'amener les points de départ très proches des points d'arrivée, la transformation sera moins "lisse", tandis qu'une valeur faible de μ favorisera des transformations lisses qui donneront parfois des points résultants assez éloignés des points théoriques d'arrivée. Encore une fois, le choix de ce paramètre est essentiel pour concilier régularité et précision des transformations. A noter toutefois que, le choix de celui-ci se révèle parfois assez délicat pour deux raisons : la première est qu'il modifie le gradient sur l'énergie, il faut donc se prémunir de ces effets néfastes en changeant le pas de gradient sur les α dans la méthode présentée ; la deuxième est qu'il est beaucoup plus difficile à intuituer que le paramètre d'échelle.

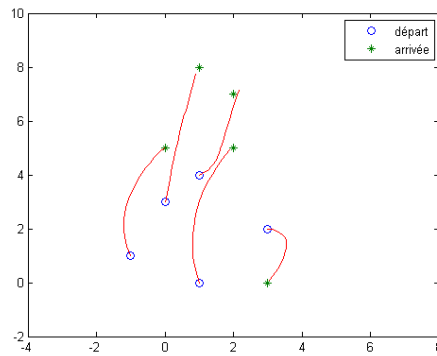
Visualisation d'un exemple dans le cas de l'appariement non linéaire.



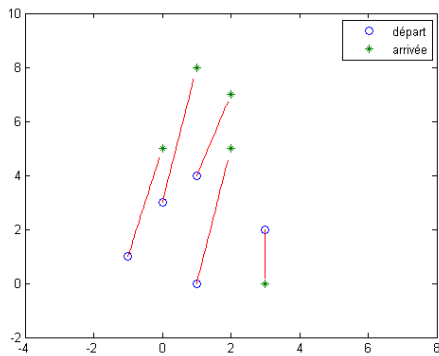
(d) $\lambda = 2 ; \mu = 1$



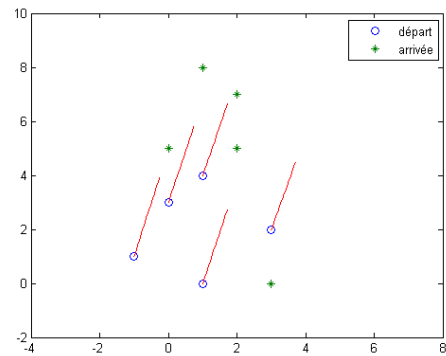
(e) $\lambda = 2 ; \mu = 10$



(f) $\lambda = 2 ; \mu = 20$



(g) $\lambda = 0.1 ; \mu = 10$



(h) $\lambda = 10 ; \mu = 1$

FIGURE 3 – Exemple d'appariement non linéaire pour différentes valeurs des paramètres λ et μ

La figure (f) montre que pour une valeur élevée de $\mu = 20$, l'appariement est quasi-exact tandis que pour une valeur plus faible comme en figure (d) ou (h) où $\mu = 1$, l'appariement est insatisfaisant, les points d'arrivée et les points cibles sont trop beaucoup trop éloignés pour prétendre à une considération ultérieure du champ de déplacement en question dans le but de comparer les formes. Quant au paramètre λ , plus sa valeur est grande, et plus les interactions entre les points sont fortes, cela confère à la forme une rigidité qui oblige chacun des landmarks à se déplacer via une même translation (figure (h)). Les landmarks peuvent se déplacer en ligne

droite indépendamment les uns des autres également si λ est trop faible, annulant les interactions. Le choix de ces paramètres est donc important si l'on souhaite réaliser des appariements qui ont une cohérence pour la suite de notre étude (figure(f)).

3.2 Multi-échelle.

Les caractéristiques propres des formes sont en fait contenues dans les transformations qui permettent de passer de l'une à l'autre et ce, à différents niveaux, i.e. à différentes échelles. C'est pourquoi il serait intéressant d'essayer d'obtenir ces informations en une seule et unique transformation.

Soient m paramètres d'échelles $(\lambda_k)_{1 \leq k \leq m}$ et $(t_k)_{0 \leq k \leq m}$ une subdivision de l'intervalle de temps $[0, 1]$, l'idée est maintenant de changer le paramètre d'échelle λ au cours du temps pour étudier le comportement de la transformation dont l'échelle varie. Sur chaque intervalle de temps $[t_{k-1}, t_k]$ de la subdivision, l'échelle est donnée par le λ_k correspondant, ainsi cette approche multi-échelle permet d'analyser les formes et leurs classes à des échelles variables.

L'expression du coût à minimiser est semblable à celle vue précédemment, à ceci près que le lambda varie au cours du temps.

$$\begin{aligned} C(x, \alpha) &= \frac{1}{2} \int_0^1 \|v_x(t)\|_{\lambda(t)}^2 dt + \mu \sum_{i=1}^N |y_i - x_i(1)|^2 \\ &= \sum_{k=1}^m \frac{1}{2} \int_{t_{k-1}}^{t_k} \|v_x(t)\|_{\lambda_k}^2 dt + \mu \sum_{i=1}^N |y_i - x_i(1)|^2 \end{aligned}$$

La forme des équations à résoudre ne change pas, elles restent les mêmes si l'on prend en compte que le paramètre d'échelle change à chaque t_k . Par exemple, le gradient par rapport à α sur le terme d'énergie est toujours $\alpha - p$, il suffit donc d'appliquer à nouveau l'algorithme précédent en intégrant les trajectoires avec un λ variable. Nous avons donc inclus, dans la méthode numérique que nous avons implémentée, des schémas de résolution d'équations différentielles pour une liste de $(\lambda_k)_{1 \leq k \leq m}$ et de $(t_k)_{0 \leq k \leq m}$. Il suffit de changer tout simplement l'équation à résoudre à chaque changement de λ , i.e. à tous les t_k .

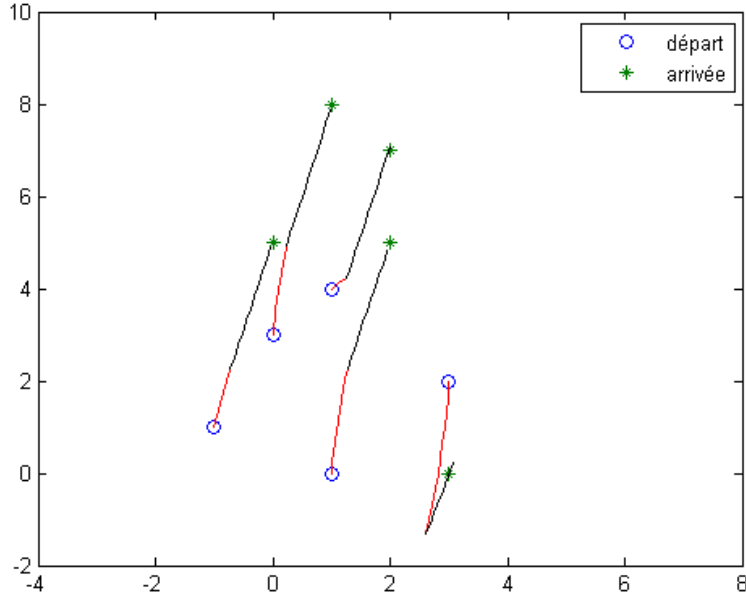
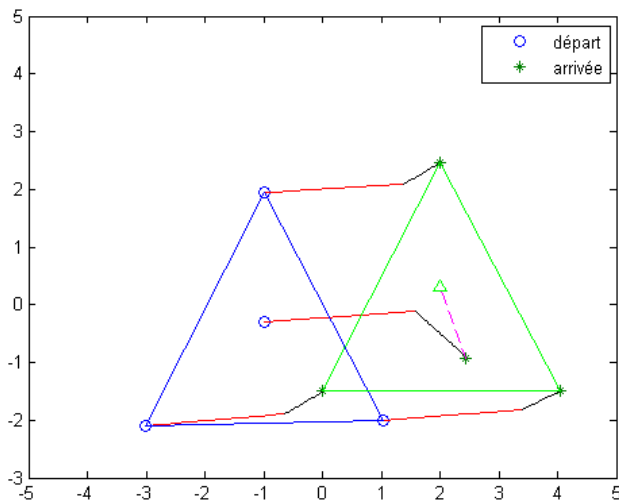
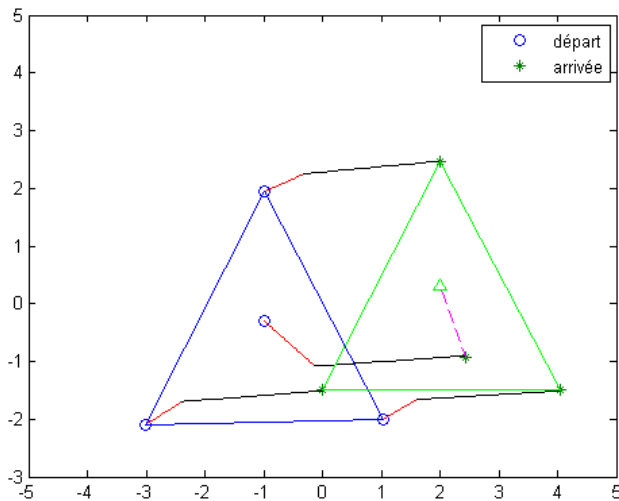


FIGURE 4 – Exemple d'un appariement multi-échelle pour $\lambda = (0.1, 10)$, $T = (0, 0.5, 1)$ et $\mu = 20$



(a) $\lambda = (6, 0.1)$, $T = (0, 0.5, 1)$ et $\mu = 100$



(b) $\lambda = (0.1, 6)$, $T = (0, 0.5, 1)$ et $\mu = 100$

Les figures précédentes montrent 2 triangles translétés vers le haut et la droite (transformation à grande échelle), avec un point supplémentaire : au centre pour le premier triangle, décalé vers le bas et la droite du centre pour le deuxième triangle (dont le centre a été représenté avec un triangle, sans l'utiliser dans l'algorithme), ce décentrage du quatrième point correspond donc à une transformation à petite échelle.

Les échelles utilisées sont : une grande puis une petite (première figure), une petite puis une grande (deuxième figure). On constate que dans les deux cas, l'essentiel de la translation commune aux points est assuré par la grande échelle, alors que la petite échelle effectue le décentrage du quatrième point. On a donc un effet de "filtre" des transformations par les différentes échelles.

3.3 Moyenne de Karcher.

Etant donnée une famille de formes, il reste maintenant à trouver comment les classer en les discriminant notamment à plusieurs échelles. Cependant, avant même de s'intéresser à la classification de celles-ci à proprement parler, il faut d'abord se ramener à une forme de référence qui va être appariée avec les autres individus, et qui nous servira pour les comparer. Le choix de cette forme est libre a priori, on peut décider de prendre tel ou tel individu comme référence, ou bien, de manière plus logique, de prendre une sorte de moyenne sur ces individus.

La première moyenne à laquelle on peut naturellement penser est bien évidemment la moyenne arithmétique. Néanmoins, d'autres moyennes existent et peuvent se révéler plus pertinentes. Examinons le cas de la moyenne de Karcher, qui est en fait, tout aussi - sinon plus - naturelle.

Si l'on note d une distance sur une espace H donné entre deux éléments x et y de cet espace, alors, pour une famille $(x^k)_k$ de H , on peut définir la fonction :

$$\begin{aligned} f : H &\rightarrow \overline{\mathbb{R}} \\ y &\mapsto \sum_k d(y, x^k)^2 \end{aligned}$$

La moyenne de Karcher est alors le point tel que :

$$\bar{x} = \operatorname{argmin}_y f(y) = \operatorname{argmin}_y \sum_k d(y, x^k)^2$$

Dans notre cas, l'écriture est la suivante : soient 2 formes y et z constituées de N points indicés, la distance géodésique entre ces deux formes est la norme du champ de vitesse d'énergie minimale, i.e.

$$d(y, z)^2 = \min_v \left(\int_0^1 \|v_x(t)\|^2 dt \right) \text{ avec } x(0) = y \text{ et } x(1) = z$$

Or minimiser cette transformation revient à trouver les α optimums puisque $v_x(t) = \sum_{i=1}^N K(x_i(t), y)\alpha_i(t)$, donc en se rappelant que $x(0) = y$ et $x(1) = z$:

$$\begin{aligned} d(y, z)^2 &= \min_{\alpha} \left(\int_0^1 \alpha(t)^t S_V(x(t)) \alpha(t) dt \right) \\ &= \min_{\alpha} \left(\int_0^1 \alpha^t S_V(x) \alpha dt \right) \end{aligned}$$

Dans les cas réels, le terme d'énergie à minimiser est différent de la distance géodésique pure, il convient donc de rajouter le terme d'attache aux données, afin d'éliminer le bruit notamment. Notons la somme des deux termes d'énergie J :

$$\begin{aligned} \min_{\alpha} J(\alpha, y, z) &= \min_{\alpha} \left(\int_0^1 \alpha^t S_{V_{\lambda}}(x) \alpha dt + \mu \sum_{i=1}^N |z_i - x_i(1)|^2 \right) \\ &= \min_{\alpha} (d(y, x(1))^2 + d_0(x(1), z)^2) \end{aligned}$$

Pour terminer, il suffit donc d'exhiber \bar{x} pour une famille $(x^k)_k$ donnée, en généralisant la notion de moyenne de Karcher pour la fonctionnelle J :

$$\begin{aligned} \min_{x_0, (\alpha^k)_k} \sum_k J(\alpha^k, x_0, x^k) &= \sum_k J(\alpha_*^k, \bar{x}, x^k) \\ &= J_0((\alpha_*^k)_k, \bar{x}, (x_k)_k) \end{aligned}$$

Nous décidons maintenant de chercher la moyenne de Karcher pour des transformations multi-échelles, où la pseudo-distance géodésique d'énergie minimale est cette fois-ci donnée par :

$$\begin{aligned} \min_v \left(\int_0^1 \|v_x(t)\|_{\lambda(t)}^2 dt \right) &= \min_v \left(\sum_{k=1}^m \int_{t_{k-1}}^{t_k} \|v_x(t)\|_{\lambda_k}^2 dt \right) \\ &= \min_\alpha \left(\sum_{k=1}^m \int_{t_{k-1}}^{t_k} \alpha^t S_{V_{\lambda_k}}(x) \alpha dt \right) \end{aligned}$$

Cette expression est encore une fois insuffisante dans les conditions réels, à cause du bruit notamment, un terme d'attache aux données est ajouté et l'énergie minimale devient :

$$\min_\alpha J(\alpha, y, z) = \min_\alpha \left(\sum_{k=1}^m \int_{t_{k-1}}^{t_k} \alpha^t S_{V_{\lambda_k}}(x) \alpha dt + \mu \sum_{i=1}^N |z_i - x_i(1)|^2 \right)$$

On généralise la notion de moyenne de Karcher pour la fonctionnelle J , on a alors :

$$\begin{aligned} \min_{x_0, (\alpha^k)_k} \sum_k J(\alpha^k, x_0, x^k) &= \sum_k J(\alpha_*^k, \bar{x}, x^k) \\ &= J_0((\alpha_*^k)_k, \bar{x}, (x_k)_k) \end{aligned}$$

Tout repose ici sur le fait que l'on puisse généraliser la moyenne de Karcher à d'autres choses que des distances, notamment des pseudo-distances. Ici, nos "distances" ne sont pas symétriques, en effet, pour deux individus y, z , et plusieurs échelles, $\min_\alpha J(\alpha, y, z) \neq \min_\alpha J(\alpha, z, y)$.

La méthode de résolution utilisée est similaire à celle utilisée pour l'optimisation des trajectoires, à savoir une descente de gradient, qui sera faite cette fois-ci sur le terme total d'énergie à la fois par rapport aux α et à x_0 . Ainsi, le gradient sur la coordonnée α^k ne change pas et reste toujours $\alpha^k - p^k$, le gradient sur x_0 est quant à lui issue de la somme des valeurs initiales des $(p^k)_k$, c'est-à-dire $\sum_k p^k(0)$.

En effet, sous certaines conditions de régularités sur les fonctions C et f , on peut différentier l'expression $\int_0^1 C(x, \alpha) dt + g(x(1))$ avec $\dot{x} = f(x, \alpha)$, on obtient :

$$\begin{aligned} \delta J &= \int_0^1 \left(\left(\frac{\partial C}{\partial x} \right) \delta x + \left(\frac{\partial C}{\partial \alpha} \right) \delta \alpha \right) dt + (Dg(x(1)) | \delta x(1)) \\ &= \int_0^1 \left(\left(\frac{\partial C}{\partial x} \right) \delta x + \left(\frac{\partial C}{\partial \alpha} \right) \delta \alpha \right) dt + (Dg(x(1)) | \delta x(1)) \\ &= \int_0^1 \left(\left(\frac{\partial C}{\partial x} \right) \delta x + \left(\frac{\partial C}{\partial \alpha} \right) \delta \alpha \right) dt - (p(1) | \delta x(1)) \\ &= \int_0^1 \left(\left(\frac{\partial C}{\partial x} - \frac{\partial f^*}{\partial x} p - \dot{p} \right) \delta x + \left(\frac{\partial C}{\partial \alpha} - \frac{\partial f^*}{\partial \alpha} \right) \delta \alpha \right) dt - (p(0) | \delta x(0)) \\ &= \int_0^1 \left(-\frac{\partial H}{\partial x} - \dot{p} \right) \delta x + \left(-\frac{\partial H}{\partial \alpha} \right) \delta \alpha dt - (p(0) | \delta x(0)) \\ &= \int_0^1 \left(-\frac{\partial H}{\partial \alpha} \right) \delta \alpha dt - (p(0) | \delta x(0)) \end{aligned}$$

Nous sommes évidemment dans un cas où les fonctions sont assez régulières pour faire tous ces calculs. Il faut en effet que les dérivées partielles de C et f satisfassent certaines conditions pour assurer l'existence et l'unicité des solutions. C'est pourquoi notamment on a l'égalité :

$$\delta x(t) = \delta x(0) + \int_0^t \left(\left(\frac{\partial f}{\partial x} \right) \delta x + \left(\frac{\partial f}{\partial \alpha} \right) \delta \alpha \right)$$

Nous avons fait beaucoup d'essais sur la moyenne de Karcher, et il se trouve qu'elle était en règle générale assez proche de la moyenne arithmétique. Cependant il y a de nombreux cas dans lesquels ces deux moyennes sont franchement différentes. Examinons ce cas : il y a un rectangle de référence en rouge qui a subi deux rotations d'angles $\frac{\pi}{2}$ et $-\frac{\pi}{2}$ qui sont respectivement les rectangles en vert et en bleu. La moyenne de Karcher apparait en violet, alors que la moyenne arithmétique apparait en noir.

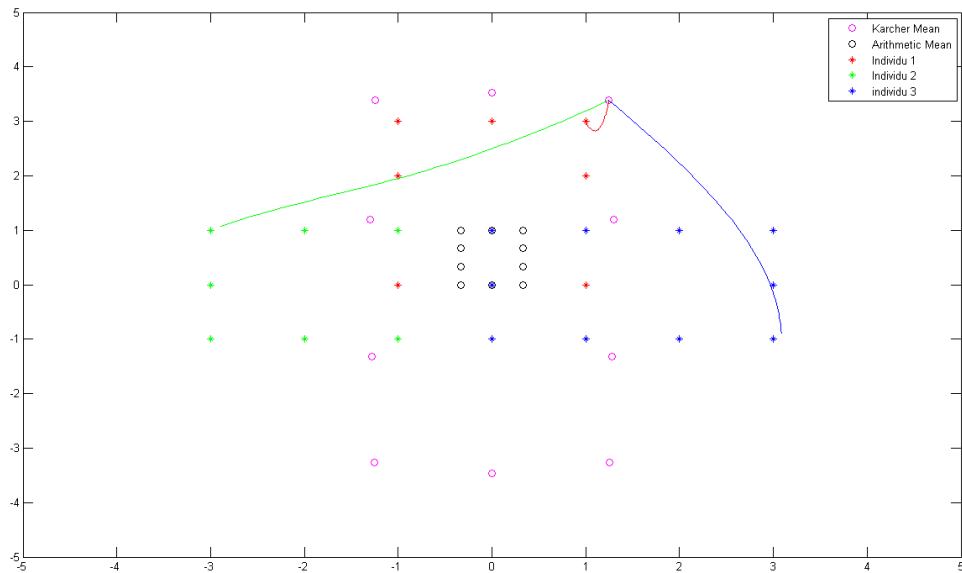


FIGURE 5 – Exemple où la moyenne de Karcher diffère de la moyenne arithmétique pour les paramètres $\lambda = 4$ et $\mu = 100$

4 Application : étude d'une maladie génétique chez les souris.

4.1 Cadre de l'étude.

Nous avons soumis les théories exposées précédemment et nos algorithmes à l'étude d'une maladie génétique. Celle-ci correspond à une mutation du gène *msx2*.

Elle existe, entre autres, chez l'homme et la souris. Chez l'homme, elle se traduit par un sous-développement de la boîte crânienne, ce qui nécessite une opération du crâne dès le plus jeune âge afin que leur cerveau ait la place pour se développer. Comme souvent en biologie, on s'intéresse à des modèles animaux, ici la souris avec mutation du gène *msx2*.

Rappelons que dans les cellules, chaque gène existe en deux exemplaires. En notant + l'allèle (version du gène) non-mutant et - l'allèle mutant, on peut distinguer trois catégories d'individus : les (homozygotes) sains ++, les hétérozygotes +-, et les (homozygotes) mutants --. Les mutants se caractérisent par des modifications du pelage, du comportement, des formes de la mâchoire et du crâne. En revanche, les hétérozygotes sont indistinguables à l'oeil nu de la souche sauvage, seule une analyse poussée permet de les distinguer d'une souris normale (dissimilarités micrométriques au niveau des dents,...).

Notre objectif était d'appliquer notre approche multi-échelles aux souris et de réaliser une analyse statistique des résultats afin de voir quelles échelles permettent de séparer les différentes classes : les échelles discriminantes indiquent l'ordre de grandeur de la taille des modifications dues à la modifications dues à la maladie. Nous souhaitons au minimum être capable de reconnaître les mutants, voire séparer les hétérozygotes des homozygotes sains.

Nos données nous ont été fournies par Yohann Simon, dentiste et chercheur au laboratoire de physio-pathologie orale de l'INSERM, Paris 5. Il a pris des radiographies des crânes de 44 souris (12 mutantes, 16 hétérozygotes, 16 homozygotes), et repéré sur chacune les coordonnées de 49 points particuliers (sommet du crâne, bas d'une dent,...). Nous avons employé la théorie des landmarks sur ces coordonnées et avons calculé la moyenne de Karcher, les moments et les trajectoires pour 6 échelles : 0.25 mm, 0.5 mm, 1 mm, 2 mm, 4 mm et 8 mm.

4.2 Analyse en composantes principales (ACP).

Nous nous proposons de faire une étude statistique des moments aux instants de changement d'échelle $(\alpha_n)_{1 \leq n \leq N}$, l'idée étant que la trajectoire minimisant le coût $\int_0^1 J(v_{x_i}(t))dt$ et amenant l'individu moyen au i -ème individu est une géodésique entièrement déterminée par la donnée de ces moment et la position initiale.

Nous avons tout d'abord réalisé une Analyse en Composantes Principales (ACP). Il s'agit d'une méthode permettant de visualiser des données de \mathbb{R}^N , $N \geq 3$. Elle consiste à chercher le plan de projection "déformant" le moins possible les données, i.e. minimisant la perte de variance

Il existe deux approches possibles : la première utilise la matrice de covariance, la deuxième la matrice de Gram des données. Nous avons choisi cette deuxième méthode car elle permet d'introduire un produit scalaire adapté, dans l'esprit des méthodes à noyaux.

Nous avons traité en parallèle et indépendamment les différentes échelles. Pour chacune des échelles, nos données X_i appartiennent à \mathbb{R}^{196} : elles se composent des positions $q_{i,p}$ et des moments $\alpha_{i,p}$ des 49 points de chaque souris à l'instant du changement d'échelle.

On considère la matrice de Gram suivante, définie par $G_{i,j} = \sum_{p=1}^N \sum_{q=1}^N \alpha_{i,p}^t K_V(x_i, x_j) \alpha_{j,q}$ où $\alpha(0)_{i,p}$ désigne le p -ème moment (i.e. associé au p -ème landmark) de l'individu i . Puisque G est symétrique, elle est diagonalisable en base orthonormée. Notons $(\gamma_k)_{1 \leq k \leq N}$ les vecteurs propres associés aux valeurs propres $(\lambda_k)_{1 \leq k \leq N}$ de G classées dans l'ordre décroissant.

Si $\lambda_k \neq 0$, on pose $U_k = \sum_{i=1}^N \frac{\gamma_{k,i} X_i}{\sqrt{\lambda_k}}$ où $\gamma_{k,i}$ désigne la i -ème coordonnée de γ_k .

Proposition Les U_k forment une base orthonormée de $E = \text{Vect}(\{X_i\})$.

En effet,

$$\begin{aligned} \langle U_k, U_l \rangle &= \frac{1}{\sqrt{\lambda_k \lambda_l}} \sum_{i,j} \gamma_k^i \gamma_l^j G_{i,j} \\ &= \frac{1}{\sqrt{\lambda_k \lambda_l}} G(\gamma_k, \gamma_l) \\ &= \frac{\lambda_k}{\sqrt{\lambda_k \lambda_l}} (\gamma_k, \gamma_l) \\ &= \delta_{k,l} \end{aligned}$$

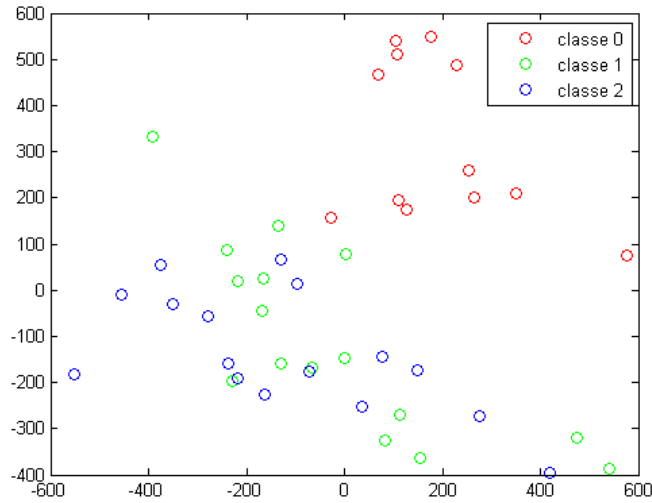
On a alors :

$$\begin{aligned} \langle U_k, X_i \rangle &= \frac{(G\gamma_k)_i}{\sqrt{\lambda_k}} \\ &= \frac{\lambda_k \gamma_k^i}{\sqrt{\lambda_k}} \\ &= \sqrt{\lambda_k} \gamma_k^i \end{aligned}$$

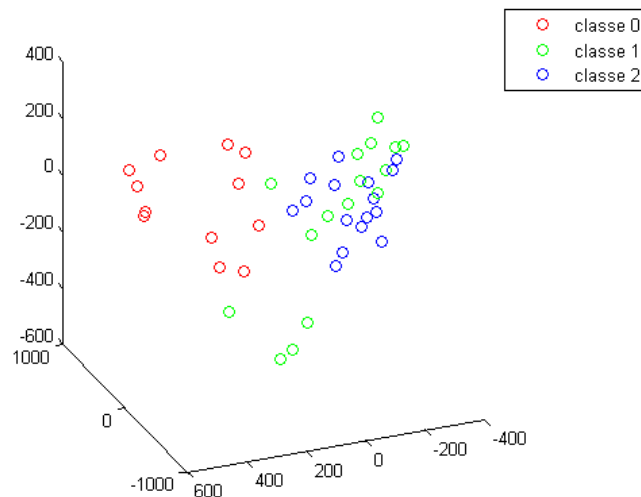
On projette alors chaque X_i sur le plan engendré par (U_1, U_2) .

La projection de nouvelles données X s'obtient par : $\langle U_k, X \rangle = \sum_{i=1}^N \frac{\gamma_{k,i} \langle X_i, X \rangle}{\sqrt{\lambda_k}}$.

Résultats de l'ACP (Première échelle : 0.25 mm, $\mu = 10$)



Cette méthode permet de séparer très nettement les mutants (rouges) des individus sains (bleus) et hétérozygotes (verts), mais ne permet pas de distinguer ces deux dernières classes qui restent très mélangées. Quels que soient le jeu de donnée et l'échelle utilisée, on obtient des figures similaires : mutants bien séparés, hétérozygotes et individus sains indistinguables. Ces résultats sont peu satisfaisants, ceci est dû au fait que la perte de variance est élevée (de l'ordre de 60% ici). On peut alors se demander si une ACP à trois dimensions serait plus concluante. La réponse est malheureusement négative, en considérant un troisième axe principal, la perte de variance reste dans notre cas supérieure à 50%. Visualisons néanmoins la projection des données sur les trois axes principaux sur la figure suivante :



Bien que l'on observe une légère amélioration des résultats, l'ACP n'en reste pas moins insuffisante. Nous avons donc utilisé une méthode plus élaborée : l'Analyse Discriminante de Fisher.

4.3 Analyse Discriminante de Fisher.

Cette méthode consiste à trouver un plongement isométrique de $E = \text{Vect}(\{X_i\})$ dans \mathbb{R}^N muni de sa structure euclidienne afin de pouvoir appliquer l'Analyse Discriminante Linéaire exposée plus bas.

Pour cela, on reprend la matrice de Gram G calculée précédemment. En notant Q une racine carrée de G , et $(\eta_i) = Qe_i$ sa i -ème colonne, $j : X_i \rightarrow \eta_i$ est un plongement isométrique : on peut montrer qu'elle est injective, et :

$$\begin{aligned} (\eta_i, \eta_j)_{L^2} &= (G^{1/2}e_i, G^{1/2}e_j) \\ &= (Ge_i, e_j) \\ &= G_{i,j} \\ &= \langle X_i, X_j \rangle \end{aligned}$$

Si on obtient de nouvelles données, on applique j à leur projection sur E . Cela revient à calculer $G^{-1/2}\gamma_x$, avec $\gamma_x = (\langle X_i, X \rangle)_i$.

On peut maintenant appliquer l'Analyse Discriminante Linéaire (LDA en anglais).

La LDA vise à trouver une projection qui maximise la variance inter-classes tout en minimisant la variance intra-classe.

Soient n données X_i , et y_1, y_2, \dots leur classe : 0 (mutants), 1 (hétérozygotes) ou 2 (*wild types*). On définit m_{y_i} la moyenne de la classe i et m la moyenne totale.

On définit $S_W = \frac{1}{n} \sum_{i=1}^n (X_i - m_{y_i})(X_i - m_{y_i})^T$ la matrice de covariance intra-classes, et

$S_B = \frac{1}{n} \sum_{i=1}^n (m_{y_i} - m)(m_{y_i} - m)^T$ la matrice de covariance inter-classes.

On cherche à maximiser le quotient de Rayleigh généralisé : $\max_w \frac{(S_B w, w)}{(S_W w, w)}$.

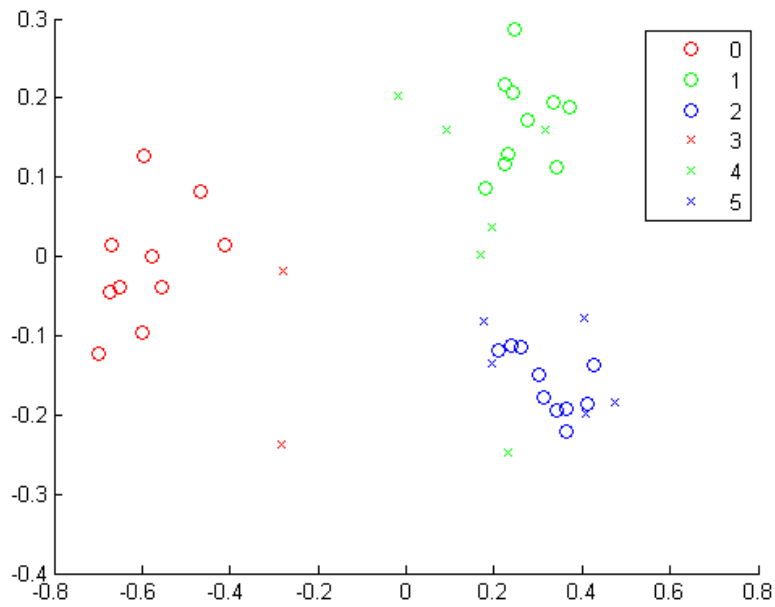
En posant $v = S_W^{-1/2} w$, cela revient à calculer : $\max_v \frac{(S_W^{-1/2} S_B S_W^{-1/2} v, v)}{v, v}$, ce qui équivaut à résoudre le problème aux valeurs propres généralisées : $S_B v = \lambda S_W v$.

On note w_1 et w_2 les vecteurs propres normés associés aux deux plus grandes valeurs propres. Pour projeter un individu (donnée initiale ou supplémentaire) sur le plan engendré par w_1 et w_2 , il suffit d'utiliser le produit scalaire euclidien.

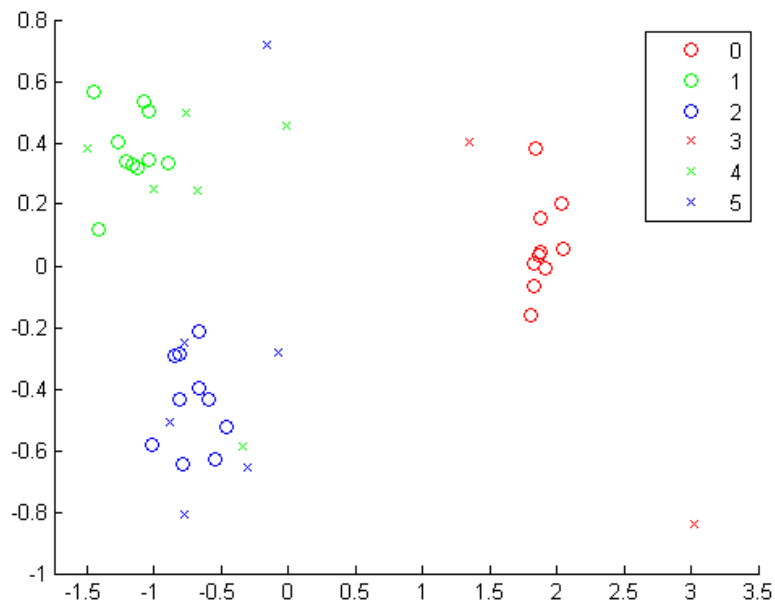
Pour classer un individu, il suffit alors de comparer les distances de sa projection avec les projections des centres de classes.

Les calculs précédents nécessitent l'inversibilité de S_W . Si celle-ci n'est pas inversible, on la "régularise", i.e. on lui ajoute $\sigma^2 Id$. Le paramètre σ doit être choisi suffisamment grand pour éviter le surapprentissage, mais pas trop afin que les groupes de données restent bien séparés.

4.3.1 Résultats de la LDA



(c) $\lambda = 0.25mm$; $\mu = 10$



(d) $\lambda = 8mm$; $\mu = 10$

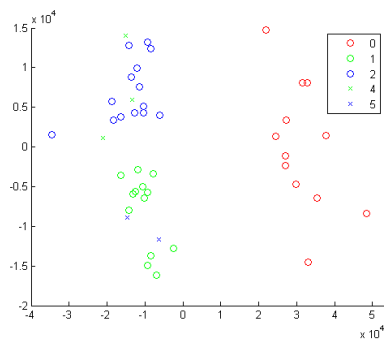
On a représenté en rond les données d'apprentissage et en croix les données de test. Les résultats sont très encourageants. Les trois groupes sont assez bien séparés (rappelons que cette séparation n'est pas possible par une observation à l'œil nu des souris!), et ce, quelle que soit l'échelle, ce qui suggère que la mutation agit à toutes les échelles.

Nous avons réalisé deux étapes supplémentaires.

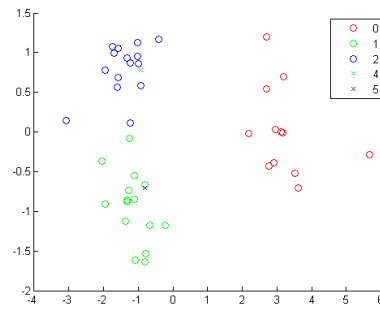
La première consistait à tester l'efficacité du classement pour différentes valeurs de régularisation afin de pouvoir optimiser le paramètre σ .

Pour cela, nous avons utilisé la méthode du "leave-one-out" (exclusion d'un point), qui consiste à utiliser toutes les données que l'on possède sauf une pour faire l'apprentissage, et utiliser la dernière comme test, pour voir si elle est correctement classée. En excluant tour-à-tour chacune des données, on récupère le nombre total d'erreurs, ce qui donne une idée de l'efficacité de l'algorithme. Nous avons optimisé séparément les σ pour chaque échelle et avons alors représenté les différents points de test, avec des ronds s'ils étaient correctement classés, et des croix s'ils étaient mal classés.

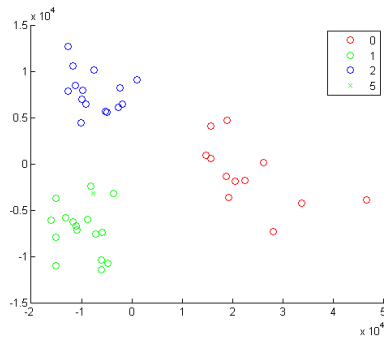
4.3.2 Résultats du leave-one-out



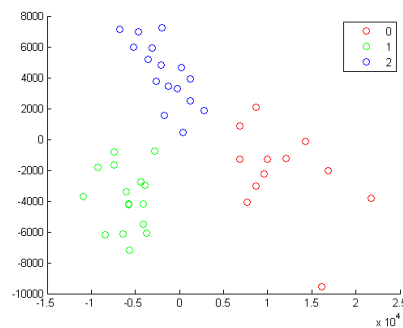
(e) $\lambda = 0.25mm$; $\mu = 10$



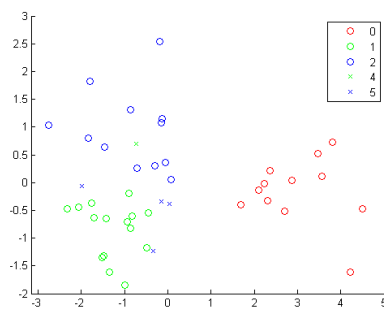
(f) $\lambda = 0.5mm$; $\mu = 10$



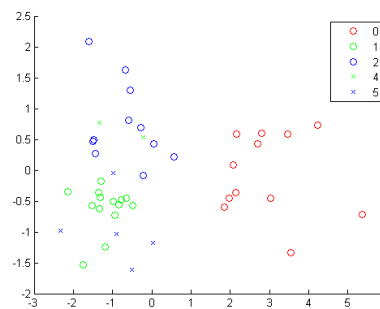
(g) $\lambda = 1.0mm$; $\mu = 10$



(h) $\lambda = 2.0mm$; $\mu = 10$



(i) $\lambda = 4.0mm$; $\mu = 10$



(j) $\lambda = 8.0mm$; $\mu = 10$

La deuxième étape a consisté à vérifier la crédibilité de notre algorithme, en lui faisant subir le test de permutation. Pour cela, on relance les algorithmes statistiques utilisés, mais en permutant les classes des individus. Si l'algorithme est bon, il est censé donner des résultats de classement moins bons (i.e. faire plus d'erreurs dans le "leave-one-out") pour les données permutées que pour les données originales. On réalise un (grand) nombre de permutations, la proportion de permutations qui donne un meilleur résultat que les données réelles s'appelle la p -value. Plus elle est proche de 0, plus l'algorithme est crédible.

Pour des raisons de temps de calculs, nous n'avons pu lancer qu'une dizaine de permutations, mais aucune n'a permis de faire aussi bien que les données originales, ce qui est un signe encourageant : on obtenait à chaque fois aux alentours de 66% d'erreur (plutôt un peu moins : l'algorithme optimisant certains paramètres, il est logique qu'il fasse mieux que le simple hasard).

Nombre d'erreurs des permutations

Permutation	non	1	2	3	4	5	6	7	8	9	10
$\lambda=0.25$ mm	5	28	24	26	28	25	25	31	24	27	22
$\lambda=0.5$ mm	2	28	23	30	27	30	25	29	26	25	26
$\lambda=1$ mm	1	29	33	29	24	29	29	25	23	29	27
$\lambda=2$ mm	0	29	35	28	28	28	31	27	26	27	27
$\lambda=4$ mm	5	22	28	29	23	27	25	27	28	27	24
$\lambda=8$ mm	7	21	27	20	22	28	25	30	25	31	21

Le fait que notre algorithme donne des résultats significatifs quelle que soit l'échelle indique que la mutation agit à l'échelle indiquée. D'autre part, les 3 échelles intermédiaires réalisent un score meilleur que les tests réalisés précédemment, qui utilisaient un algorithme d'appariement linéaire plus simples, et ne parvenaient pas à obtenir moins de 3 erreurs, ce qui montre la supériorité de la méthode non linéaire sur la méthode linéaire, même dans le cas de petites déformations étudiées ici.

5 Conclusion

L'analyse des données a été effectuée de façon satisfaisante avec six paramètres d'échelles. On peut se demander si le nombre d'échelles était adapté, et s'il n'aurait pas été plus efficace d'en garder simplement 3 (une petite, une moyenne, une grande). D'autre part, nous avons utilisé les échelles dans un ordre croissant, il serait intéressant de relancer le programme avec des échelles décroissantes afin de comparer les résultats. De même, on peut s'interroger sur le nombre de points qu'il est nécessaire de définir pour que l'on puisse décrire la forme considérée de façon satisfaisante. Bien évidemment, ce nombre dépend de la pertinence des points choisis. Il est par exemple inutile de relever plus de deux points pour décrire un segment. Par ailleurs, on peut se demander ce qu'on aurait obtenu en utilisant des noyaux non gaussiens (noyaux de Cauchy par exemple), voire en appliquant une théorie différente à nos données.

Ce stage nous a permis d'avoir une approche assez globale de la recherche en mathématiques, en nous faisant découvrir trois aspects bien différents : découverte et assimilation d'une théorie, réalisation de programmes pour mettre en oeuvre nos connaissances nouvelles, et application sur un cas pratique, suivie d'une analyse de nos résultats. Il nous a mis aux prises avec certains aspects de la recherche : nécessité de se répartir efficacement le travail ou encore importance d'optimiser les codes afin de réduire les temps de calcul.

6 Bibliographie

- Trouvé A, Younes L, (2011) Shape Spaces, in *Handbook of Mathematical Methods in Imaging* : 1311-1325 & 1350-1360
- Glaunès J, (2005) *Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie numérique*
- Vert J-P (2010) *Kernel Methods* (cours de Master de l'école des Mines)
- Tibshirani R, Hastie T, Friedman J. (2009) *The elements of Statistical Learning*
- Baudat G, Anouar F, *Generalized Discriminant Analysis Using a Kernel Approach*

7 Annexes

Quelques programmes Matlab réalisés

Appariement linéaire exact

```
function v = exact_matching(X1,X2,lambda)

%Match exactly X1 on X2
%X1 are the starting landmarks and X2 are the finishing ones.
%Syntax for landmarks of n point is X1=[x1 x2 x3 ... xn;
%                                     y1 y2 y3 ... yn];

%Length of the shape :
[P N] = size(X1) ;

%Displacement vector :
C1 = X2 - X1 ;
C = C1(:);

%Gaussian kernel :
Sv=exp(-(( repmat(X1(1,:),N,1)-repmat(X1(1,:)',1,N)).^2+( repmat(X1(2,:),N,1)-repmat(X1(2,:)')
Sv=kron(Sv,eye(P));

alpha = Sv\C;
alpha = reshape(alpha,P,N);

%Interpolating function :
v=@(y) alpha*exp(-(( repmat(y(1,:),N,1)-repmat(X1(1,:)',1,size(y,2))).^2+( repmat(y(2,:),N,1)-
```

Appariement non linéaire multi-échelles

```
function [a1,p1,q1,E]=multiRK3_shooting(X1,X2,lambda,mu,T,pas,delta,iter,fig,W,aa)

%Non-linear shooting method with a list of
%lambda=[lambda_1 , lambda_2.....,labmda_n]
%over T=[0 t_1 ... t_(n-1) 1]
%pas : step for time , delta : step for gradient descent
%iter : number of iterations
%fig : 0 hides figures, 1 draws matching iterations
%aa : initialization of trajectories from initial momenta aa

%Size and Time :
[P,N]=size(X1);
Tref=0:pas:1;
K=floor(T./pas)+1;

%Gaussian kernels :
Sv=@(x,l) exp(-(( repmat(x(1,:),N,1)-repmat(x(1,:)',1,N)).^2+( repmat(x(2,:),N,1)-repmat(x(2,
Sv=@(x,l) kron(Sv(reshape(x,P,N),l),eye(P)));

%Hamiltonian operator :
g=@(x) mu*norm(x-X2(:))^2;
```

```

dg=@(x) 2*mu*(x-X2(:));

%dqH :
dqH=@(q,p,u,l) [];
for i=1:N
    L=@(q,p,u,l) (q-repmat(q(:,i),1,N))*(exp(-((q(1,:)'-repmat(q(1,i),N,1)).^2+(q(2,:)'-repmat(q(1,i),N,1)).^2+mu*(q(1,i)-X2(i))^2)).^2+mu*(q(1,i)-X2(i))^2);
    dqH=@(q,p,u,l) [dqH(q,p,u,l); L(q,p,u,l)./1^2];
end
dqH=@(q,p,u,l) dqH(reshape(q,P,N),reshape(p,P,N),reshape(u,P,N),l);

%Initialization :
if ( nargin<11)
    a1=zeros(length(Tref),N*P);
    q1=repmat(X1(:)',length(Tref),1);
    p1=repmat(-2*mu*(X1(:)-X2(:))',length(Tref),1);
else
    a1=aa;
    a=interp1(Tref,a1,'linear','pp');

    f=@(t,x,l) Sv(x,l)*ppval(a,t);
    [Tref q1]=multiRK3(f,T,lambda,X1(:),pas);
    q=interp1(Tref,q1,'linear','pp');

    h=@(t,p,l) dqH(ppval(q,1-t),p,ppval(a,1-t),l);
    [Tref p1] = multiRK3(h,T,lambda,-dg(q1(end,:))',pas);
    p1=p1(length(Tref):-1:1,:);
end

%Energy
E=NRJ(a1,q1,Sv,g,T,lambda,pas);
test=1;
r=0.0001;
i=0;
imin=iter(1);
if length(iter)>1
    imax=iter(2);
else
    imax=10000;
end

nb_div=0;
%Iteration :
while test &&(i<imax)
    i=i+1;
    a0=a1;
    p0=p1;
    a1=a1-delta*(a1-p1);
    a=interp1(Tref,a1,'linear','pp');

    f=@(t,x,l) Sv(x,l)*ppval(a,t);

```



```

[Tref q1]=multiRK3(f,T,lambda,X1(:),pas);
q=interp1(Tref,q1,'linear','pp');

h=@(t,p,l) dqH(ppval(q,1-t),p,ppval(a,1-t),l);
[Tref p1] = multiRK3(h,T,lambda,-dg(q1(end,:))',pas);
p1=p1(length(Tref):-1:1,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Energy :
E_=E;
E=NRJ(a1,q1,Sv,g,T,lambda,pas);

test=(abs(E_-E)>r*abs(E)) || (i<=imin);

if E>E_
    nb_div=nb_div+1;
    delta=delta/2;
    E=E_;
    a1=a0;
    p1=p0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if fig>0
    figure(fig)
    clf
    plot(X1(1,:),X1(2:,:),'o',X2(1,:),X2(2:),'*')
    axis(W)
    legend('départ','arrivée')
    hold on
    for j=1:length(lambda)
        plot(q1(K(j):K(j+1),1:2:N*P),q1(K(j):K(j+1),2:2:N*P),'Color',[mod(j,2) 0 0])
        drawnow
    end
    hold off
end

end
end
fprintf(' i=%3.1d    nb_div=%3.1d    E=%8.3f \n',i,nb_div,E)

RK3 multi-échelles
function [out_t,out_y]=multiRK3(f,T,lambda,y0,h)

y=y0;
t=T(1);
out_y=y;
out_t=t;

for i=2:length(T)
    l=lambda(i-1);
    while t<=(T(i)-h+0.0001)

```

```

        p1=f(t,y,1);
        p2=f(t+h/2,y+h*p1/2,1);
        p3=f(t+3*h/4,y+3*h*p2/4,1);
        y=y+h*(2*p1+3*p2+4*p3)./9;
        t=t+h;
        out_y = [out_y , y];
        out_t = [out_t;t];
    end
end

out_y=out_y';

Moyenne de Karcher
function [aa,qq,KM]=Karcher_Mean(Da,lambda,mu,T,pas,delta,iter,deltaKM,iterKM,fig,F,X1)

%Size :
P=2;
D=Da;
N=size(D,1)/2;
nb_ind=size(D,2);

%Barycentering :
if (length(fig)>1 && fig(2)>0)
    B=[sum(D(1:2:2*N,:));sum(D(2:2:2*N,:))]./N;
    D=D-repmat(B,N,1);
end

%Centering :
X=sum(D,2)./nb_ind;
moy=reshape(X,P,N);

%Initialization :
if (nargin > 11)
    KM=X1;
else
    KM=moy;
end
if (length(fig)>1 && fig(2)>0)
    KM=KM-repmat(sum(KM,2),1,N)./N;
end

%Initial trajectories :
aa={};
p0=0;
Et=0;
w=0;

for r=1:nb_ind
    X2=reshape(D(:,r),P,N);

```

```

    [aa1,p,q,E]=multiRK3_shooting(KM,X2,lambda,mu,T,pas,delta,iter,2*fig(1),F);
    aa{r}=aa1;
    p0=p0+p(1,:)' ;
    Et=Et+E;
end
fprintf(' w=%3.1d      Et=%8.3e      DeltaKM=%8.10e \n',w,Et,deltaKM)

%Iteration :
wmin=iterKM(1);
if length(iterKM)>1
    wmax=iterKM(2);
else
    wmax=10000;
end
w=0;
r_KM=0.0001;
testKM=1;

while testKM && (w<wmax)
    w=w+1;
    a=aa;
    KM_=KM;
    KM=KM+deltaKM*reshape(p0,P,N);
    p00=p0;
    p0=0;
    Et_=Et;
    Et=0;

    if fig(1)>0
        figure(fig(1))
        clf
        plot(KM(1,:),KM(2,:),'ok',KM_(1,:),KM_(2,:),'og')
        legend('fresh KM','old KM')
    end

    for r=1:nb_ind
        X2=reshape(D(:,r),P,N);
        [aa1,p,q,E]=multiRK3_shooting(KM,X2,lambda,mu,T,pas,delta,iter,2*fig(1),F,aa{r});
        aa{r}=aa1;
        p0=p0+p(1,:)' ;
        Et=Et+E;
    end

    testKM=(abs(Et_-Et)>r_KM*abs(Et)) || (w<=wmin);

    if Et>Et_
        deltaKM=deltaKM/2;
        Et=Et_;
        aa=a;
        KM=KM_;
    end
end

```

```

    p0=p00;
end

fprintf(' w=%3.1d    Et=%8.3e    DeltaKM=%8.10e \n',w,Et,deltaKM)

end

%Momenta and trajectories :
qq={};
Tref=0:pas:1;
Sv=@(x,l) exp(-(( repmat(x(1,:),N,1)-repmat(x(1,:) ',1,N)).^2+( repmat(x(2,:),N,1)-repmat(x(2,
Sv=@(x,l) kron(Sv(reshape(x,P,N),1),eye(P)));
for r=1:nb_ind
    a=interp1(Tref,aa{r}, 'linear', 'pp');
    f=@(t,x,l) Sv(x,l)*ppval(a,t);
    [Tref qq{r}]=multiRK3(f,T,lambda,KM(:),pas);
end

```