NNT : 2016SACLN069

Université Paris-Saclay

École Doctorale de Mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École Normale Supérieure de Paris-Saclay

Laboratoires d'accueil :
Centre de Mathématiques et de Leurs Applications, UMR 8536 CNRS
ARAMIS Lab, INRIA Paris, CNRS UMR 7225, INSERM U 1127, UPMC, Institut du Cerveau et de la Moelle épinière (ICM)

THÈSE DE DOCTORAT

Spécialité : Mathématiques appliquées

Barbara GRIS

Modular approach on shape spaces, sub-Riemannian geometry and computational anatomy

Date de soutenance : 5 Décembre 2016

Après avis des rapporteurs : Tom FLETCHER (University of Utah)
Simon MASNOU (Université Lyon 1)

Jury de soutenance :

Tom FLETCHER	(University of Utah)	Rapporteur
Simon MASNOU	(Université Lyon 1)	Rapporteur
Julie DELON	(Université Paris Descartes)	Examinatrice
Emmanuel TRÉLAT	(Université Pierre et Marie Curie)	Examinateur
Laurent YOUNES	(Johns Hopkins University)	Examinateur
Stanley DURRLEMAN	(INRIA)	Directeur de thèse
Alain TROUVÉ	(ENS de Cachan)	Directeur de thèse

Abstract

This thesis is dedicated to the development of a new deformation model to study shapes. Deformations, and diffeormophisms in particular, have played a tremendous role in the field of statistical shape analysis, as a proxy to measure and interpret differences between similar objects but with different shapes. Diffeomorphisms usually result from the integration of a flow of regular velocity fields, whose parameters have not enabled so far a full control of the local behaviour of the deformation. We propose a new model in which vector fields, and then diffeomorphisms, are built on the combination of a few local and interpretable vector fields. These vector fields are generated thanks to a structure which we named deformation module. Deformation modules generate vector fields of a particular type (for instance: a scaling) chosen in advance: they allow to incorporate a constraint in the deformation model. These constraints can correspond either to an additional knowledge one would have on the shapes under study, or to a point of view from which one would want to study these shapes. In a first chapter we introduce this new notion of deformation module and we give several examples to show how diverse they can be. We also explain how one can easily build complex deformation modules adapted to complex constraints by combining simple base deformation modules. Then we introduce the construction of modular large deformations as flows of vector fields generated by a deformation module. Vector fields generated by a deformation module are parametrized by two variables: a geometrical one named geometrical descriptor and a control one. We build these large deformations so that the geometrical descriptor follows the deformation of the ambient space. Then defining a modular large deformation corresponds to defining an initial geometrical descriptor and a trajectory of controls. We also associate a notion of cost for each couple of geometrical descriptor and control. In a second chapter we explain how we can use a given deformation module to study data shapes. We first build a sub-Riemannian structure on the space defined as the product of the data shape space and the space of geometrical descriptors. The sub-Riemannian metric comes from the chosen cost: we equip the shape space with a chosen metric, which is not in general the pull-back of a metric on vector fields but takes into account the way vector fields are built with the chosen constraints. Thanks to this structure we define a sub-Riemannian distance on this new space and we show the existence, under some mild assumptions, of geodesics (trajectories whose length equals the distance between the starting and ending points). The study of geodesics amounts to an optimal control problem, and they can be estimated thanks to an Hamiltonian framework: in particular we show that they can be parametrized by an initial variable named momentum. Thanks to these notions of distance, and geodesic, we can define optimal modular large deformations transporting a source shape into a target shape. We also define the modular atlas of a population of shapes which is made of a mean shape, and one modular large deformation per shape transforming the mean shape into these shape. In the discussion we study different alternative models where geodesics are parametrized in lower dimension. In a third chapter we present the algorithm that was implemented in order to compute these modular large deformations and the gradient descent to estimate the optimal ones as well as mean shapes. In a last chapter we introduce several numerical examples thanks to which we study different aspects of our model. In particular we show that the choice of the used deformation module influences the form of the estimated mean shape, and that by choosing an adapted deformation module we are able to perform in a satisfying and robust way simultaneously rigid and non linear registration. In the last example we study shapes without any prior knowledge, then we use a module corresponding to weak constraints and we show that the atlas computation still gives interesting results.

Résumé en français

Dans cette thèse nous développons un nouveau modèle de déformation pour étudier les formes. Les déformations, et les difféomorphismes en particulier, jouent un rôle fondamental dans l'étude statistique de formes, comme un moyen de mesurer et d'interpréter les différences entre des objets similaires. Les difféomorphismes résultent généralement d'une intégration d'un flot régulier de champs de vitesses, dont les paramètres n'ont jamais encore vraiment permis de contrôler localement les déformations. Nous proposons un nouveau modèle dans lequel les champs de vitesses sont construits grâce à la combinaison de quelques champs de vecteurs locaux et interprétables. Ces champs de vecteurs sont générés à l'aide d'une structure que nous appelons module de déformation. Un module de déformation génère un champs de vecteurs d'un type particulier (par exemple : homothétie) choisi à l'avance: cela permet d'incorporer des contraintes dans le modèle de déformation. Ces contraintes peuvent correspondre à un savoir que l'on a sur les formes étudiées, ou à un point de vue à partir duquel on veut étudier ces formes. Dans un premier chapitre nous définissons les modules de déformation et nous en donnons des exemples variés. Nous expliquons également comment construire facilement un module de déformation adapté à des contraintes complexes en combinant des modules de déformations simples. Ensuite nous construisons des grandes déformations modulaires en tant que flots de champs de vecteurs générés par un module de déformation. Les champs de vecteurs générés par un module de déformation sont paramétrés par deux variables : une géométrique (descripteur géométrique) et une de contrôle. Nous associons également un coût à chaque couple de descripteur géométrique et de contrôle. Dans un deuxième chapitre nous expliquons comment utiliser un module de déformation donné pour étudier des formes. Nous construisons tout d'abord une structure sous-Riemannienne sur l'espace défini comme le produit de l'espace de formes et de celui des descripteurs géométriques. La métrique sous-Riemannienne vient du coût choisi : nous munissons le nouvel espace d'une métrique choisie, qui en général n'est pas le pull-back d'une métrique sur les champs de vecteurs mais tient compte la manière dont les champs de vecteurs sont construits à partir des contraintes. Grâce à cette structure nous définissons une distance sous-Riemannienne et nous montrons l'existence des géodésiques (trajectoires dont la longueur vaut la distance entre les points de départ et d'arrivée). L'étude des géodésiques se ramène à un problème de contrôle optimal, elles peuvent être obtenues grâce à un formalisme Hamiltonien. En particulier nous montrons qu'elles peuvent être paramétrées par une variable initiale, le moment. Après cela nous présentons les grandes déformations modulaires optimales transportant une forme source sur une forme cible. Nous définissons également l'atlas modulaire d'une population de formes par la donnée d'une forme moyenne et d'une grande déformation modulaire par forme transformant la forme moyenne en cette forme. Dans la discussion nous étudions un modèle alternatif dans lequel les géodésiques sont paramétrées en dimension plus petite. Dans un troisième chapitre nous présentons l'algorithme implémenté pour obtenir les grandes déformations ainsi que la descente de gradient estimant les atlas. Dans un dernier chapitre nous présentons plusieurs exemples numériques grâce auxquels nous étudions certains aspects de notre modèle. En particulier nous montrons que le choix du module de déformation utilisé influence la forme moyenne, et que choisir un module de déformation adapté permet d'effectuer simultanément des recalages rigides et non linéaires. Dans le dernier exemple nous étudions des formes sans a priori, nous utilisons donc un module correspondant à des contraintes faibles et nous montrons que l'atlas obtenu est toujours intéressant.

Remerciements

Je voudrais tout d'abord remercier mes directeurs de thèse Stanley Durrleman et Alain Trouvé pour leur soutien, leur écoute et leur disponibilité. Ils m'ont proposé un sujet de recherche passionnant, qui a beaucoup évolué. Je leur suis sincèrement reconnaissante de m'avoir aiguillée tout au long de ma thèse, de m'avoir transmis leur passion des belles mathématiques associées à l'étude de problèmes réels. J'ai eu beaucoup de chance de découvrir la recherche à leurs côtés, et cela a été un grand plaisir !

Je voudrais également remercier les deux rapporteurs de ma thèse Tom Fletcher et Simon Masnou pour leurs lectures approfondies de mon manuscrit et leurs commentaires pertinents. Je remercie tout particulièrement Simon pour l'intérêt qu'il a porté à mon travail, ainsi que pour les discussions très intéressantes que nous avons eues. Je remercie également Julie Delon, Emmanuel Trélat et Laurent Younes d'avoir accepté de faire partie du jury, et notamment Laurent Younes pour être venu de Baltimore.

Les nombreuses discussions avec toutes les personnes de la communauté "Shape" ont participé à rendre ma thèse agréable, notamment au cours des rencontres annuelles et je les remercie chaleureusement. Je remercie tout particulièrement Benjamin, Irène, Nicolas, Pierre et Sylvain pour leur amitié et leur soutien.

Je remercie de manière générale les équipes d'Aramis et du CMLA grâce auxquelles j'ai eu la chance de travailler dans une ambiance agréable et pleine de vie ! Un grand merci à Alexandre B., Alexandre R., Alexis, Ana, Anne, Benjamin, Carlos, Catherine, Chabha, Claire, Fabrizio, Fanny, Federico, François, Igor, Jean-Baptiste, Jeremy, Jorge, Junhao, Hugo, Kelly, Ludovic, Manon, Martina, Michael, Matthieu, Marie-Constance, Mario, Maxime, Olivier, Pietro, Sabrina, Sonia, Takoua, Thomas, Tristan, Xavier pour les happy hour, les pauses foot, les papiros et les nombreuses discussions qui ont adouci les périodes difficiles de la fin de thèse. Merci également à Adrien, Ariane, Arthur, Carlo, Charles, Enric, Gabriele, Giacommo, Lara, Marie B., Marie de M., Miguel, Maxime, Rafael, Samy, Sandrine, Tristan, Yves pour la bonne ambiance du CMLA !

Merci à mes proches et amis rencontrés au fil des années pour leur présence et leur soutien sans lequel je n'aurais jamais fini cette thèse. Un merci spécial à mes amis de toujours Caro, Lily, Rebecca, Sarah et Vincent pour tous nos moments ensembles, passés et à venir !

Je voudrais également remercier ma belle-famille, notamment Chantal, Magali, Régis et Rémi pour leur gentillesse et leur accueil chaleureux. Les week-end à Reims et les Noëls ensemble sont toujours des moments que j'apprécie énormément !

Je réserve une pensée particulière à ma famille qui m'a toujours soutenue et encouragée. En particulier je remercie profondément mes parents et ma sœur Irina pour avoir toujours été là pour moi, et pour leur confiance inébranlable. C'est grâce à eux si j'en suis là aujourd'hui.

Enfin je remercie Loïc de partager ma vie, et de la rendre meilleure jour après jour.

Contents

Chapter 1

Introduction (français)

Contents

1.1 L'analyse de forme dans le contexte de l'imagerie médicale

L'interprétation automatique de données d'imagerie est un défi importants des domaines de l'imagerie médicale et de la vision artificielle. Ces données sont des représentations de vrais objets en dimension 2 ou 3: ce peut être des images (IRM par exemple), ou des données géométriques (telles que des courbes ou des surfaces obtenues par segmentation). Elles encodent des informations complexes, et même si parfois cette information peut être interprétée visuellement, en général il n'est pas possible pour l'œil humain d'appréhender ces formes dans leur totalité. Par exemple, étant donnée une IRM cérébrale, les plis les plus profonds du cortex peuvent être labellisés manuellement, mais identifier les structures les plus fines est une tâches difficiles. De même, étudier les ressemblances et différences entre trois ou quatre sujets peut parfois être fait manuellement, mais afin de comprendre les différences anatomiques au sein de plus grandes populations, des méthodes automatiques sont nécessaires. En pratique étudier de telles grandes populations est nécessaire par exemple pour détecter les caractéristiques anatomiques associées à une maladie particulière, ou pour regrouper les différences anatomiques parmi une population en différentes classes.
Le but de l'analyse de forme est précisément de développer des méthodes permettant de comprendre de telles données de forme de manière globale et automatique. Une approche efficace pour analyser automatiquement une forme est de développer des outils permettant de la *comparer* automatiquement à une population de formes similaires qui ont déjà été étudiées. Pour cela, il faut tout d'abord développer des méthodes permettant d'analyser une *population de formes* et ensuite être capable de *comparer* des formes. L'analyse d'une population de formes peut être faite via l'estimation d'une forme moyenne, appelée le template, qui reflète la structure sous-jacente de la population; et simultanément via la comparaison des formes de la population à ce template afin de comprendre la variabilité au sein de cette population. Ainsi le point clé ici est le développement de méthodes permettant de comparer des formes.
Une approche introduite par d'Arcy Thompson [T+42], et ayant prouvé son intérêt, est d'adopter un point de vue géométrique et d'étudier les déformations transformant la première forme en la seconde. Dans le cas particulier où les formes étudiées sont constituées de landmarks, une idée présentée par F. L. Bookstein dans [Boo97, B+89] est d'interpoler les déplacements de landmarks entre les deux formes par des splines, permettant d'obtenir un champ de vecteurs lisse et dense. Cette méthode permit de résoudre de nombreux problèmes, notamment dans le domaine de l'imagerie médicale. Néanmoins elle souffre d'une importante limite : il n'y a aucune garantie que le champs de vecteurs obtenu soit inversible. Ceci est problématique : un intérêt de ce cadre géométrique est de mettre en correspondance certaines régions des données et donc une déformation bijective est nécessaire. Dans sa *Pattern Theory* [GK93, GCK12, Gre96, Gre93], U. Grenander développa une nouvelle idée pour étudier les formes : il les considère sous l'action d'un groupe de déformations. Dans [MCAG93], une forme *idéale* (le *template*) est considérée et les auteurs construisent des déformations inversibles qui transforment cette forme idéale en les formes données. Ainsi une connaissance particulière sur cette

forme idéale (par exemple une segmentation) peut être automatiquement adaptée aux nouvelles données. Cependant, en pratique, les déformations qui sont utilisées sont des approximations linéaires de difféomorphismes (*petites déformations*), et donc ne permettent pas de capturer les déformations de grande ampleur entre le template et les formes données. Dans [CRM96] est présenté un cadre théorique permettant pour la première fois de construire de manière systématiques des grandes déformations difféomorphiques. Les déformations sont obtenues par intégration d'une trajectoire de champs de vecteurs. On considère ainsi les trajectoires φ de difféomorphismes commençant à $\varphi_{t=0} = Id$ et satisfaisant :

$$\dot{\varphi}_t = v_t \circ \varphi_t.$$

Dans ce cadre, l'ensemble des difféomorphismes qui est utilisé pour étudier un ensemble de formes n'est pas directement défini: il s'agit de d'abord définir les trajectoires de champs de vecteurs considérées, et ensuite de construire par intégration l'ensemble de grandes déformations correspondant. Cette construction a été largement utilisée en analyse de forme [AG04, JDJG04, KWB08, RFS03, Thi98, Tog98, VPPA09, WBR+07] car elle permet de construire des grandes déformations en ne spécifiant que les déplacement infinitésimaux. Différents choix de trajectoires de champs de vecteurs permettent de construire différents modèles de déformation. Dans une première grande classe de modèles présentée dans les Sections 1.2.1, 1.2.2 et 1.2.3, un espace de champs de vecteurs est fixé et on considère toutes les trajectoires intégrables de cet espace. Dans ces modèles il n'y a pas de structure imposée sr les déformations à l'avance :cela permet de construire des ensembles riches de difféomorphismes et des déformations adaptées pour apparier les formes, mais cela empêche de les interpréter naturellement. En effet, comme les champs de vecteurs ne sont pas paramétrés à l'avance avec des variables ayant du sens, ces modèles ne fournissent pas d'outils pour comprendre les déformations obtenues. Afin de surmonter ce problème, simultanément, ont été développés des modèles paramétriques, une présentation en est donnée Sections 1.2.5,1.2.6 and 1.2.7. Dans ces approches, à chaque instant de la trajectoire, le champs de vecteurs est contraint à être obtenu via la combinaison d'un petit nombre de générateurs de champs de vecteurs créant des champs de vecteurs particuliers. En choisissant certains générateurs, on peut construire des trajectoires de difféomorphismes correspondant à un certain point de vue (un *a priori*) que l'on veut avoir sur les formes. Cependant, à notre connaissance, aucun de ces cadres ne permet de construire des générateurs correspondant à un a priori complexe générique. De plus, aucun d'entre eux ne permet de munir ces champs de vecteurs d'une métrique dépendant de ces a priori. C'est ce que nous voulons développer : le but est la construction d'un vocabulaire qui serait utilisé pour décrire les différences entre les formes, et d'une métrique qui mesurerait ces différences en tenant compte les choix de vocabulaire. Ce chapitre est organisé comme suit : dans la Section 1.2 nous présentons les différents modèles de déformation qui ont été développés dans un but similaire au notre, et nous introduisons les raisons qui nous ont poussées à développer un modèle de déformation *modulaire*. Dans la Section 1.3 est présenté un résumé des contributions de cette thèse.

1.2 Les difféomorphismes en Anatomie Computationnelle

1.2.1 LDDMM

Les travaux fondateurs de U. Grenander [Gre93] et G.E. Christensen et al. [CRM96] définissent les notions de template et de grandes déformations difféomorphiques construites comme flot de trajectoires de champs de vecteurs. Dans [DGM98], les auteurs étudient les conditions théoriques nécessaires pour obtenir l'existence d'une trajectoire optimal de champs de vecteurs transportant une forme source aussi proche que possible d'une forme cible. Ils étudient également ce problème d'un point de vue Bayesien. Dans [Tro95, Tro98], A. Trouvé introduit un nouveau cadre générique afin d'étudier des motifs au sein d'une population de formes. Dans cet article, des formes particulières (*motifs*) sont considérés : ce sont des fonctions mesurables d'une variété Riemannienne compacte de dimension finie sans bord vers une variété X de dimension finie. Ceci correspond par exemple au cas de courbes fermées sur $\mathbb{R}^2$. Afin d'étudier ces formes, un espace de Banach de champs de vecteurs, noté B, est construit. Ensuite est défini le groupe A_B des valeurs finales des flots de trajectoires de champs de vecteurs appartenant à B. Ce modèle fournit un cadre mathématiques rigoureux afin de construire des grandes déformations difféomorphiques, ainsi qu'une métrique (et donc une distance) sur ce groupe de difféomorphisme. Afin d'étudier les différences entre deux formes f_0 et f_1, l'idée est de rechercher un difféomorphisme dans A_B (c'est-à-dire une trajectoire de champs de vecteurs dans B) aussi proche que possible de l'Identité et transportant f_0 aussi proche que possible de f_1. L'existence d'une telle trajectoire optimale est prouvée. De plus, cette distance permet de mesurer la ressemblance entre deux formes et est utilisée dans cet article pour associer une forme à une forme-motif. Des équations d'Euler-Lagrange pour les trajectoires optimales furent étudiées dans [BMTY05], menant à un cadre connu comme la Large Deformations Diffeomorphic Metric Mapping (LDDMM) [MYT14, YAM09, You10]. Même si la définition de forme qui est considérée dans ce cadre est satisfaite par de nombreux exemples, il n'est pas donné de définition générique.

1.2.2 Notion nécessaire: espace de formes

La formalisation de la notion de forme a été considérée de plusieurs manières [BBM14, Ken84, KBCL09]. Dans cette thèse nous considérerons l'approche développée par S. Arguillère [Arg14, Arg15a, Arg15b]. Dans [Arg14], un espace de formes $\mathcal{O}$ est défini comme une variété sur laquelle un groupe G de difféomorphismes agit continument (voir Section 3.2 pour une définition précise). Une forme est alors définie comme un élément de cet espace. Dans ce cadre, on fixe un Espace de Hilbert à Noyau Reproduisant (ENR, voir 3.1) de champs de vecteurs V et ensuite on construit le groupe G_V des valeurs finales prises par les flots de trajectoires de champs de vecteurs de V. Ce groupe G_V est un sous-groupe de G, il est muni d'une métrique grâce à V et sera utilisé pour étudier les formes de $\mathcal{O}$. De même que dans [Tro95], afin d'étudier les différences entre deux formes f_0 et f_1, on recherche un difféomorphisme dans G_V (c'est-à-dire une trajectoire de champs de vecteurs de V) aussi proche que possible de l'Identité et transportant f_0 aussi

proche que possible de f_1. Son existence est démontrée et il est montré que la trajectoire v_t de champs de vecteurs correspondante peut être paramétrée par une variable duale initiale appelée le *moment*. De plus à chaque instant t, v_t est *porté* par la forme dans un sens qui sera détaillé Section 3.2. À partir de la métrique sur G_V, on peut construire une métrique sous-Riemannienne sur $\mathcal{O}$ et ensuite mesurer les différences entre f_0 et f_1. Cette mesure dépend de l'espace V et une question importante est le choix de l'espace V adapté à un problème particulier. Différents choix d'espaces de formes et d'ENR V dans la littérature correspondent à ce cadre générique.

1.2.3 Divers espaces de formes et ENR: divers cadres

Dans [Gla05, GVM04] sont détaillés les cas des landmarks, mais également des mesures et des courants qui permettent de considérer des ensembles de points non labellisés , ainsi que des courbes et des surfaces non paramétrées. Les auteurs se restreignent au cas des ENR scalaires Gaussiens, c'est-à-dire où le noyau définissant l'ENR est scalaire Gaussien et est donc défini par un paramètre scalaire *l'échelle*, représentant l'échelle caractéristique des déformations. Comme le choix de l'échelle σ de l'ENR scalaire gaussien peut être un problème important, plusieurs approches multi-échelles ont été développées. Dans [RVW+11] le noyau de l'ENR est une somme de noyaux Gaussiens à différentes échelles. Ceci permet d'obtenir des trajectoires de champs de vecteurs capturant la variabilité à différentes échelle simultanément et qui semblent plus naturels dans le cas où il y a des différences à plusieurs échelles caractéristiques entre les données. Dans autre approche multi-échelle initiée dans [SLNP11], une famille W d'espaces de champs de vecteurs, et non plus un seul espace, est considérée. Chacun de ces espace est un ENR à une échelle différente. L'action d'une famille de champs de vecteurs appartenant à W sur une forme est construite comme la somme des actions de chacun des champs de vecteurs. Cependant, il est montré dans [BRV12] que cette approche est équivalente à la précédente [RVW+11], ce problème a été étudié dans [SLNP13]. Dans toutes ces approches, les formes considérées sont d'ordre zéro dans le sens qu'elles ne portent de l'information qu'à propos des localisations des points. Il est montré dans [Gla05] que pour les trajectoires optimales, à chaque instant le champs de vecteur est une somme de translations localisées par la fonction noyau. Dans [SNDP13], une autre catégorie de formes est considérée. Dans ce cas elle portent une information d'ordre supérieur: ce sont des images et on ne considère pas seulement les valeurs de l'image en chaque point mais aussi les dérivées de l'image en ces points. En conséquence, les trajectoires optimales de champs de vecteurs sont paramétrés par des moments d'ordres supérieurs et donc les champs de vecteurs des trajectoires optimales sont exprimées à l'aide des dérivées du noyau. Ceci permet d'obtenir des déformations locales plus complexes comme des applications localement affines. Une étude théorique de ce cadre a été menée dans [Jac13], en particulier sur les réductions dues aux symétries que l'on peut obtenir.

Dans toutes les approches précédemment citées, en choisissant un espace de champs de vecteurs adaptés on construit des grandes déformations adaptées à un problème particulier. Cependant, le moment (variable duale paramétrant les trajectoires optimales) est de même dimension que les formes considérées et cette dimension peut être très grande en pratique. Cette paramétrisation peut donc être redondante. Dans un modèle sparse

présenté dans [DPC+14], S. Durrleman considère les géodésiques générées par un petit nombre de landmarks de sorte qu'elles sont paramétrées en petite dimension. Ensuite on peut étudier comment ces géodésiques appartenant à un espace de petites dimension peuvent déformer des formes de grandes dimension. Dans ce cadre l'idée est de trouver la dimension réelle nécessaire des trajectoires qui doivent être considérées pour étudier la variabilité au sein d'une population. Cependant ces champs de vecteurs sont toujours des sommes de translations locales (car ce sont des géodésiques pour des landmarks) et donc l'interprétation des motifs non linéaires est limitée.

1.2.4 Des déformations plus structurées ?

Une limite générale de ces modèles non paramétriques est le fait que la nature des champs de vecteurs optimaux dépend de la nature des formes et non de la nature des différences que l'on veut étudier. Il n'est pas possible d'imposer un certain motif de déformation, correspondant à un savoir préalable. C'est pourtant nécessaire dans certains cas : par exemple pour des images biologiques, certains motifs de déformation peuvent être totalement non pertinents et il peut être intéressant de construire un modèle de déformation qui ne les autorise pas mais au contraire en favorise des plus réalistes. C'est une des raisons pour lesquelles furent développés des approches *paramétriques* dans lesquelles les déformations sont *structurées* a priori. Dans ces modèles les champs de vecteurs à chaque instant sont contraints à être la somme d'un petit nombre de champs de vecteurs paramétrés en petite dimension grâce à des variables de *contrôle*, de sorte que les motifs de déformation sont contrôlés et peuvent être interprétés à l'aide des valeurs des contrôles. L'idée est ici de réduire l'ensemble des difféomorphismes considérés afin d'améliorer leur compréhension et leur adaptation à un problème particulier.

1.2.5 Cadre poly-affine

Dans un cadre poly-affine présenté dans [APA05, ACAP09], le but est de construire des déformations obtenues par fusion de différentes transformations localement affines. Si les applications localement affines sont sommées, la déformation résultante est en général non inversible. C'est pourquoi dans ce cadre les déformations sont obtenues par intégration de trajectoires de champs de vecteurs. À une déformation affine T est associée une trajectoire de champs de vecteurs dont le flot vaut T à $t = 1$. Alors si on veut fusionner différents déformations affines T_i agissant dans différentes régions Ω_i définies par des fonctions poinds w_i, on définit les trajectoires de champs de vecteurs v_i correspondantes et on intègre la trajectoire de champs de vecteurs $\sum_i w_i v_i$: la fusion des différentes déformations localement affines se fait au niveau des champs de vitesse. Ce cadre permet de construire des grandes déformations localement affines de sorte que la fusion de ces déformations localement affines satisfait de bonnes propriétés. De plus ce cadre poly-affine est particulèrement adapté à certaines situations, comme par exemple le recalage de mouvement d'os [SPR12] ou cardiaques [MSBP15, RDSP15]. Cependant, les poids w_i (et donc les régions Ω_i d'action de chaque déformation localement affine) restent fixes pendant l'intégration du flot. Ceci empêche de construire de très grandes

déformations puisque la partie de la forme supposée être déplacée par une déformation T_i peut sortir de la région correspondante Ω_i.

1.2.6 Modèle GRID

Une autre approche étudie la fusion de déformations locales à travers la fusion des champs de vitesses : le modèle GRID [GSS07, Por10, PGV07, PV11, SSDG05]. Ici les déformations locales ne sont pas contraintes à être localement affines mais peuvent être plus génériques. Cette approche a pour but de modéliser la croissance par une série de petites déformations locales. Chacune de ces petites déformations est localisée autour d'un point appelé *graine* qui définit une région locale dans laquelle la petite déformation agit. La déformation globale est obtenue par intégration temporelle discrète d'une trajectoire de tels champs de vecteurs. À chaque instant une région d'activation est définie (correspondant à une région d'activation de gène) et une graine de cette région engendre un champs de vecteurs local. Cette activation de graine, la durée d'activation et d'autres paramètres sont modélisés à l'aide d'un cadre probabiliste et leurs lois sont estimées à l'aide d'un maximum de vraisemblance. Ce cadre permet de modéliser des croissances biologiques et l'approche probabiliste est adaptée à la considération de gène d'activation de croissance. Cependant il ne permet pas de construire un vocabulaire de déformations qui serait utilisé pour transformer une forme moyenne en différents sujets d'une population, alors que c'est un problème important comme précisé dans [Gre96, GCK12]. De plus les générateurs de déformations (graines) sont définies à chaque instant et donc ne sont pas transportés par la déformation globale pendant l'intégration du flot, ce qui peut mener à des faiblesses dans l'interprétation des trajectoires obtenues.

1.2.7 Modèle des Diffeons

Dans [You12] est présentée une approche paramétrique dans laquelle, au contraire, les générateurs des déformations sont contraints à être transportés par le flot de difféomorphismes. Dans ce cadre une variété S de *formes* est considérée, ainsi qu'un espace de champs de vecteurs V qui agit infinitésimalement dessus. Cet espace de champs de vecteurs définit un groupe G_V de diffeomorphismes qui peuvent s'écrire comme la valeur finale prise par un flot d'une trajectiore de V et G_V agit sur S. À chaque forme m de S est associée une famille finie de champs de vecteurs $\gamma_i(m)$, appelés *diffeons*, qui définit un sous espace V_m de V. Ensuite ne sont considérées que les trajectoires de champs de vecteurs $v : t \in [0,1] \mapsto v_t \in V$ telles que pour chaque t, v_t peut s'écrire $v_t = \sum_i \alpha_i(t)\gamma_i(m_t)$ avec $\alpha_i(t) \in \mathbb{R}$ et m_t la valeur transportée de la forme originale $m_{t=0}$ par le flot de v. Trouver une trajectoire optimale transportant une forme source m_0 vers une forme cible m_1 revient alors à un problème de contrôle optimal dans lequel le coût associé à chaque champs de vecteur v est $|v|_V^2$. Afin que ce coût soit facilement calculable en pratique, il faut restreindre les champs de vecteurs possibles $\gamma_i(m)$ à un certain ensemble : des translations locales dont les centres et les matrices de covariance sont des paramètres géométriques dépendant partiellement de la forme m et transportés par le flot. Les vecteurs de ces translations sont des variables de contrôle dont la trajectoire doit être optimisée pour transporter la forme source vers la forme cible. Ce cadre

définit une méthode pour construire des difféomorphismes sous contraintes dépendant des formes, permettant de définir une nouvelle métrique sur l'espace de formes. Deux exemples principaux sont donnés dans cet article. Dans le premier, les formes de S sont approximées par une discrétisation et cette discrétisation est utilisée pour approximer les trajectoires optimales de champs de vecteurs du cadre non-paramétrique. Dans le second exemple, des diffeons appelés "pushers" et "pullers" sont définis. Ils engendrent des translations locales centrées en des points différents de ceux de la forme et ont donc tendance à la "pousser" ou la "tirer". Cependant une méthode générique pour construire des diffeons adaptés à un problème particulier doit encore être définie. Par exemple une méthode pour fusionner différents types de *simples* diffeons en un diffeons *plus complexe* serait utile. De plus le coût associé à chaque champs de vecteurs est donné par le carré de sa norme, mesurant la "facilité" avec laquelle ce champs de vecteur est créé dans V et non la facilité avec laquelle il est créé en tant que combinaison des générateurs $\gamma_i(m)$: il n'est pas possible ici de définir une métrique plus adaptée un problème donné.

1.2.8 Conclusion

Grâce aux cadres non paramétrés (et non structurés) présentés Sections 1.2.1, 1.2.2 et 1.2.3, en choisissant une représentation appropriée des formes et un espace approprié de champs de vecteurs, il est possible d'obtenir un modèle de déformation permettant d'obtenir un bon appariement de formes données. Cependant, comme souligné dans la Section 1.2.4 ces cadres ne permettent pas d'imposer une structure particulière aux déformations générées. Ceci peut également mener à manque d'interprétabilité des résultats. Les cadres paramétriques présentés aux Sections 1.2.5, 1.2.6 et 1.2.7 permettent de surmonter partiellement ce problème. Dans ces modèles, en choisissant une famille de générateurs de champs de vecteurs, et en contraignant les champs de vecteurs à être engendrés par eux, il est possible d'imposer une certaine structure aux déformations. Cependant aucun de ces cadres ne donne de méthode générique pour définir facilement des générateurs qui seraient adaptés à l'étude de formes particulières et qui satisferaient également les propriétés nécessaires théoriquement (pour démontrer l'existence de trajectoires optimales par exemple). Nous avons également vu que définir l'évolution de ces générateurs pendant l'intégration du flot peut poser problème. Dans les cadres poly-affine (Section 1.2.5) et GRID (Section 1.2.6) il n'est pas possible de choisir cette évolution. Dans le dernier cadre des Diffeons (Section 1.2.7) cette évolution est plus générique puisqu'elle vient d'une action infinitésimale sur un espace de formes et peut donc être modifiée. Cependant pour des raisons numériques les champs de vecteurs sont contraints à être des sommes de translations locales et donc ne peuvent pas être adaptée de manière générique à un problème donné. Une dernière limite importante de ces cadres est qu'il n'est pas possible de définir un coût associé aux champs de vecteurs générés qui serait *cohérent* avec les a priori introduits: ce coût est toujours construit grâce à la norme du champs de vecteur dans un espace de champs de vecteurs qui est *fixé*. Ainsi si deux choix de générateurs permettent de construire le même champs de vecteurs mais que l'un est plus *naturel* que l'autre, leurs coûts seront égaux : le coût ne tient pas compte de la structure imposée au modèle de déformation.

1.3 Résumé des contributions

L'objet de cette thèse est de définir un nouveau modèle de déformation permettant de construire des déformations pouvant être localement contraintes à un certain type de déformations et interprétables. Comme expliqué précédemment, il faudrait simultanément construire des coûts associés à ces déformations qui soit pertinents par rapport aux contraintes. De même que dans les modèles précédents (présentés aux Sections 1.2.5, 1.2.6 et 1.2.7), les déformations sont construites comme des flots de trajectoires de champs de vecteurs et l'incorporation de contraintes dans le modèle de déformation correspond à fixer une famille $\mathcal{V} \doteq (v_i)_{i\in I}$ de générateurs de champs de vecteurs. Une fois que cette famille est fixée, l'idée est de considérer seulement les champs de vecteurs pouvant s'écrire comme combinaison linéaire d'une sous-famille finie de $\mathcal{V}$, c'est-à-dire que nous imposons à chaque instant qu'il existe $(\alpha_k)_{1\leq k\leq N} \in \mathbb{R}^N$ de sorte que le champs de vecteurs puisse s'écrire $\sum_k \alpha_k v_{i_k}$. Contrairement au modèle GRID (Section 1.2.6, nous voulons considérer les trajectoires v_t de tels champs de vecteurs telles que les générateurs v_{i_k}sont *transportés* par le flot de v. Ce transport est réalisé grâce à une action des champs de vecteurs sur les générateurs, qui peut par exemple être l'action nulle (comme pour le modèle poly-affine) ou correspondre à une action sur une variable de formes qui paramétreraient les générateurs de manière similaire au modèle des Diffeons. En conséquence si les générateurs initiaux $v_{i_k}(t = 0)$ sont fixés, rechercher la meilleure trajectoire de champs de vecteurs transportant une forme source f_0 aussi proche que possible d'une forme cible f_1 correspond à chercher la trajectoire de variables α_k telle que la trajectoire correspondante de champs de vecteurs transporte f_0 aussi proche que possible de f_1. Les variable α_k correspondent à des *contrôles* et ce problème d'appariement est un *problème de contrôle*.
Le choix de la famille de générateurs de champs de vecteurs $\mathcal{V}$ est un point clé de ce cadre et l'idée est de définir ces générateurs de façon à ce qu'ils capture la variabilité de caractéristiques géométriques intéressantes entre les formes. Cette idée est dans l'esprit de l'approche développée dans [PFY+99, PTC00, YPJM01] où les auteurs définissent des atomes médiaux appelés *M-reps*, qui contiennent de une information géométrique locale et complexe. Différents types de ces atomes médiaux sont définis de sorte qu'ils peuvent décrire de manière pertinente la géométrie d'une forme donnée : par exemple l'extrémité d'une forme 2D peut être représentée grâce à un atome *corner-end* si c'est un coin, ou un atome *rounded* si c'est une portion de cercle. Ensuite un réseau de tels atomes médiaux donennt une représentation *structurée* et *interprétable* de la forme. Construire de tels réseaux et comparer les paramètres d'un atome particulier permet ensuite d'étudier la variabilité de la caractéristique correspondante indépendamment des autres types de variabilité. Dans notre approche nous adoptons un point de vue similaire mais (comme souligné précédemment) au lieu de décrire la géométrie de la forme elle-même, nous voulons décrire les *déformations* encodant la variabilité des formes. Ainsi nous définirons des atomes de déformation qui contiendront une information géométrique complexe à propos de la déformation qu'ils génèrent. De même que pour les M-reps, nous voulons construire une paramétrisation pertinente de ces atomes, de sorte que la représentation des déformations que nous obtenons soit facilement interprétable.

Nous présentons ici les principales notions et les principaux résultats de cette thèse.

Les démonstrations et des exemples variés seront présentés dans les chapitres 4 et 5.

1.3.1 Modèle de déformation

Nous utilisons ici les notions de flot de difféomorphismes et de forme présentées dans [Arg14] et rappelées en Section 3.2. Soient d et ℓ des entiers non nuls, $C_0^\ell(\mathbb{R}^d)$ est l'espace des champs de vecteurs de classe C^ℓ sur $\mathbb{R}^d$ dont les dérivées d'ordre inférieur ou égal à ℓ tendent vers zero à l'infini. On le munit de la norme $|v|_\ell = \sup\{|\frac{\partial^{\ell_1+\cdots+\ell_d} v(x)}{\partial x_1^{\ell_1}\cdots x_d^{\ell_d}}| \mid x \in \mathbb{R}^d, (\ell_1,\cdots,\ell_d) \in \mathbb{N}^d, \ell_1+\cdot+\ell_d \leq \ell\}$ telle sorte que ce soit un espace de Banach. On définit $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ l'espace des difféomorphismes C^ℓ de $\mathbb{R}^d$ qui convergent vers l'identité à l'infini. C'est un ouvert de l'espace de Banach affine $Id+C_0^\ell(\mathbb{R}^d)$ et il est donc muni d'une structure différentielle naturelle. Nous allons considérer des trajectoires particulières de $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ définies comme flot de trajectoires de $C_0^\ell(\mathbb{R}^d)$.

Proposition 1. *Soit v un élément de $L^1([0,1], C_0^\ell(\mathbb{R}^d))$, c'est-à-dire un champ de vecteurs dépendant du temps tel que $t \in [0,1] \mapsto |v(t)|_\ell$ est intégrable. Alors il existe une unique solution φ^v absolument continue, appelée **le flot** de v, au système d'équation suivant :*

$$\begin{cases} \dot{\varphi}^v(t) &= v(t) \circ \varphi^v(t) \\ \varphi(0) &= Id \end{cases}$$

où $\phi^v(t) \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$ pour tout $t \in [0,1]$.

Cette proposition nous permet de considérer des flots de champs de vecteurs dépendant du temps, où le "temps" t fait référence à la variable d'intégration. Comme nous sommes intéressés par la manière dont ces flots peuvent déformer des formes, nous devons préciser ce qu'est une forme et comment un difféomorphisme de $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ peut agir dessus. Nous utilisons la définition suivante, introduite dans [Arg14].

Definition 1. *Soit $\mathcal{O}$ une variété de Banach de dimension finie. Nous supposons que le groupe $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ agit continument sur $\mathcal{O}$ selon l'action*

$$\begin{array}{rcl} \mathrm{Diff}_0^\ell(\mathbb{R}^d) \times \mathcal{O} & \to & \mathcal{O} \\ (\varphi, o) & \mapsto & \varphi \cdot o \end{array} \tag{1.1}$$

*On dit que $\mathcal{O}$ est un C^k **espace de formes d'ordre** ℓ sur $\mathbb{R}^d$ si les conditions suivantes sont satisfaites :*

1. *Pour chaque $o \in \mathcal{O}$, $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto \phi \cdot o$ est Lipschitzienne par rapport à la norme $|\cdot|_\ell$ et différentiable à $Id_{\mathbb{R}^d}$. Cette différentielle sera notée ξ_o et appelée **l'action infinitésimale** de $C_0^\ell(\mathbb{R}^d)$.*

2. *L'application $\xi : (o,v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto \xi_o v$ est continue et sa restriction à $\mathcal{O} \times C_0^{\ell+k}(\mathbb{R}^d)$ est de classe C^k.*

*Un élément o du $\mathcal{O}$ est appelée **forme**, et $\mathbb{R}^d$ est **l'espace ambiant**.*

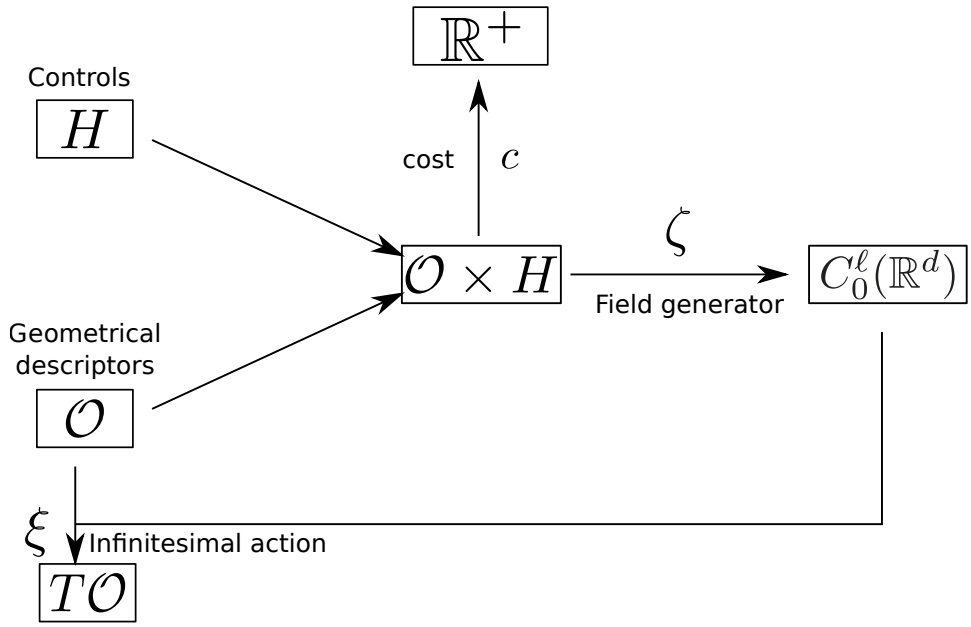


Figure 1.1: Schematic view of a deformation module.

Notre cadre théorique repose sur l'introduction de la notion de *module de déformation*. Un module de déformation est une structure capable de générer des champs de vecteurs d'un type choisi et paramétré en petite dimension.

Definition 2. *Soient $k, \ell \in \mathbb{N}^*$. On dit que $M = (\mathcal{O}, H, \zeta, \xi, c)$ est un* ***module de déformation*** *C^k* ***d'order*** *ℓ, de* ***descripteurs géométriques*** *dans $\mathcal{O}$, de* ***controls*** *dans H,* ***d'action infinitesimale*** *ξ, de générateur de champs ζ et de coût c si*

- *$\mathcal{O}$ est un C^k espace de formes d'ordre ℓ d'action infinitésimale $\xi : C_0^\ell(\mathbb{R}^d) \times \mathcal{O} \longrightarrow T\mathcal{O}$,*
- *H est un espace vectoriel euclidien de dimension finie,*
- *$\zeta : (o, h) \in \mathcal{O} \times H \to (o, \zeta_o(h)) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d)$ est continue, avec $h \mapsto \zeta_o(h)$ linéaire and $o \mapsto \zeta_o$ de classe C^k,*
- *$c : (o, h) \in \mathcal{O} \times H \to c_o(h) \in \mathbb{R}^+$ est une application continue telle que $o \mapsto c_o$ est lisse et pour tout $o \in \mathcal{O}$, $h \mapsto c_o(h)$ est une forme quadratique définie positive sur H, définissant donc une métrique sur $\mathcal{O} \times H$.*

Une vue schématique d'un module de déformation est présentée Figure 1.1. Les champs de vecteurs générés par un module de déformation sont ainsi paramétrés par deux variables: les descripteurs géométriques et les controls. Les descripteurs géométriques vont paramétrer la géométrie (par exemple la localisation) du champs de vecteurs généré. Les controls vont eux paramétrer la force avec laquelle le champs de vecteur est utilisé. Des exemples nombreux et variés seront présentés dans le Chapitre 4, Section 4.2. Les descripteurs géométriques appartiennent à un espace de formes et seront donc appelés

formes. Nous utiliserons également ce terme pour désigner les données à étudier puisque ce seront également des formes au sens de la définition 1. En général ces deux types d'objets (descripteurs géométriques et données à étudier) ne seront pas égaux. Cependant, étant donné un espace de formes $\mathcal{F}$ de données à étudier, ainsi que son action infinitésimale $\xi^{\mathcal{F}}$, on peut définir un module de déformation appelé *module de déformation silencieux induit par* $\mathcal{F}$. Ce module de déformation est appelé silencieux car ils va générer des champs de vecteurs toujours nuls. On le définit par un espace de descripteurs géométriques égal à $\mathcal{F}$, un espace de controls nul, un générateur de champs nul, une action infinitésimale égale à $\xi^{\mathcal{F}}$ et un coût nul. L'intérêt de ce type de module de déformation silencieux sera rapidement présenté Section 1.3.3.2, et plus longuement dans le Chapitre 5.

Nous allons nous restreindre aux modules de déformation qui satisfont la proposition suivante, assurant que leurs coûts sont reliés (mais non nécessairement égaux) à la norme du champ de vecteurs généré.

Definition 3. *Soit $M = (\mathcal{O}, H, \zeta, \xi, c)$ un module de déformation C^k d'ordre ℓ. On dit que M satisfait la* ***condition de plongement uniforme (UEC)*** *s'il existe un espace de Hilbert de champs de vecteurs V continûment inclus dans $C_0^{\ell+k}(\mathbb{R}^d)$, et une constante $C > 0$ tels que pour tout $o \in \mathcal{O}$, et pour tout $h \in H$, $\zeta_o(h) \in V$ et*

$$|\zeta_o(h)|_V^2 \leq C c_o(h).$$

La notion de module de déformation permet d'incorporer des contraintes dans le modèle de déformation. En effet, si on fixe un module de déformation, alors on peut ne considérer que les champs de vecteurs pouvant être créés à l'aide du générateur de champs. C'est une première étape dans l'introduction de contraintes. Les contraintes que l'on voudrait incorporer peuvent correspondre à un une connaissance additionnelle que l'on aurait sur la population de formes, ou alors à un point de vue particulier selon lequel on voudrait étudier la population. En pratique ces contraintes peuvent être relativement complexes et définir le module de déformation correspondant peut être délicat. Cependant il est relativement facile de définir des modules de déformations correspondant à des contraintes *simples*. Nous allons alors définir la combinaison de modules de déformations, permettant de définir des modules de déformations plus complexes. Une vue schématique d'une combinaison de trois modules de déformations est présentée Figure 1.2.

Definition 4. *Soient $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, des modules de déformation C^k d'ordre ℓ. On définit le* ***module combiné*** *des modules M^l par $\mathcal{C}(M^l, l = 1 \cdots L) = (\mathcal{O}, H, \zeta, \xi, c)$ où $\mathcal{O} \doteq \prod_l \mathcal{O}^l$, $H \doteq \prod_l H^l$ and for $o = (o^l)_l \in \mathcal{O}$, $\zeta_o : h = (h^l) \in H \mapsto \sum_l \zeta_{o^l}^l(h^l)$, $\xi_o : v \in C_0^\ell(\mathbb{R}^d) \mapsto (\xi_{o^l}^l(v))_l \in T_o\mathcal{O}$ et $c_o : h = (h^l) \in H \mapsto \sum_l c_{o^l}^l(h^l)$.*

Une propriété clé de notre cadre est la stabilité par combinaison suivante:

Proposition 2. *Si $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, sont des modules de déformation C^k d'ordre ℓ, alors $\mathcal{C}(M^l, l = 1 \cdots L)$ l'est également. De plus si chaque module de déformation M^l satisfait UEC, alors $\mathcal{C}(M^l, l = 1 \cdots L)$ satisfait également cette condition.*

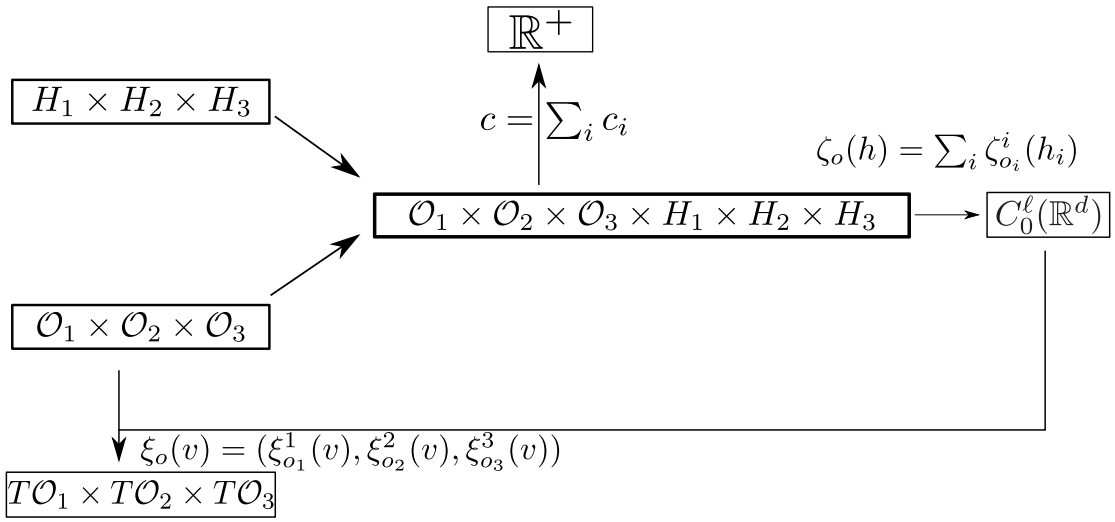


Figure 1.2: Schematic view of a combination of three deformation modules.

Ainsi afin de définir un module de déformation adapté à une contrainte complexe, il suffit de la voir comme la combinaison de contraintes plus simples et de combiner les modules de déformations correspondant à ces contraintes simples. Nous allons maintenant expliquer comment construire des grandes déformations à l'aide de ce cadre modulaire. Fixons un module de déformation $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfaisant UEC. Nous allons considérer les trajectoires de champs de vecteurs $v : t \in [0,1] \mapsto v_t \in V$ qui sont modulaires, c'est-à-dire qui peuvent s'écrire à chaque instant $v_t = \zeta_{o_t}(h_t)$ avec $(o_t, h_t) \in \mathcal{O} \times H$. Durant l'intégration de la trajectoire nous voulons que le descripteur géométrique soit transporté par le flot et donc, en notant φ^v le flot de v, nous allons imposer qu'à chaque instant v_t appartienne à $\zeta_{o_t}(H)$ avec $o_t = \varphi^v_t(o_{t=0})$.

Definition 5. *Soient $a, b \in \mathcal{O}$. On note $\Omega_{a,b}$ l'ensemble des courbes mesurables $t \mapsto (o_t, h_t) \in \mathcal{O} \times H$ avec o_t absolument continue (a.c.), commençant en a, finissant en b, telles que pour presque tout $t \in [0,1]$, $\dot{o}_t = \xi_{o_t}(v_t)$, où $v_t \doteq \zeta_{o_t}(h_t)$, et*

$$E(o,h) \doteq \int_0^1 c_{o_t}(h_t)dt < \infty.$$

*La quantité $E(o,h)$ est appelée **l'énergie** de (o,h) et $\Omega_{a,b}$ est l'ensemble des **chemins contrôlés d'énergie finie** débutant en a et finissant en b.*

Si la condition UEC est satisfaire, il est possible de construire des grandes déformations à partir des trajectoires de $\Omega_{a,b}$:

Proposition 3. *Soit $M = (\mathcal{O}, H, \zeta, \xi, c)$ un module de déformation satisfaisant UEC et soient $a, b \in \mathcal{O}$. Soit $(o,h) \in \Omega_{a,b}$, pour tout t on définit $v_t \doteq \zeta_{o_t}(h_t)$.Alors $v \in L^2([0,1], V) \subset L^1$, son flot φ^v existe, $h \in L^2([0,1], H)$ et pour tout $t \in [0,1]$, $o_t = \phi^v_t \cdot o_0$. Le difféomorphime final $\varphi^v_{t=1}$ est une **une grande déformation modulaire générée par** a.*

L'énergie associée à une trajectoire (o,h) $\Omega_{a,b}$ est $E(o,h) = \int c_o(h)$. Il est important de noter que cette énergie ne dépend pas directement de la norme du champs de vecteurs $\zeta_o(h)$ généré à chaque instant. Il dépend en fait de la manière dont il peut être généré par le descripteur géométrique. Cette énergie correspond donc à une *information modulaire.*

1.3.2 Formalisme sous-Riemannien

Fixons $M = (\mathcal{O}, H, \zeta, \xi, c)$ un module de déformation C^k d'ordre ℓ satisfaisant UEC. Nous allons présenter comment les grandes déformations modulaires construites précédemment permettent de munir l'espace des descripteurs géométriques $\mathcal{O}$ d'une norme adaptée. Nous définissons le morphisme de fibrés vectoriels suivant:

$$\rho : (o,h) \in \mathcal{O} \times H \mapsto (o, \xi_o \circ \zeta_o(h)) \in T\mathcal{O}$$

On note pour $(o,h) \in \mathcal{O} \times H$, $\rho(o,h) = (o, \rho_o(h))$. Ce fibré ρ associe à chaque descripteur géométrique $o \in \mathcal{O}$ le sous-espace $\rho_o(H) \subset T_o\mathcal{O}$ de toutes les vitesses de o qui peuvent être générées par lui même. De plus le coût c induit une métrique Riemannienne lisse g sur le fibré $\mathcal{O} \times H$. Alors $(\mathcal{O} \times H, g, \rho)$ définit une structure sous-riemannienne sur $\mathcal{O}$. Cette structure permet de munir $\mathcal{O}$ d'une distance sous-Riemannienne D qui est liée au coût c par la proposition suivante :

Proposition 4. *Soient $a, b \in \mathcal{O}$ tels que $D(a,b)$ est finie. Alors*

$$Dist(a,b)^2 = \inf\{\int_0^1 c_o(h) \mid h \in L^2([0,1], H), \dot{o} = \rho_o(h), o_{t=0} = a, o_{t=1} = b\}$$

De plus, sous de faibles conditions, comme M satisfaite UEC, on peut montrer le résultat suivant :

Theorem 1. *Si $\Omega_{a,b}$ est non vide, l'énergie E atteint son minimum sur $\Omega_{a,b}$.*

1.3.3 Comparaison modulaire de formes

Nous utilisons cette distance sous-Riemannienne pour mesurer les différences entre deux formes.

1.3.3.1 Appariement de formes

Soit $M = (\mathcal{O}, H, \zeta, \xi, c)$ un module de déformation C^k d'ordre ℓ satisfaisant UEC. Nous supposons tout d'abord que nous voulons comparer deux formes de $\mathcal{O}$. En pratique ce n'est pas vrai, ce cas général sera présenté dans la section suivante. Si la distance $D(a,b)$ est finie, on peut étudier les différences entre ces deux formes en cherchant la *meilleure* grande déformation modulaire transformation a en b. Pour ceci on minimiserait $\int_0^1 c_o(h)$ pour $(o,h) \in \Omega_{a,b}$. Cependant en pratique les formes que l'on considère sont bruitées et donc on ne s'intéresse pas à ce cas d'appariement exacte mais au cas de l'appariement inexacte. Celui-ci revient à minimiser

$$J_{a,b}(h) = \frac{1}{\sigma^2}\mu(o_{t=1}, b) + \int c_{o_t}(h_t)dt$$

où $\sigma \in \mathbb{R}_+^*$, $o_{t=0} = a$ et $\dot{o} = \xi_o \circ \zeta_o(h)$. Le premier terme de cette somme est appelée le **terme d'attache aux données** et le second **terme de régularité**. Le résultat suivant permet de charactériser les trajectoires optimales de controls :

Theorem 2. *Nous rappelons que $M = (\mathcal{O}, H, \zeta, \xi, c)$ est un module de déformation C^k d'ordre ℓ satisfaisant UEC, avec $k, l \geq 2$. Nous supposons que μ est C^1. Si $h \in L^2([0,1], H)$ minimise $J_{a,b}$, alors il existe une trajectoire $\eta : t \in [0,1] \longrightarrow \eta_t \in T^*_{o_t}\mathcal{O}$ telle que, avec*

$$\mathcal{H} : (o, \eta, h) \in T^*\mathcal{O} \times H \mapsto \Big(\eta | \xi_o(\zeta_o(h))\Big) - \frac{1}{2} c_o(h)$$

*le **Hamiltonien** du système, $\eta_{t=1} = -\partial_1 \mu(o_{t=1}, b)$ et (dans un système local de coordonnées)*

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \xi_o \circ \zeta_o(h) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o} \\ \frac{\partial \mathcal{H}}{\partial h} & = & 0 \end{array} \right. \tag{1.2}$$

Remark 1. *Comme c_o est définie positive, il existe un opérateur symétrique inversible $C : o \in \mathcal{O} \mapsto C_o \in L(H, H^*)$ tel que pour tout $(o, h) \in \mathcal{O} \times H$, $c_o(h) = (C_o h | h)$. Alors la troisième égalité de (1.2) permet d'obtenir h: $h = C_o^{-1} \rho_o^* \eta$ avec $\rho_o = \xi_o \circ \zeta_o$ et ρ_o^* telle que $(\eta | \rho_o(h))_{T_o^*\mathcal{O}} = (\rho_o^* \eta | h)_H$.*

À chaque moment initial $\eta_{t=0} \in T_a^*\mathcal{O}$ peut être associée une trajectoire de controls en intégrant le système d'équations (1.2) $\eta_{t=0} \in T_a^*\mathcal{O}$. Une telle trajectoire est appelée *géodésique*, ou *optimale*. Alors, afin d'obtenir le minimum de $J_{a,b}$ nous utilisons le principe du *shooting géodésique*: nous estimons grâce à une descente de gradient un moment initial $\eta_{t=0} \in T_a^*\mathcal{O}$ tel que la trajectoire de controls correspondante minimise $J_{a,b}$.

1.3.3.2 L'appariement de formes en pratique

Contrairement à ce que nous venons de présenter, en général on veut apparier deux formes f_0 et f_1, appartenant à un même espace de formes $\mathcal{F}$, à l'aide d'un module de déformation $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$. Même si à première vue ce problème est différent du précédent, nous allons montrer qu'il s'y ramène facilement. En effet, construisons $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ le module de déformation silencieux induit par $\mathcal{F}$ défini Section 1.3.1. Ce module de déformation ne génère ainsi que des champs de vecteurs nuls, mais ressent l'action de champs de vecteurs extérieurs. Nous pouvons alors considérer sa combinaison $M \doteq C(M^1, M^2) \doteq (\mathcal{O}, H\zeta, \xi, c)$ avec M^1. Les champs de vecteurs générés par ce nouveau module de déformation, et leurs coûts associés, sont les mêmes que ceux générés par M^1 puisque par définition, pour $o = (o^1, f) \in \mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$ et $h = (h^1, 0) \in H = H^1 \times \{0\}$, le champs de vecteur généré est $\zeta_o(h) = \zeta^1_{o^1}(h^1) + \zeta^2_f(0) = \zeta^1_{o^1}(h^1)$ et le coût est $c_o(h) = c^1_{o^1}(h^1) + c^2_f(0) = c^1_{o^1}(h^1)$. Cependant les trajectoires optimales de controls ne seront pas le même pour M que celles pour M^1 car dans le cas du module de déformation M nous tenons compte de l'action des champs de vecteurs générés à la fois sur la composante dans $\mathcal{O}^1$ et celle dans $\mathcal{F}$ grâce à l'action infinitésimale combinée $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (\xi^1_{o^1}(v), \xi^{\mathcal{F}}_f(v))$ avec $o = (o^1, f)$. Un exemple simple où

les trajectoires optimales générées par M^1 et M^2 sont différentes est présentée Section 5.4.1.2.
Grâce à cette combinaison, apparier f_0 et f_1 à l'aide du module de déformation M^1 revient à résoudre le problème d'appariement présenté dans la section précédente pour le nouveau module de déformation M, avec un terme d'attache aux données $\mu(o_{t=1}, b)$ ne dépendant que de la composante dans $\mathcal{F}$. Si l'on fixe la valeur initiale $o_0^1 \in \mathcal{O}^1$, les trajectoires optimales de controls sont alors paramétrées par une valeur initiale de moment $\eta \in T^*_{(o^1,f_0)}\mathcal{O}$. Il faut alors optimiser à la fois la composante de $\mathcal{O}^1$ du descripteur géométrique initial, et également le moment initial $\eta \in T^*_{(o^1,f_0)}\mathcal{O}$ de telle sorte que la trajectoire de controls correspondante minimise

$$J(h) = \int_0^1 c_o(h) + \mu(f(t=1), f_1)$$

où $o = (o^1, f)$ débute à (o_0^1, f_0) et vérifie $\dot{o} = \xi_o \circ \zeta_o(h) = (\xi^1_{o^1} \circ \zeta^1_{o^1}(h^1), \xi^{\mathcal{F}}_f \circ \zeta^1_{o^1}(h^1))$ avec $h = (h^1, 0) \in H = H^1 \times \{0\}$.

Il est important de noter ici que le moment initial a deux composantes : une dans $T^*_{o^1}\mathcal{O}^1$ et une dans $T^*_{f_0}\mathcal{F}$. Ainsi les trajectoires de controls optimales sont paramétrées en dimension plus grande que la dimension des formes f_i. Dans la Section 5.4 nous présenterons un cadre dans lequel on considère un sous-espace de trajectoires optimales paramétrées en plus petite dimension.

1.3.4 Analyse modulaire de la variabilité d'une population de formes

Ici nous expliquons comment ce modèle modulaire permet d'étudier la variabilité au sein d'une population de formes. Soient $f^1_{target}, \cdots, f^P_{targets}$ des formes d'un espace de formes commun $\mathcal{F}$. Étudions cette population de formes à l'aide d'un module de déformation $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$. Comme précédemment, nous construisons le module silencieux induit par $\mathcal{F}$, et nous le combinons avec $\tilde{M}$. Nous obtenons alors un module de déformation $M = (\mathcal{O}, H, \zeta, \xi, c)$ générant les mêmes champs de vecteurs que $\tilde{M}$, mais qui permet de tenir compte de leur action sur les formes de $\mathcal{F}$. Dans ce cadre estimer un atlas des formes f^k_{target} à l'aide de $\tilde{M}$ revient à minimiser

$$J : (o_{temp}, h^1, \cdots, h^P) \in \mathcal{O} \times (L^2([0,1], H))^P \mapsto \frac{1}{\sigma^2} \sum_k \mu(f^k_{t=1}, f^k_{target}) + \int_0^1 c_{o^k}(h^k)$$

avec pour chaque k, $o^k_{t=0} = o_{temp}$ et $\dot{o}^k = \xi_{o^k} \circ \zeta_{o^k}(h^k)$. On estime ici une valeur initiale de descripteur géométrique commune à tous les sujets, mais les trajectoires o^k sont spécifiques à chaque sujet puisqu'elles dépendent des trajectoires de controls h^k. De même que dans le cas de l'appariement, les trajectoires optimales de controls peuvent être paramétrées par des moments initiaux attachés à o_{temp}. Estimer un atlas revient alors à estimer un *template* $o_{temp} \in \mathcal{O}$ et P moments initiaux $\eta^k_0 \in T^*_{o_{temp}}\mathcal{O}$ de sorte que les trajectoires de controls correspondantes minimisent J. Nous insistons sur le fait que le template o_{temp} appartient à $\mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$. Sa composante dans $\mathcal{F}$ correspond

intuitivement à l'idée de forme moyenne, alors que sa composante dans $\mathcal{O}^1$ correspond à une caractérisation géométrique de la variabilité au sein de la population de formes.

1.3.5 Un algorithm "plug-and-play"

L'algorithme permettant d'implémenter l'appariement de formes et l'estimation d'atlas de formes sera présenté dans le Chapitre 6. Nous utilisons une implémentation orientée objet car notre cadre y est très adaptée. Une classe abstraite appelée ABSTRACTMODULE définie la notion de module de déformation, et tous les types de modules de déformations héritent de cette classe. Les méthodes de cette classe implémentent les fonction ζ, ξ et c définissant le module de déformation. Ensuite, une méta-classe MODULARDIFFEO a pour attribut une liste d'ABSTRACTMODULE. Ses méthodes implémentent les règles simples de combinaisons présentées dans la Section 1.3.1. Grâce à cette classe, on peut obtenir les trajectoires optimales de controls. L'idée importante est que cette classe n'utilise que les *méthodes abstraites* de la classe abstraite ABSTRACMODULE et n'est à implémenter qu'une fois. Ensuite, si l'on veut utiliser un nouveau type de module de déformation, il suffit de définir ses fonctions ζ, ξ et c, et il est directement incorporé au modèle de déformation global. En pratique une grande quantité de modules de déformations de base a été implémentée, et il suffit seulement de choisir ceux que nous voulons utiliser pour étudier un problème particulier. Si on a un savoir a priori sur la population de formes que l'on veut étudier, on peut utiliser une combinaison des modules de déformations de base correspondant. Si, au contraire, on n'a pas de telle connaissance a priori, il est possible d'utiliser par exemple une combinaison de modules de déformation générant des translations, rotations et scalings locaux à différentes échelles (un exemple sera présenté à la Section 7.4. Nous avons intégré cet algorithm dans le logiciel Deformetrica [DPC+14], de sorte que le terme d'attache aux données $\mu(f_k(t=1), f_k^{target})$, et ses dérivées sont automatiquement implémentés pour une large collection de formes : nuages de points, courbes, maillages de surfaces en 2D et 3D. Nous présentons dans le Chapitre 7 plusieurs problèmes d'appariement et de constructions d'atlas que nous avons étudiés à l'aide de notre algorithme modulaire.

Chapter 2

Introduction

Contents

2.1 Shape analysis context in medical imaging

In both medical imaging and computer vision, a major challenge is the automatic interpretation of imaging data. These data are representations of real 2D or 3D objects: they can be either images (MRI for instance) or geometrical data (such as curves or surfaces obtained through a segmentation process). They encode complex information, and even though sometimes part of this information can be visually interpreted, in general it is not possible for a human eye to comprehend these shapes in their totality. For instance, given a brain MRI, deepest folds in the cerebral cortex can be manually labelled; but identifying thinner structures remains a hard task. Similarly, studying resemblances and dissimilarities between three or four subjects can sometimes be done manually but in order to understand anatomical differences within larger populations, automatic methods are needed. In practice studying such large populations is necessary for instance in order to detect anatomical characteristics associated to a particular disease, or to assemble anatomical differences amongst a population into key classes.
Shape analysis precisely aims at developing methods allowing an automatic and global understanding of such shape data. An efficient approach to analyse automatically a shape is the development of tools allowing to automatically *compare* it to a population of similar data which have already been studied. To do this, one needs to develop beforehand methods to analyse a *population of shapes* and then to be able to *compare* shapes. The analysis of a population of shapes can be performed via first the computation of a mean shape, called *template*, which reflects the underlying structure of the population; and second the comparison of shapes of the population to this template, in order to understand its variability. Thus, the key point here is the development of methods to compare shapes.
A successful approach, pioneered by D'Arcy Thompson [T+42], is to adopt a geometrical point of view and to study deformations that transform the first shape into the second one. In the restricted case where shapes are landmarks, an idea presented by F. L. Bookstein in [Boo97, B+89] is to interpolate the landmark displacements between two given shapes by splines, leading to a dense and smooth vector field. Even though this approach enabled to solve many problems, in particular in medical imaging, it suffers from an important limitation: there is no guarantee that the resulting vector field is invertible. This is a problem: an interest of this geometrical framework is to match corresponding regions of the data and then a one-to-one deformation is required. In its Pattern Theory [GK93, GCK12, Gre96, Gre93], U. Grenander developed a new idea to study shapes: they are considered under the action of a group of deformations. In [MCAG93], an *ideal* shape, namely a template, is considered and invertible deformations matching this ideal shape to data shapes are built. Then a knowledge on the ideal template, such as segmentations, can be automatically adapted to the new data. However, in practice, deformations that are used are linear approximations of diffeomorphism, namely *small deformations*, and then do not allow large differences between the template and data. In [CRM96] a first framework to build systematic *diffeomorphic large deformations* is presented: deformations are obtained through the integration of a trajectory of vector fields. One considers a trajectory φ of diffeomorphisms starting at

$\varphi_{t=0} = Id$ and satisfying:

$$\dot{\varphi}_t = v_t \circ \varphi_t.$$

In this framework, the set of diffeomorphisms that will be used to study shapes is not directly defined: one first specifies trajectories of vector fields and then builds corresponding large deformations by integrating them. This construction of diffeomorphisms has been widely used in shape analysis [AG04, JDJG04, KWB08, RFS03, Thi98, Tog98, VPPA09, WBR+07] as it enables to build large deformations by specifying only infinitesimal displacements. Various choices of trajectories of vector fields lead to various deformation models. In a first large class of models, presented in Sections 2.2.1, 2.2.2 and 2.2.3, a space of vector fields is chosen and one considers integrable trajectories of this space. In these frameworks there is no structure imposed on deformations in advanced: this allows to build rich sets of diffeomorphisms and adapted matching deformations between shapes, but prevents from natural interpretation of them. Indeed, as vector fields are not parametrized beforehand with meaningful variables, these frameworks do not provide tools to understand resulting deformations. In order to overcome this issue, simultaneously, were developed several parametric frameworks, an overview will be presented in Sections 2.2.5,2.2.6 and 2.2.7. In these approaches at each time of the trajectory the vector field is constrained to be a combination of a few generators that create particular vector fields. By choosing particular generators, one can build trajectories of diffeomorphisms corresponding to a certain point of view–a *prior*–one wants to have on shapes. However, to our knowledge non of these frameworks allow to build generators corresponding to a generic, complex, prior. Besides, non of them allow to equip these vector fields with a metric depending on these prior. This is what we intend to do: the goal is the construction of a vocabulary that would be used to describe differences between shapes, and a metric that would measure these differences taking into account the choice of vocabulary.
This Chapter is organized as follow: in Section 2.2 we present an overview of the different deformation models that have been developed and we introduce the reasons that led us into building a *modular* deformation model. In Section 2.3 we will present the first two approaches that we have developed before defining the notion of deformation modules. In Section 2.4 can be found a summary of the contributions of this thesis.

2.2 Diffeomorphisms in Computational Anatomy

2.2.1 LDDMM Framework

The seminal works of U. Grenander [Gre93] and G.E. Christensen and al. [CRM96] define notions of template and diffeomorphic large deformations defined as flow of trajectories of vector fields. In [DGM98] authors study the required theoretical conditions under which the existence of an optimal trajectory of vector fields transporting a source shape as close as possible to a target shape can be proved. They also study this problem from a Bayesian point of view. In [Tro95, Tro98], A. Trouvé introduced an original generic framework to study patterns in a population of shapes. In this article, particular shapes (*patterns*) are considered: they are measurable functions from a finite dimensional

Riemannian compact manifold M without boundary to a finite dimensional manifold X. This includes for instance the case of closed curves in $\mathbb{R}^2$. In order to study these shapes, a Banach space of vector fields B is fixed. Then the group A_B of final values of flows of trajectories of vector fields lying in B is built. This framework gives a rigorous mathematical framework to build diffeomorphic large deformations as well as a metric (and then a distance) on this group of diffeomorphisms. In order to study the differences between two shapes f_0 and f_1, the idea is to search a diffeomorphism in A_B (*ie* a trajectory of vector fields in B) which is as close as possible to the Identity and which transports f_0 as close as possible to f_1. The existence of such optimal trajectories of vector fields is proved. Besides, this distance allows to measure the resemblance between two shapes and is used in this article to associate an observed shape with a pattern-shape chosen among predefined pattern-shapes. Euler-Lagrange equations of optimal trajectories were studied in [BMTY05], leading to a framework known as the Large Deformations Diffeomorphic Metric Mapping (LDDMM) [MYT14, YAM09, You10]. Even though the definition of shapes that is considered in this framework matches many examples, no formalised definition is given.

2.2.2 Required notion: shape spaces

The formalisation of the notion of shape has been considered from many points of view [BBM14, Ken84, KBCL09]. In this thesis we will consider the one developed by S. Arguillère [Arg14, Arg15a, Arg15b]. In [Arg15a], a shape space $\mathcal{O}$ is defined as an open set of a Banach space on which the group G of diffeomorphisms acts continuously (see Section 3.2 for a more precise definition). A shape is then defined as an element of this space. In this framework one sets a space of vector fields V of $\mathbb{R}^d$ ($d = 2$ or 3) which is supposed to be a Reproducing Kernel Hilbert Space (RKHS): it is a Hilbert space such that for all x in $\mathbb{R}^d$, $v \in V \mapsto v(x) \in \mathbb{R}^d$ is a continuous function (see Section 3.1 for a more precise definition). Once this space V is setted, the group G_V made of final values of flows of trajectories of vector fields lying in V is built. This group G_V is a subset of G, it is equipped with a metric thanks to V and will be used to study shapes of $\mathcal{O}$. As in [Tro95], in order to study the differences between two shapes f_0 and f_1, one searches a diffeomorphism in G_V (*ie* a trajectory of vector fields in V) which is as close as possible to the Identity and which transports f_0 as close as possible to f_1. Its existence is proved and it is shown that the corresponding trajectory of vector fields v_t can be parametrized by an initial dual variable named *momentum*. Besides at each time t, v_t is *carried* by the shape in a sense that will be detailed in 3.2, see Proposition 17. From the metric on G_V can be built a sub-Riemannian metric on $\mathcal{O}$ and then one can measure the differences between f_0 and f_1. This measurement depends on the space V and one challenge in this framework is to choose a space V which is adapted to a given problem. Several choices of shape spaces and RKHS V of the literature correspond to this generic framework.

2.2.3 Various shape spaces and RKHS: various frameworks

In [Gla05, GVM04] is detailed the cases of landmarks but also measures and currents which allow to consider unlabelled sets of points, and unparametrized curves and sur-

faces. Authors restricted to the case of scalar gaussian RKHS, *i.e.* when the kernel defining the RKHS is a scalar Gaussian one and is then defined by a scalar parameter called *scale*, representing the characteristic scale of deformations. As the choice of the scale σ of the RKHS became a important issue, several multi-scale frameworks have been developed. In [RVW+11] the kernel of the RKHS is a sum of Gaussian kernels of different scales. This leads to optimal trajectories of vector fields capturing variability at different scales simultaneously and which look more natural in cases of differences at several characteristic scales between data. Another multi-scale approach was initiated in [SLNP11] where instead of one space of vector fields, a family W of such spaces (each one being a RKHS at a particular scale) is considered. The action of a family of vector fields belonging to W on a shape is built as the sum of the actions of all vector fields. However it is shown in [BRV12] that this approach is equivalent to the previous one [RVW+11], this issue was studied in [SLNP13]. In all these frameworks, shapes that are considered are of zeroth-order in the sense that they carry information about points' location. It can be shown [Gla05] that for optimal trajectories, at each time the vector field is a sum of translations localised by the kernel function. In [SNDP13], another category of shapes are considered. Here they carry higher-order information: they are images and not only values of the image at each point are considered, but also its derivatives at these points. As a consequence, optimal trajectories of vector fields are parametrized by higher-order momenta and then vector fields of optimal trajectories are expressed thanks to derivatives of the kernel, allowing locally more complex deformations as, for instance, locally affine ones. A theoretical study of this framework is given in [Jac13], in particular reductions due to symmetries are provided.
In all these frameworks, by choosing an appropriate space of vector fields, one can build large deformations adapted to a particular problem. However, momenta parametrizing optimal trajectories are of same dimension as the shapes that are considered, which can be very high in practice. This parametrization of optimal trajectories can therefore be redundant. In a sparse model presented in [DPC+14], S. Durrleman considers geodesics generated by a small number of landmarks so that they are parametrized in low dimension. Then one can study how these geodesics lying in a low-dimensional space can deform shapes of high dimension. In this framework the idea is to find the real necessary dimension of trajectories that needs to be considered in order to study the variability of a given population. However, these vector fields are always sums of local translations (as they are geodesics for landmarks), therefore limiting the interpretations of non linear patterns.

2.2.4 More structured deformations ?

A general limitation in these non-parametric settings is that the nature of optimal vector fields depends on the nature of shapes and not on the nature of the differences that one wants to study. It is not possible to impose a certain pattern of deformation, corresponding to a prior knowledge. Yet this is necessary in some cases: for instance with biological images, some deformation patterns can be totally irrelevant and it might be interesting to build a deformation model which does not allow them but favour more realistic ones. This is one reason why were developed *parametric* approaches in which

deformations are *structured* beforehand. In these frameworks vector fields at each time are constrained to be a sum of a few interpretable vector fields, parametrized in small dimension thanks to *control* variables, so that the deformation pattern is controlled and can be interpreted from controls' values. The idea is here to reduce the set of considered diffeomorphisms in order to improve their comprehension and their specificity to a given problem.

2.2.5 Poly-affine framework

In a poly-affine framework presented in [APA05, ACAP09], the goal is to build deformations that are the fusion of different local affine deformations. If locally affine deformations are added, the resulting deformation is in general non invertible. This is why in this framework deformations are generated by integrating trajectories of vector fields. To an affine deformation T is associated a trajectory of vector field whose flow is equal to T at time $t = 1$. Then if one wants to fuse different local affine deformations T_i acting in areas Ω_i defined by weighting functions w_i, one defines the associated trajectories of vector fields v_i and then integrates the trajectory of vector fields $\sum_i w_i v_i$: the fusion of different local affine deformations is made at the velocity field level. This framework enables to build large deformations that are locally affine, so that the fusion of these local affine deformation satisfies good properties. Besides, this poly-affine framework is particularly adapted to certain situations, for instance in the registration of bones motion [SPR12] or for cardiac motions [MSBP15, RDSP15]. However, weights w_i (and then areas Ω_i of action for each local affine transformation) remain fixed during the integration of the flow. This prevents very large deformations as the part of the shape that is supposed to be displaced by the deformation T_i can leave from the corresponding area Ω_i.

2.2.6 GRID model

Another approach studies the fusion of local deformations through the fusion of velocity fields: the GRID model [GSS07, Por10, PGV07, PV11, SSDG05]. Here local deformations are not constrained to be affine but can be more generic. This approach aims at modelling growth by a sequence of local, small deformations. Each of this small deformations is located around a point called a *seed* which defines a local area in which the small deformation occurs. The global deformation is a discrete temporal integration of a trajectory of such vector fields. At each time an area of activation is defined (corresponding to an area of gene activation) and a seed in this area generates a local vector field. The activation of seeds, the duration of activation and other parameters are modelled through a probabilistic framework and parameters of their laws are estimated through a maximum-likelihood process. This framework enables to model biological growth and the probabilistic approach is adapted to the consideration of gene activation of growth process [Por08]. However it does not allow to build a vocabulary of deformations that would be used to transform a template shape into different subjects of a population while this is an important issue as stressed in [Gre96, GCK12]. Besides generators of

deformations (seeds) are defined at each time and then are not transported by the global deformation during the integration of the flow, which can lead to lack in interpretation.

2.2.7 Diffeons framework

In [You12] is presented a parametric approach where, on the contrary, generators of deformations are forced to be transported by the flow of diffeomorphisms. In this framework a manifold S of *shapes* is considered, as well as a space of vector fields V that can act infinitesimally on it. From one can define the group G_V of diffeomorphisms that can be written as the final value of a flow of a trajectory of V, and show that G_V acts on S. To each shape m of S is associated a finite family of vector fields $\gamma_i(m)$, called *diffeons*, which defines a subspace V_m of V. Then one only considers trajectories of vector fields $v : t \in [0,1] \mapsto v_t \in V$ such that for each t, v_t can be written $v_t = \sum_i \alpha_i(t)\gamma_i(m_t)$ with $\alpha_i(t) \in \mathbb{R}$ and m_t the transported value of the original shape $m_{t=0}$ by the flow of v. The problem of finding an optimal trajectory of vector fields transporting a source shape m_0 to a target shape m_1 then amounts to an optimal control problem where the cost associated to each vector field v is $|v|_V^2$. In order for this cost to be easily computable in practice, one needs to restrict possible vector fields $\gamma_i(m)$ to a certain set: local translations whose center and covariance matrix are geometrical parameters depending partially on the shape m and transported by the flow. Vectors of these translations are control variables whose trajectory is to be optimized to transport the source shape onto the target shape. This framework defines a method to build diffeomorphisms through constraints depending on shapes, leading to a new metric on the shape space. Two main examples are given in this article. In the first one, shapes of S are approximated by a discretization and this discretization is used to approximate optimal trajectories of vector fields obtained in the non-parametric framework. In the second example diffeons called "pushers" and "pullers" are defined. They generate local translations centered away from the shapes and then tend to "push" or "pull" it. However a generic method to build diffeons adapted to a particular problem still needs to be specified. For instance a method to merge different type of *simple* diffeons into a more *complex* one would be useful. Besides the cost associated to each vector field is given by the squared norm, measuring how "easy" this vector field is to build in V but not how easy it is to built as combination of generators $\gamma_i(m)$: it is not possible here to define a metric more adapted to a given problem.

2.2.8 Conclusion

Thanks to the non-parametrized (and non structured) frameworks presented in Sections 2.2.1, 2.2.2 and 2.2.3, by choosing an appropriate representation of shapes and an appropriate space of vector fields, it is possible to obtain a deformation model allowing good registration of given shapes. However, as emphasized in Section 2.2.4, these frameworks do not allow to impose a particular structure on the generated deformations. This can also lead to a lack of interpretability for these deformations. The parametrized frameworks introduced in Sections 2.2.5, 2.2.6 and 2.2.7 allow partially to overcome this issue. In these models, by choosing a family of generators of vector fields, and constrain-

ing vector fields to be generated by them, one can impose a certain structure on the generated deformations. However none of these frameworks go with a generic methods to define easily generators which would be adapted to the study of particular shapes and simultaneously satisfy properties that are theoretically required (to prove existence of optimal trajectories for instance). We also saw that the question of the evolution of these generators during the integration of the flow is an issue. In frameworks poly-affine (Section 2.2.5) and GRID (Section 2.2.6) it is not possible to choose this evolution. In the last framework of Diffeons (Section 2.2.7), this evolution is more generic as it comes from the infinitesimal action of a shape space and could therefore be modified. However for numerical issue vector fields are constrained to be sums of local translations and then cannot be generically adapted to a given problem. A last important limitations in these frameworks is that there is no possibility to define a cost associated to the generated vector fields which would be *coherent* with the priors that are introduced: this cost is always given by the norm of the vector field in a *fixed* space of vector fields. Then if two choices of generators allow to build the same vector fields, while one choice is more *natural* than the other, their costs will be equal: this cost does not take into account the structure which is imposed to the deformation model.

2.3 Toward a generic modular framework

The object of this thesis is to define a new deformation model allowing to build deformations that can be locally constrained to a certain type of deformations and interpretable. As explained previously, this should come with costs associated to these deformations relevant with respect to these constraints. Similarly to previous frameworks (presented in Sections 2.2.5, 2.2.6 and 2.2.7), deformations are built as final values of a flows of trajectories of vector fields and the incorporation of constraints in the deformation model corresponds to the setting of a family $\mathcal{V} \doteq (v_i)_{i\in I}$ of generators of vector fields. Once this family is setted, the idea is to consider only vector fields that are a linear combination of a finite sub-family of $\mathcal{V}$, *ie* we impose that at each time there exists $(\alpha_k)_{1\leq k\leq N} \in \mathbb{R}^N$ so that the vector field can be written $\sum_k \alpha_k v_{i_k}$. Unlike in the GRID model (Section 2.2.6), we want to consider trajectories v_t of such vector fields such that generators v_{i_k} are *transported* by the flow of v. This transport is realised thanks to an action of vector fields on generators, which can for example be the null action (similarly to the poly-affine framework, see Section 2.2.5) or correspond to an action on shapes that would parametrize generators v_{i_k} similarly to the Diffeons model, see Section 2.2.7. Then if initial generators $v_{i_k}(t = 0)$ are setted, searching the best trajectory of vector fields so that its flow transports an initial shape f_0 as close a possible to a target shape f_1 corresponds to searching the best trajectory of variables α_k so that the corresponding trajectory of vector fields transports f_0 as close as possible to f_1. Variables α_k correspond to *controls* and this matching problem is a *control problem*.

The choice of the family of generators of vector fields $\mathcal{V}$ is a key point of the framework and the idea is to define these generators such that they can capture the variability of interesting geometrical features between shapes. This idea is in the spirit of the approach developed in [PFY+99, PTC00, YPJM01] where authors define medial atoms, named

M-reps, which contain complex local geometrical information. Several types of these medial atoms are defined so that they can describe in a relevant manner the geometry of a given shape: for instance the extremity of a $2D$ shape can be represented thanks to a *corner-end* atom if is a corner, or a *rounded* atom if it is a portion of a circle. Then a net of such medial atoms give a *structured* and easily *interpretable* representation of the shape. Building similar nets of two shapes and comparing parameters of a particular atom allows afterwards to study the variability of corresponding features, separately from other variability. In our approach we adopt a similar point of view but (as emphasized previously) instead of describing the geometry of a shape in its own, we intend to describe *deformations* encoding shape variability. Then we will define deformation atoms which will contain complex local geometrical information about the deformation they generate. In the same way as for M-reps, we want to build a relevant parametrisation of these atoms, so that the representation of deformations we obtain thanks to them can be easily interpreted.

In a first framework which is presented in Section 2.3.1, we constrained vector fields to be sums of local scaling and rotations. We developed a naive and intuitive approach, we made some choices of modelling that seemed intuitively satisfying but led to several limitations. Then we decided to develop a new framework where choices of modelling would be more natural from a theoretical point of view so that it would be more satisfying. In this model vector fields are sums of local translations of different characteristic scales. It is presented in Section 2.3.2 and contains important ideas that will be used in the deformation module framework which introduced in Section 2.3.3 and will be presented in Chapter 4.

2.3.1 Local scalings and rotations: a naive approach

In this section we restrict ourselves to the two-dimension case so we set $d = 2$, but this framework is generalisable to a any dimension d. As explained previously, the strategy is to impose, at each time, the vector field to be a finite sum of generators belonging to a fixed family of vector fields. A first idea is for these generators to be infinitesimal local scalings and local rotations. Let $\Psi : \mathbb{R}^+ \mapsto [0, 1]$ be a smooth local function: we suppose that $\Psi(0) = 1$ and $\Psi(x)$ decreases to 0 when x increases to infinity. We define the local scaling of scale $\sigma \in \mathbb{R}^+$ and centred at c by

$$S(\sigma, c, \cdot) : x \in \mathbb{R}^d \mapsto \Psi\Big(\frac{|x - c|}{\sigma}\Big)(x - c)$$

and the local rotation of scale $\sigma \in \mathbb{R}^+$, centred at c by

$$R(\sigma, c, \cdot) : x \in \mathbb{R}^d \mapsto \Psi\Big(\frac{|x - c|}{\sigma}\Big)\beta \wedge (x - c)$$

where β is orthogonal to the 2D plane. For all c and σ, $S(c, \sigma, \cdot)$ and $R(c, \sigma, \cdot)$ belong to the Banach set $C^2_0(\mathbb{R}^2)$ of vector fields of class C^2 that tend to 0 at infinity. We consider

vector fields w that are a linear combination of such generators, *ie* that can be written

$$v = \sum_{i=1}^{n_s} \alpha_i S(\sigma_i, c_i, \cdot) + \sum_{i=n_s+1}^{n_r+n_s} \alpha_i R(\sigma_i, c_i, \cdot)$$

with $n_s, n_r \in \mathbb{N}^*$, $\alpha_i \in \mathbb{R}$, $\sigma_i \in \mathbb{R}^+$ and $c_i \in \mathbb{R}^d$. We set V this space of vector fields. We now set numbers of local scalings n_s, and of local rotations n_r, and we will study large deformations that can be built thanks to n_s scalings and n_r rotations. For any trajectory of controls α_i, centers c_i and scale σ_i so that the trajectory of vector fields

$$v : t \in [0,1] \mapsto v_t = \sum_{i=1}^{n_s} \alpha_i(t) S(\sigma_i(t), c_i(t), \cdot) + \sum_{i=n_s+1}^{n_r+n_s} \alpha_i(t) R(\sigma_i(t), c_i(t), \cdot)$$

belongs to $L^1([0,1], C_0^1(\mathbb{R}^2))$, the solution of

$$\left\{ \begin{array}{rcl} \phi_{t=0} & = & Id \\ \dot{\phi}_t & = & v_t \circ \phi_t \end{array} \right.$$

exists and is uniquely defined. This defines the process of generation of large deformations from trajectories (regular enough) of variables $(\alpha_i, c_i, \sigma_i)$. For $\theta \doteq (c_i, \sigma_i)$ and controls α_i we set

$$v_{\theta,\alpha} \doteq \sum_{i=1}^{n_s} \alpha_i S(\sigma_i, c_i, \cdot) + \sum_{i=n_s+1}^{n_r+n_s} \alpha_i R(\sigma_i, c_i, \cdot) \tag{2.1}$$

The second step of the construction of the deformation model is the choice of trajectories of geometrical variables $\theta = (c_i, \sigma_i)$ that we will allow. Indeed, they need to evolve in a coherent way with respect to the integrated diffeomorphism. In order to obtain such coherent trajectories, the idea is to set the action $w \cdot \theta$ of vector fields $w \in C_0^2(\mathbb{R}^2)$ on these variables θ and then to consider trajectories of variables such that $\dot{\theta} = v_{\theta,\alpha} \cdot \theta$ for a given trajectory of controls α. The action of vector fields w on centers c is given naturally given by the application of the transformation: $v \cdot c \doteq v(c)$. For scales σ attached to the point c, we will use the action $v \cdot \sigma = \frac{\sigma}{2} div(v)(c)$ which derives from the action of diffeomorphisms $\phi \cdot \sigma = \det(\mathrm{d}\phi_c)^{\frac{1}{2}} \sigma$ (corresponding to the local change of volume induced by ϕ in c). From an initial set of variables $\theta_0 = (c_i^0, \sigma_i^0)$ and trajectories of controls (regular enough) $\alpha : [0,1] \mapsto \mathbb{R}^{n_s+n_r}$, one can build a trajectory of vector fields $t \mapsto v_{\theta_t,\alpha_t}$ with $\theta_{t=0} = \theta_0$ and $\dot{\theta}_t = v_{\theta_t,\alpha_t} \cdot \theta = \Big(v_{\theta_t,\alpha_t} \cdot c_i(t), v_{\theta_t,\alpha_t} \cdot \sigma_i(t) \Big)$. We will now consider such trajectories since their local generators evolve with the flow as required. The last step in the definition of the framework is the association of a cost to each trajectory defined by an initial θ_0 and a trajectory of controls α. This cost will be defined as the integration of a cost $C(\theta, \alpha)$ associated to value of the variable θ and the control α. We want this cost to measure the difference between a vector field v_θ and the null one, taking account of the imposed form the vector field. There is no natural choice here and we make the choice of a cost depending only on the controls: $C(\theta, \alpha) = \frac{1}{2}|\alpha|^2 = \frac{1}{2} \sum_{i=1}^{n_s+n_r} \alpha_i^2$. If $t \mapsto (\theta_t, \alpha_t)$ is such that $\int_0^1 C(\theta, \alpha)$ is finite, θ is absolute continuous and satisfies $\dot{\theta} = v_{\theta,\alpha} \cdot \theta$ then the flow of $v_{\theta,\alpha}$ exists. Reciprocally, if

α is such that $\int_0^1 \frac{1}{2}|\alpha|^2$ is finite, then for all $\theta_{t=0}$, the differential equation $\dot{\theta} = v_{\theta,\alpha} \cdot \theta$ has an absolute continuous solution θ.
In order to define completely the deformation model, we need to specify the local function Ψ. It needs to satisfying several requirements: it needs to be maximal and equal to 1 at 0. Besides, if we consider only one scaling $S(\sigma, c, \cdot)$, it seems natural that the evolution of its scale σ under its own action is similar to the evolution of the distance between the centre c and a point initially at distance σ from c. In other word, if x is such that $|x - c| = \sigma$, then we want that $\frac{\mathrm{d}|x-c|^2}{\mathrm{d}t} = \frac{\mathrm{d}(\sigma^2)}{\mathrm{d}t}$ which leads to $|x - c|^2\Psi(\frac{|x-c|}{\sigma}) = \sigma^2$ and then $\Psi(\frac{|x-c|}{\sigma}) = 1$. This demands then that Ψ is constant equal to 1 on $[0, 1]$ and then decreases to 0. A satisfying function Ψ is then given by:

$$\Psi : u \in \mathbb{R}^+ \mapsto \begin{cases} 1 \text{ if } u \leq 1 \\ 1 - \exp(\frac{-1}{u^2-1}) \text{ if } u > 1 \end{cases}$$

We see that even though this function satisfies all requirements, it seems unnecessary complicated: the action on scales leads to unnatural requirements. Besides this action is not really satisfying when several scalings and rotations are considered. For instance if we consider one scaling of small scale and one rotation of large scale such that the center of the scaling is close to the center of the rotation and needs to generate a shrinking, then the (large) scale of the rotation will decrease while it may not be relevant: the evolution of the scale of a scaling or a rotation depends on the vector fields at its centre while it should take into account the vector field in an area whose size depends on the scale. A last drawback of this framework is that in practice combinations of local scalings and rotations would not be sufficient to study differences between two shapes. Then one should enrich this framework with other generators of vector fields. However, the cost C is not linked to the norm of the generated vector field and then ensuring the existence of geodesics, or even of the flow, might not be possible.

2.3.2 Multi-scale sum of local translations

In order to deal with notions of action and cost that could be more naturally defined we restricted ourselves to points for geometrical parameter, therefore setting as a prior scales of local deformations. We developed a new deformation model as an intermediate model that would enable to prevent theoretical and numerical issues presented in the previous one, before defining a more satisfying model.

We concentrate here on local deformations defined thanks to local translations at fixed scales. In order to ensure theoretical coherence, we consider costs inspired from (but not equal to) the norm of the generated vector in a Hilbert space. However we keep the notion of local deformations acting in areas fixed by a geometrical parameter (the center of the local translation) and we express the cost as a function of geometrical parameters and control variables. Let us define this deformation model more in detail. Let $\sigma_1, \cdots, \sigma_N \in \mathbb{R}_+^*$ be N characteristic scales for the local deformations that we will build. We set for each i, V_i the scalar-gaussian RKHS defined by its kernel $K_i : (x, y) \in$

$(\mathbb{R}^d)^2 \mapsto \exp(\frac{|x-y|^2}{\sigma_i^2})$. We then define $W = \prod_i V_i$ and the mapping

$$\pi : \left\{ \begin{array}{c} W \mapsto V \\ v = (v_1, ..., v_N) \mapsto \sum_i v_i \end{array} \right.$$

where $V \doteq \pi(W)$. We equip V with the following norm: for $v \in V$,

$$|v|_V^2 = \inf\{\sum_i |v_i|_{V_i}^2 \mid \forall i, v_i \in V_i \text{ and } \pi((v_i)_i) = v\}.$$

Proposition 5. *V is a Hilbert space continuously injected in $C_0^\ell(\mathbb{R}^d)$ for all $\ell \geq 1$.*

Proof. This directly results from the fact that spaces V_i are Hilbert spaces continuously injected in $C_0^\ell(\mathbb{R}^d)$ (see [Gla05]). □

As we explained previously, we want to study local deformations generated by sums of local translations acting in areas defined by centres of local translations. Then for each i let us define a number of translations D_i and, for a family of points $e_i = (z_j^i)_{1 \leq j \leq D_i}$, a sub-space of vector fields

$$V_{e_i} = \{v \in V_i \mid \exists \alpha_1^i, ..., \alpha_N^i \in \mathbb{R}^d : v = \sum_j K_i(z_j^i, \cdot)\alpha_j^i\}.$$

For $e = (e_i)_{1 \leq i \leq N} = (z_j^i)_{1 \leq i \leq N, 1 \leq j \leq D_i}$, we can now define the set of vector fields carried by e: $V_e \doteq \pi(W_e)$ with $W_e = \prod_i V_{e_i}$. We suppose that for each i, j, j', $z_j^i \neq z_{j'}^i$, one can check that $\pi_{|W_e}$ is injective. Then space V_e can be equipped with the following inner product: let $v, v' \in V_e$ and let $w = (w_i)_{1 \leq i \leq N}$, $w' = (w_i')_{1 \leq i \leq N} \in W_e$ the unique elements of W_e such that $\pi(w) = v$, $\pi(w') = v'$, we define $\langle v, v' \rangle_{V_e} = \sum_i \langle w_i, w_i' \rangle_{V_i}$.

Remark 2. *The expression of the norm of V_e is natural: if $v = \sum_{i,j} K_i(z_j^i, \cdot)\alpha_j^i \in V_e$,*

$$\begin{array}{rcl} \|v\|_{V_e}^2 = \langle v, v \rangle_{V_e} & = & \sum_i \langle \sum_j K_i(z_j^i, \cdot)\alpha_j^i, \sum_j K_i(z_j^i, \cdot)\alpha_j^i \rangle_{V_i} \\ & = & \sum_i \sum_{j,j'} \alpha_j^{iT} K_i(z_j^i, z_{j'}^i)\alpha_j^i \end{array}$$

Note that in this expression there is no interaction terms between components belonging to different V_{e_i}: in general $\| \cdot \|_{V_e}^2 \neq \| \cdot \|_V^2$.

In the following we will use these notations, for $e = (e_i)_{1 \leq i \leq N} = (z_j^i)_{1 \leq i \leq N, 1 \leq j \leq D_i}$:

- $K_i(e_i, e_i)$ stands for the matrices whose component (j, k) is $K_i(z_j^i, z_k^i)$,
- $K_i(e_i, x)$ stands for the column vector whose jth-component is $K_i(z_j^i, x)$,
- $K_i(x, e_i)$ stands for the row vector whose jth-component is $K_i(x, z_j^i)$.

Proposition 6. *Let $e = (e_i)_{1 \leq i \leq N} = (z_j^i)_{1 \leq i \leq N, 1 \leq j \leq D_i}$ absolute continuous such that for each i, j, j', $z_j^i \neq z_{j'}^i$, then V_e equipped with $\langle \cdot, \cdot \rangle_{V_e}$ is a RKHS and its kernel is*

$$K_e : (x, y) \in (\mathbb{R}^d)^2 \mapsto \sum_i K_i(x, e_i) K_i(e_i, e_i)^{-1} K(e_i, y) \in \mathbb{R}$$

Proof. Let $x \in \mathbb{R}^d$, $\alpha \in \mathbb{R}^d$, $v \in V_e$,

$$K_e(x,\cdot)\alpha = \sum_i \sum_j (K_i(x,e_i)K_i(e_i,e_i)^{-1})_j K(z_j^i,\cdot)\alpha$$

then $K_e(x,\cdot)\alpha$ belongs to V_e. Besides as v belongs to V_e, there exists $(\beta_k^i)_k$ so that

$$v = \sum_i \sum_k K_i(z_k^i,\cdot)\beta_k^i.$$

We set for each i, $K_{e_i} : (w,y) \in (\mathbb{R}^d)^2 \mapsto K_i(w,e_i)K_i(e_i,e_i)^{-1}K(e_i,y) \in \mathbb{R}$, then

$$\begin{array}{rcl} \langle v, K_e(x,\cdot)\alpha\rangle_{V_e} & = & \sum_i \langle \sum_k K_i(z_k^i,\cdot)\beta_k^i, K_{e_i}(x,\cdot)\alpha\rangle_{V_i} \\ & = & \sum_i \langle \sum_k K_i(z_k^i,\cdot)\beta_k^i, K_i(x,e_i)K_i(e_i,e_i)^{-1}K(e_i,\cdot)\alpha\rangle_{V_i} \\ & = & \sum_i \sum_j \sum_k \langle K_i(z_k^i,\cdot)\beta_k^i, (K_i(x,e_i)K_i(e_i,e_i)^{-1})_j K_i(z_j^i,\cdot)\alpha\rangle_{V_i} \\ & = & \sum_i \sum_{j,k} (K_i(x,e_i)K_i(e_i,e_i)^{-1})_j K_i(z_j^i,z_k^i)\langle \beta_k^i,\alpha\rangle_{\mathbb{R}^d} \end{array}$$

Besides for i,k fixed, $\sum_j (K_i(x,e_i)K_i(e_i,e_i)^{-1})_j K_i(z_j^i,z_k^i) = K_i(x,e_i)K_i(e_i,e_i)^{-1}K_i(e_i,z_k^i)$ and as $K_i(e_i,z_k^i)$ is the k-th column of $K_i(e_i,e_i)$, the column vector $K_i(e_i,e_i)^{-1}K_i(e_i,z_k^i)$ is the k-th column of the identity matrix and then $\sum_j (K_i(x,e_i)K_i(e_i,e_i)^{-1})_j K_i(z_j^i,z_k^i) = K_i(x,z_j^i)$. We then deduce that

$$\langle v, K_e(x,\cdot)\alpha\rangle_{V_e} = \sum_i \sum_k K_i(z_j^i,x)\langle \beta_k^i,\alpha\rangle_{\mathbb{R}^d} = \langle v(x),\alpha\rangle_{\mathbb{R}^d}$$

which concludes the proof.

□

The norm that we have just setted on V_e enables to associate a cost $c(e,\alpha)$ to each value of geometrical parameter $e = (z_j^i)_{i,j}$ and controls $\alpha = (\alpha_j^i)_{i,j}$: $c(e,\alpha) = |\sum_{i,j} K_i(z_j^i,\cdot)\alpha_j^i|_{V_e}^2$. We emphasize here that this cost is not equal to the norm of the generated vector field in the constant space $V = \pi(W)$ but in the space V_e that depends on the geometrical variable e.

We can now study large deformations that can be generated by vector fields belonging to spaces V_e. For a purpose of interpretation, we require that the family of points e is transported by the flow. Then we consider trajectories of families of points $e = (z_j^i)_{i,j}$ and controls $\alpha = (\alpha_j^i)_{i,j}$ such that the trajectory of vector field $v_{e,\alpha} \doteq \sum_{i,j} K_i(z_j^i,\cdot)\alpha_j^i$ can be integrated and for each time t, $\dot{e} = v_{e_t,\alpha_t} \cdot e = (v_{e_t,\alpha_t}(z_j^i(t)))_{i,j}$.

Proposition 7. *Let e and α such that, for each t, $\dot{e} = v_{e_t,\alpha_t} \cdot e = (v_{e_t,\alpha_t}(z_j^i(t)))_{i,j}$ and $\sum_{i,j} \int_0^1 |\alpha_j^i(t)|^2 \mathrm{d}t < \infty$ (i.e. $\alpha \in L^2([0,1],\mathbb{R}^{d\sum_i D_i})$, then $\int_0^1 c(e_t,\alpha_t)\mathrm{d}t < \infty$, $t \mapsto v_{e_t,\alpha_t}$ belongs to $L^2([0,1],V)$ and its flow exists.*

Proof. Let e and α such trajectories,

$$\begin{array}{rcl} \int_0^1 c(e_t,\alpha_t)\mathrm{d}t & = & \int_0^1 \sum_i |\sum_{j,j'} K_i(z_j^i, z_{j'}^i)\alpha_j^{iT}\alpha_{j'}^i|\mathrm{d}t \leq \int_0^1 \sum_{i,j,j'} |\alpha_j^{iT}\alpha_{j'}^i|\mathrm{d}t \\ & \leq & \int_0^1 \sum_{i,j,j'} \frac{1}{2}(|\alpha_j^{iT}|^2 + |\alpha_{j'}^i|^2)\mathrm{d}t \leq \int_0^1 \sum_{i,j} \frac{1}{2} \times 2D_i|\alpha_j^i|^2\mathrm{d}t \\ & \leq & (\max D_i)\int_0^1 \sum_{i,j} |\alpha_j^i|^2\mathrm{d}t \leq (\max D_i)\int_0^1 |\alpha|^2\mathrm{d}t \\ & < & \infty \end{array}$$

Besides, for each t, as $v_{e_t,\alpha_t} = \sum_i \sum_j K_i(z_j^i(t),\cdot)\alpha_j^i(t)$, by definition

$$|v_{e_t,\alpha_t}|_V^2 \leq |\sum_i \sum_j K_i(z_j^i(t),\cdot)\alpha_j^i(t)|_{V_i}^2 = c(e_t,\alpha_t).$$

Then $t \mapsto v_{e_t,\alpha_t}$ belongs to $L^2([0,1],V)$ and from the continuous injection of V to $C_0^1(\mathbb{R}^d)$ one obtains the existence of its flow. □

We have setted the deformation model, and we can now study the inexact matching problem. In a first simple case, let us suppose that we want to match an initial set of points e^0 to another one e^1. It corresponds to minimizing the following functional with respect to the trajectory $\alpha \in L^2([0,1],\mathbb{R}^{d\sum_i D_i})$:

$$E(\alpha, e^0, e^1) = \int_0^1 c(e_t,\alpha_t) + \frac{1}{\lambda} D(e_{t=1}, e^1)$$

where $\lambda \in \mathbb{R}_+^*$ is a fixed parameter, D is the Euclidean distance, $e_{t=0} = e^0$ and for each t $\dot{e}_t = v_{e_t,\alpha_t} \cdot e_t$. We can show that minimizing trajectories of $\alpha \in L^2([0,1],\mathbb{R}^{d\sum_i D_i})$ exist, we do not present the proof here as it is in fact a particular case of the general framework which will be presented in Chapters 4 and 5. We can also show the following proposition (similarly this is a particular case of the proof which will be presented in Chapters 4 and 5):

Proposition 8. *Let α be such a minimizing trajectory, and let e be the absolute continuous trajectory defined by $e_{t=0} = e^0$ and for each t, $\dot{e}_t = v_{e_t,\alpha_t} \cdot e_t$. Then there exists a trajectory $\eta : t \in [0,1] \mapsto \mathbb{R}^{d\sum_i D_i}$ such that with*

$$H : (\eta, e, \alpha) \in \mathbb{R}^{d\sum_i D_i} \times \mathbb{R}^{d\sum_i D_i} \times \mathbb{R}^{d\sum_i D_i} \mapsto \langle \eta, v_{e,\alpha}.e\rangle - \frac{1}{2}\|v_{e,\alpha}\|_{V_e}^2$$

$$\begin{cases} \dot{e} = v_{e,\alpha} \cdot e \\ \dot{\eta} = -\frac{\partial H}{\partial e}(\eta, e, \alpha) \\ \frac{\partial H}{\partial \alpha} = 0 \end{cases}$$

From the last equality one can compute optimal α from a value of points e and dual variable (called *momentum*) η: $\forall i$,

$$\alpha^i = \sum_l \eta^l K_i(e^l, e^i) K_i(e^i, e^i)^{-1}$$

with α^i the matrix whose j-th column is α_j^i and η^l the matrix whose k-th column is η_k^l.

Remark 3. *It can be easily shown that if, for each i, $\alpha^i = \sum_l \eta^l K_i(e^l, e^i) K_i(e^i, e^i)^{-1}$ then $v_{e,\alpha} = \sum_{l,k} K_{W_e}(z_k^l, \cdot)\eta_k^l$.*

In practice one does dot want to match points e that generate the deformations, but other shapes f^1, f^2. We will restrict ourselves to the case of landmarks: $f^0 = (f_k^0)_{1\leq k\leq P}$, $f^1 = (f_k^1)_{1\leq k\leq P}$. Then one wants to minimize the following functional with respect to e^0 and α :

$$E(\alpha, e^0, f^0, f^1) = \int_0^1 c(e_t, \alpha_t) + \frac{1}{\lambda} D(f_{t=1}, f^1)$$

where $\lambda \in \mathbb{R}_+^*$ is a fixed parameter, D is the Euclidean distance, $e_{t=0} = e^0$, $f_{t=0} = f^0$ and for each t, $\dot{e}_t = v_{e_t,\alpha_t} \cdot e_t$, $\dot{f}_t = v_{e_t,\alpha_t} \cdot f_t = \Big(v_{e_t,\alpha_t}(f_k(t))\Big)_k$ with $f = (f_k)_{1\leq k\leq P}$. Once again, we can express optimal trajectories in function of a dual variable of the total geometrical variable $m \doteq (e, f)$:

Proposition 9. *Let α be such a minimizing trajectory, and let $m = (e, f)$ be the absolute continuous trajectory defined by $m_{t=0} = (e^0, f^0)$ and for each t, $\dot{m}_t = (v_{e_t,\alpha_t} \cdot e_t, v_{e_t,\alpha_t} \cdot f_t)$. Then there exists a trajectory $\eta : t \in [0, 1] \mapsto \mathbb{R}^{d\sum_i D_i} \times \mathbb{R}^{dP}$ such that with*

$$H : (\eta, m, \alpha) \in \Big(\mathbb{R}^{d\sum_i D_i} \times \mathbb{R}^{dP}\Big) \times \Big(\mathbb{R}^{d\sum_i D_i} \times \mathbb{R}^{dP}\Big) \times \mathbb{R}^{d\sum_i D_i} \mapsto \langle \eta, v_{e,\alpha}.m\rangle - \frac{1}{2}\|v_{e,\alpha}\|_{V_e}^2$$

$$\begin{cases} \dot{m} = v_{e,\alpha} \cdot m \\ \dot{\eta} = -\frac{\partial H}{\partial m}(\eta, m, \alpha) \\ \frac{\partial H}{\partial \alpha} = 0 \end{cases} \tag{2.2}$$

Remark 4. *Note that in this more general case, the dimension of the control α and the momentum η are no longer equal.*

From the last equality one can compute optimal α from a value of points $m = (e, f)$ and dual variable $\eta = (\eta^e, \eta^f)$: for each i,

$$\alpha^i = \sum_l \eta^l K_i(e^l, e^i) K_i(e^i, e^i)^{-1} + \eta^f K_i(f, e^i) K_i(e^i, e^i)^{-1}$$

with η^l the matrix with dual variables of e_l and η^f the matrix with the dual variable of the landmarks f. Then from initial values of e^0 and $\eta = (\eta^e, \eta^f)$, one can integrate Equation (2.2) and obtain a trajectory of controls α and a trajectory of vector fields $v_{e,\alpha}$.

Remark 5. *In the expression of these new trajectories, one can see that even though they do not generate any vector field, points of the shape f have an active momentum which influences the optimal control in the same way as other momenta. This suggests that points of the shape could be considered as silent points (i.e. generating a null vector field) and then incorporated in the deformation model. This is the strategy that will be developed in the generic model that will be presented in Section 5.2.2.*

In this framework, the inexact matching problem amounts to estimating initial values of points e^0 and momentum $\eta = (\eta^e, \eta^f)$ such that the corresponding trajectories of

controls α and points (e, f) minimizes $\int_0^1 c(e_t, \alpha_t) + \frac{1}{\lambda} D(f_{t=1}, f^1)$. In order to enforce local deformations to be more complex than translations, we can force points e^i corresponding to a common scale to be pooled together by groups of $d+1$ points (where d is the dimension of the ambient space) so that locally around such a group of points, the dimension of vector fields that can be generated is $d(d+1)$ which is the dimension of the space of affine deformations in dimension d. This way we can interpret locally the generated vector field thanks to the values of the corresponding controls. However, during the integration of the flow, each point is transported separately and then points do not stay pooled together in case of very large deformations. Here we can see that it would be more relevant to consider one point that would be transported by the flow, and would generate a vector field thanks to $d+1$ points that would depend on this central point. This is the strategy developed in Section 4.2.5. Besides in this framework there is no possibility to force the vector field to be of a particular type (for example local scaling) in a certain area.

2.3.3 Intuition of deformation modules

In order to deal with the limitations of the previous model, the idea is to generalize the construction of spaces V_e. We define a process to generate vector fields thanks to a function ζ which takes in input a geometrical variable (which were previously points z_j^i), a control variable (which were previously controls α_j^i) and returns a vector field. This function enables to equip the space of geometrical variable with a fiber bundle structure where the fiber attached to a particular geometrical variable is the space of all vector fields that can be generated by it thanks to ζ. Trajectories that will be considered are *horizontal trajectories* of this vector bundle, *ie* trajectories of geometrical variables and vector fields (e, v) such that $\dot{e} = v \cdot e$ since for interpretation purpose one wants the geometrical variables to evolve with the trajectory of vector fields. The action of vector fields on these variables needs to be specified. We associate a norm to each of the fibers so that it enables to equip this fiber bundle with a sub-Riemannian structure. This norm can be defined as a cost depending on the geometrical variable and the control. We see now that five elements need to be specified in order for the deformation model to be complete: a space of geometrical variables, a space of controls, a function generating vector fields, an action of vector fields on geometrical variables, and a cost. All these elements will be gathered in a structure that we name *deformation module*. It will be detailed in Chapter 4. One can see that by choosing a particular function ζ to generate vector fields, one force them to be of a certain type and then incorporate prior in the deformation model. In our framework we design a hierarchical approach so that complex priors can be easily merged into a deformation module thanks to a combination process.

2.4 Summary of contributions

In this section will be presented main notions and results of this thesis. Demonstrations and various examples will not be presented in this section, they will be detailed in Chapters 4 and 5.

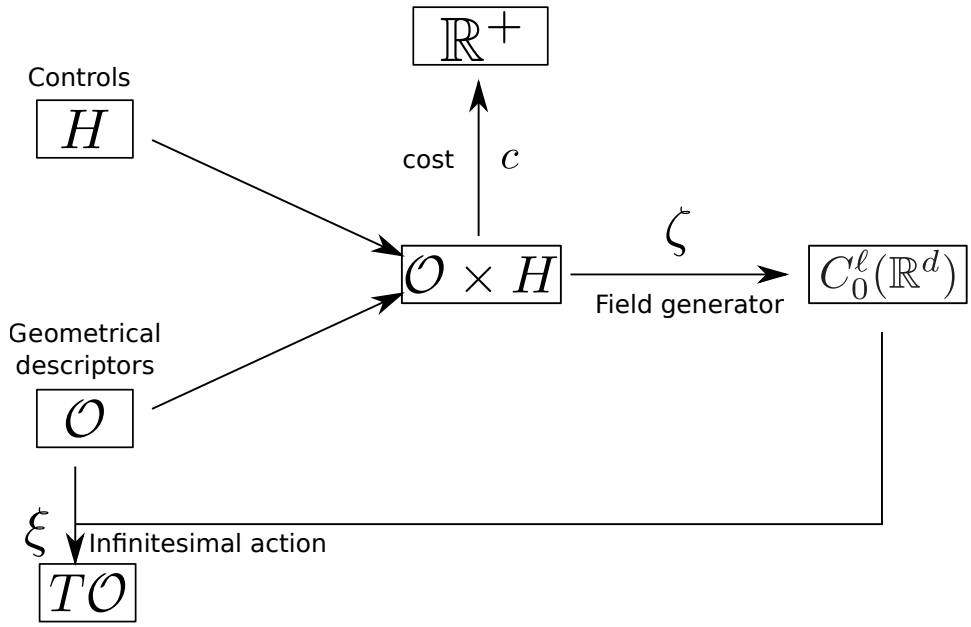


Figure 2.1: Schematic view of a deformation module.

2.4.1 Deformation model

Our framework relies on the new notion of *deformation module*. A deformation module is a structure which is capable of generating vector fields of a particular chosen type and which are parametrized in small dimension. We use here the notion of shape space (of finite dimension) as defined in [Arg14] which will be presented in Section 3.2.

Definition 6. *Let $k, \ell \in \mathbb{N}^*$. We define $M = (\mathcal{O}, H, \zeta, \xi, c)$ as a C^k-**deformation module of order** ℓ with **geometrical descriptors** in $\mathcal{O}$, **controls** in H, **infinitesimal action** ξ, **field generator** ζ and **cost** c, if*

- *$\mathcal{O}$ is a C^k-shape space of $\mathbb{R}^d$ of order ℓ with infinitesimal action $\xi : C_0^\ell(\mathbb{R}^d) \times \mathcal{O} \longrightarrow T\mathcal{O}$,*
- *H is a finite dimensional Euclidean space,*
- *$\zeta : (o, h) \in \mathcal{O} \times H \to (o, \zeta_o(h)) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d)$ is continuous, with $h \mapsto \zeta_o(h)$ linear and $o \mapsto \zeta_o$ of class C^k,*
- *$c : (o, h) \in \mathcal{O} \times H \to c_o(h) \in \mathbb{R}^+$ is a continuous mapping such that $o \mapsto c_o$ is smooth and for all $o \in \mathcal{O}$, $h \mapsto c_o(h)$ is a positive quadratic form on H, thus defining smooth metric on $\mathcal{O} \times H$.*

Various examples of deformation modules will be presented in Section 4.2. A schematic view of a deformation module is presented in Figure 2.1. Vector fields generated by a deformation module are parametrized by two variables: geometrical descriptors and controls. Geometrical descriptors code for the geometry (for instance the location)

of the generated vector field. Controls code for the manner (for instance the strength) with which the generated vector field is used.

Note that geometrical descriptors belong to a shape space and then will be called *shapes*. We will also use the name *shape* to refer to data that we want to study. Even though data and geometrical descriptors satisfy the notion of shapes, in general they are not the same object. However, given a data shape space $\mathcal{F}$ and its infinitesimal action $\xi^{\mathcal{F}}$, we can define a deformation module called *Silent deformation module induced by* $\mathcal{F}$. This deformation module is called silent because the vector fields which it generates are always null. However its geometrical descriptor can feel vector fields thanks to the infinitesimal action $\xi^{\mathcal{F}}$. This deformation module is defined by a space of geometrical descriptor equal to $\mathcal{F}$, a null space of control, a null field generator, an infinitesimal action equal to $\xi^{\mathcal{F}}$ and a null cost. The interest of introducing this deformation module will be briefly presented in Section 2.4.3.2 and in detail in Section 5.

We will consider deformation modules which satisfy the following condition, ensuring that their costs are related (but not necessary equal) to the norm of the generated vector field.

Definition 7. *Let $M = (\mathcal{O}, H, \zeta, \xi, c)$ be a C^k-deformation module of order ℓ. We say that M satisfies the **Uniform Embedding Condition (UEC)** if there exists a Hilbert space of vector fields V continuously embedded in $C_0^{\ell+k}(\mathbb{R}^d)$ and a constant $C > 0$ such that for all $o \in \mathcal{O}$ and for all $h \in H$, $\zeta_o(h) \in V$ and*

$$|\zeta_o(h)|_V^2 \leq C c_o(h).$$

Deformation modules enable to incorporate constraints in the deformation model. Indeed if one fixes a deformation module, then one can consider only vector fields that can be generated by the field generator of this deformation module. This is a first step in the introduction of constraints. Constraints that one wants to incorporate can correspond to an additional knowledge that one would have, or to a particular point of view from which one would want to study a population of shapes. However in practice these constraints are complex ones, and then defining the appropriate deformation module is not straightforward. In practice it is easy to define deformation modules corresponding to *simple* constraints. Then we define the combination of deformation modules in order to build deformation modules adapted to complex constraints.

Definition 8. *Let $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, be L C^k-deformation modules of order ℓ. We define the **compound module** of modules M^l by $\mathcal{C}(M^l, l = 1 \cdots L) = (\mathcal{O}, H, \zeta, \xi, c)$ where $\mathcal{O} \doteq \prod_l \mathcal{O}^l$, $H \doteq \prod_l H^l$ and for $o = (o^l)_l \in \mathcal{O}$, $\zeta_o : h = (h^l) \in H \mapsto \sum_l \zeta^l_{o^l}(h^l)$, $\xi_o : v \in C_0^\ell(\mathbb{R}^d) \mapsto (\xi^l_{o^l}(v))_l \in T_o\mathcal{O}$ and $c_o : h = (h^l) \in H \mapsto \sum_l c^l_{o^l}(h^l)$.*

See Figure 2.2 for a schematic view of the combination of three deformation modules. A key point of our framework is the following stability result under combination:

Proposition 10. *If $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, are C^k-deformation modules of order ℓ, then $\mathcal{C}(M^l, l = 1 \cdots L)$ is a C^k-deformation module of order ℓ. Furthermore, if each M^l satisfies UEC, then $\mathcal{C}(M^l, l = 1 \cdots L)$ also satisfies UEC.*

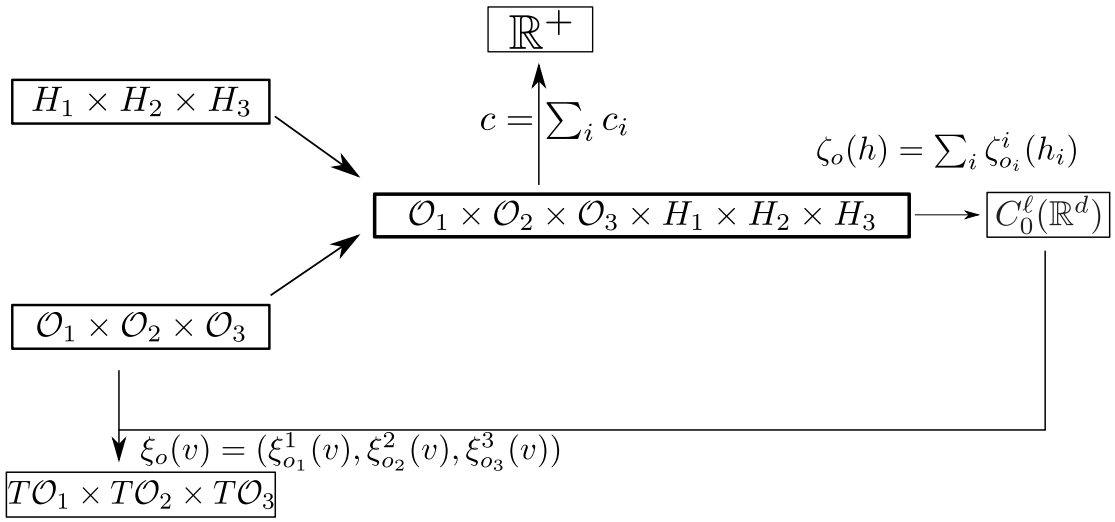


Figure 2.2: Schematic view of a combination of three deformation modules.

Then in order to define a deformation module adapted to a complex constraints, one only has to define this complex constraints as the combination of different simpler ones and to combine the corresponding deformation modules. We will now present how we can build large deformations thanks to this modular framework. Let us set a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfying UEC. We will consider trajectories of vector fields $v : t \in [0,1] \mapsto v_t \in V$ that are modular, meaning that at each time t one can write $v_t = \zeta_{o_t}(h_t)$ with $(o_t, h_t) \in \mathcal{O} \times H$. During the integration of the trajectory we want the geometrical descriptor of the module to be transported by the flow and therefore, denoting φ^v the flow of v (see Proposition 15), that v_t belongs to $\zeta_{o_t}(H)$, with $o_t = \varphi^v_t(o_{t=0})$.

Definition 9. *Let $a, b \in \mathcal{O}$. We denote $\Omega_{a,b}$ the set of mesurable curves $t \mapsto (o_t, h_t) \in \mathcal{O} \times H$ where o_t is absolutely continuous (a.c.), starting from a and ending at b, such that, for almost every $t \in [0,1]$, $\dot{o}_t = \xi_{o_t}(v_t)$, where $v_t \doteq \zeta_{o_t}(h_t)$, and*

$$E(o,h) \doteq \int_0^1 c_{o_t}(h_t)dt < \infty.$$

*The quantity $E(o,h)$ is called the **energy** of (o,h) and $\Omega_{a,b}$ is the set of **controlled paths of finite energy** starting at a and ending at b.*

If UEC is satisfied, we can build large deformations from these trajectories :

Proposition 11. *Let us suppose that M satisfies UEC. Let $(o,h) \in \Omega_{a,b}$ and for each t, $v_t = \zeta_{o_t}(h_t)$. Then $v \in L^2([0,1], V) \subset L^1$, the flow φ^v exists, $h \in L^2([0,1], H)$ and for each $t \in [0,1]$, $o_t = \phi^v_t.o_0$. The final diffeomorphism $\varphi^v_{t=1}$ is called a **modular large deformation generated by** a.*

The energy associated to a trajectory $(o,h) \in \Omega_{a,b}$ is $E(o,h) = \int_0^1 c_o(h)$. Note that this energy does not directly depend on the norm of the generated vector field $\zeta_o(h)$ at

each time. It depends on the geometrical descriptor o and on the way the vector field can be generated by it. Then this energy really corresponds to a *modular information*.

2.4.2 Sub-Riemannian setting

Let us set $M = (\mathcal{O}, H, \zeta, \xi, c)$ a C^k-deformation module of order ℓ satisfying UEC. We will now present how the modular large deformations built previously can enable to equip the shape space of geometrical descriptors $\mathcal{O}$ with an adapted distance. We define the following morphism of bundles:

$$\rho : (o, h) \in \mathcal{O} \times H \mapsto (o, \xi_o \circ \zeta_o(h)) \in T\mathcal{O}$$

This bundle defines at each geometrical descriptor $o \in \mathcal{O}$, the subspace $\rho_o(H) \subset T_o\mathcal{O}$ of all speeds of o that can be generated by the action of o on itself. Moreover, the cost c induces a smooth Riemannian metric g on the vector bundle $\mathcal{O} \times H$. Then $(\mathcal{O} \times H, g, \rho)$ defines a sub-Riemannian structure on $\mathcal{O}$. This structure comes with a sub-Riemannian distance D on $\mathcal{O}$ which is linked to the cost thanks to the following Proposition.

Proposition 12. *Let $a, b \in \mathcal{O}$ such that $D(a, b)$ is finite. Then*

$$D(a, b)^2 = \inf\{\int_0^1 c_o(h) \mid h \in L^2([0, 1], H), \dot{o} = \rho_o(h), o_{t=0} = a, o_{t=1} = b\}$$

Besides, under some mild assumptions, as M satisfies the UEC we obtain the following result:

Theorem 3. *If $\Omega_{a,b}$ is non-empty, the energy E reaches its minimum on $\Omega_{a,b}$.*

2.4.3 Modular comparison of shapes

We use this sub-Riemannian distance to measure differences between two shapes.

2.4.3.1 Matching problem

Let us set $M = (\mathcal{O}, H, \zeta, \xi, c)$ a C^k-deformation module of order ℓ satisfying UEC. We first suppose that we want to compare two shapes a and b belonging to $\mathcal{O}$. If the distance $D(a, b)$ is finite, we can study differences between them by searching *the best* modular large deformation transforming a into b. Then we would want to minimize $\int_0^1 c_o(h)$ amongst trajectories (o, h) of $\Omega_{a,b}$. However in practice shapes are noisy and then we perform inexact matching. It corresponds to minimizing

$$J_{a,b}(h) = \frac{1}{\lambda^2}\mu(o_{t=1}, b) + \int_0^1 c_{o_t}(h_t)dt \tag{2.3}$$

where $\lambda \in \mathbb{R}_+^*$, $o_{t=0} = a$ and $\dot{o} = \xi_o \circ \zeta_o(h)$. The first term of this sum is called the **Data term** while the second one is the **Regularity term**. The following result enables to characterize optimal control trajectories:

Theorem 4. *We recall that $M = (\mathcal{O}, H, \zeta, \xi, c)$ is a C^k-deformation module of order C^ℓ satisfying UEC with $k, l \geq 2$. We suppose that μ is C^1. If $h \in L^2([0,1], H)$ minimizes functional $J_{a,b}$ then, with $o : [0,1] \mapsto \mathcal{O}$ starting at a and satisfying $\dot{o} = \xi_o \circ \zeta_o(h)$, there exists a path $\eta : t \in [0,1] \longrightarrow \eta_t \in T^*_{o_t}\mathcal{O}$ such that with*

$$\mathcal{H} : (o, \eta, h) \in T^*\mathcal{O} \times H \mapsto \Big(\eta | \xi_o(\zeta_o(h))\Big) - \frac{1}{2} c_o(h)$$

*the **Hamiltonian** of the system, $\eta_{t=1} = -\frac{1}{\lambda^2}\partial_1\mu(o_{t=1}, b)$ and (in a local chart)*

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \xi_o \circ \zeta_o(h) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o} \\ \frac{\partial \mathcal{H}}{\partial h} & = & 0 \end{array} \right. \tag{2.4}$$

Remark 6. *As c_o is positive definite, there exists an invertible symmetric operator $C : o \in \mathcal{O} \mapsto C_o \in L(H, H^*)$ such that for all $(o, h) \in \mathcal{O} \times H$, $c_o(h) = (C_o h | h)$. Then the third equality in (2.4) allows to compute h: $h = C_o^{-1}\rho_o^*\eta$ with $\rho_o = \xi_o \circ \zeta_o$ and $\rho_o^* : T_o^*\mathcal{O} \mapsto H^*$ such that $(\eta | \rho_o(h))_{T_o^*\mathcal{O}} = (\rho_o^*\eta | h)_{H^*}$.*

To each initial value of initial momentum $\eta_{t=0} \in T_a^*\mathcal{O}$ can be associated a trajectory of control by integrating Equations (2.4). Then in order to compute the minimum of $J_{a,b}$ we use the *geodesic shooting* approach: we estimate (thanks to a gradient descent) an initial momentum $\eta_{t=0} \in T_a^*\mathcal{O}$ such that the corresponding trajectory of controls minimizes $J_{a,b}$.

2.4.3.2 The matching problem in practice

Unlike what we just presented, in general one wants to match two shapes f_0 and f_1 belonging to a common shape space $\mathcal{F}$ thanks to a user-defined deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$. However this new problem amounts to the previous one. Indeed let us build $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ the *silent deformation module induced by shape space* $\mathcal{F}$. We recall that it is defined as follow: its space of geometrical descriptor is $\mathcal{O}^2 \doteq \mathcal{F}$, its space of control is $H^2 \doteq \{0\}$ (null vector space), its field generator is $\zeta^2 : (o, h) \in \mathcal{O}^2 \times H^2 \mapsto 0$, its infinitesimal action is $\xi^2 \doteq \xi^{\mathcal{F}}$ the infinitesimal action given by $\mathcal{F}$ and the cost is $c^2 : (o, h) \in \mathcal{O}^2 \times H^2 \mapsto 0$. Then we can consider $M \doteq C(M^1, M^2) \doteq (\mathcal{O}, H\zeta, \xi, c)$ the combination of M^1 and M^2. Vector fields that can be generated by this new deformation module, and their associated costs, are the same as those generated by M^1 since by definition for $o = (o^1, f) \in \mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$, $h = (h^1, 0) \in H = H^1 \times H^2$, the generated vector field is $\zeta_o(h) = \zeta^1_{o^1}(h^1) + \zeta^{\mathcal{F}}_f(0) = \zeta^1_{o^1}(h^1)$ and the cost is $c_o(h) = c^1_{o^1}(h^1) + c^2_f(0) = c^1_{o^1}(h^1)$. However optimal control trajectories will not be the same as those generated by deformation module M^1 as here we keep track of their action on both the $\mathcal{O}^1$-component and the $\mathcal{F}$-component thanks to the compound infinitesimal action $\xi : (o, v) \in \mathcal{O} \times C^\ell_0(\mathbb{R}^d) \mapsto (\xi^1_{o^1}(v), \xi^{\mathcal{F}}_f(v))$ with $o = (o^1, f)$. An example of the difference between these optimal trajectories is studied in Section 5.4.1.2.

Thanks to this combination, matching f_0 onto f_1 thanks to deformation module M^1

amounts to solve the matching problem presented previously for M with a data attachment term $\mu(o_{t=1}, b)$ only depending on the $\mathcal{F}$-component (as we do not know the target value for the $\mathcal{O}^1$-component). For a fixed initial value of $o_0^1 \in \mathcal{O}^1$, optimal trajectories of controls (*i.e.* minimizing Equation 2.3) are then parametrized by an initial momentum $\eta \in T^*_{(o^1,f_0)}\mathcal{O}$. Here one needs to optimize both the initial value $o_0^1 \in \mathcal{O}^1$ and the initial momentum $\eta \in T^*_{(o^1,f_0)}\mathcal{O}$ so that the corresponding trajectory of controls minimizes

$$J(h) = \int_0^1 c_o(h) + \mu(f(t=1), f_1)$$

where $o = (o^1, f)$ starts at (o_0^1, f_0) and satisfies $\dot{o} = \xi_o \circ \zeta_o(h) = (\xi^1_{o^1} \circ \zeta^1_{o^1}(h^1), \xi^{\mathcal{F}}_f \circ \zeta^1_{o^1}(h^1))$ with $h = (h^1, 0) \in H = H^1 \times \{0\}$.

Note that here the initial momentum parametrizing optimal trajectories has two component: one in $T^*_{o^1}\mathcal{O}^1$ and one in $T_{f_0}\mathcal{F}$. Then optimal trajectories (for fixed value of initial geometrical descriptor) are parametrized in dimension superior than the dimension of shapes f_i. In Section 5.4 is presented a framework in which we consider a subspace of optimal trajectories, parametrized in small dimension.

2.4.4 Modular analysis of shape variability

Here we explain how this modular framework allows to study the variability amongst a population of shapes. Let $f^1_{target}, \cdots, f^P_{targets}$ be P shapes of a common shape space $\mathcal{F}$, and let us study this population of shapes thanks to a user-defined deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$. As previously we first build the silent deformation module induced by $\mathcal{F}$ and then combine it with $\tilde{M}$. We obtain a compound deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ whose generated vector fields are the same as these of $\tilde{M}$ but that enables to keep track of the way shapes of $\mathcal{F}$ are transported by such vector fields. In this framework the computation of an atlas of targets f^k_{target} thanks to $\tilde{M}$ amounts to minimizing

$$J : (o_{temp}, h^1, \cdots, h^P) \in \mathcal{O} \times (L^2([0,1], H))^P \mapsto \frac{1}{\sigma^2} \sum_k \mu(f^k_{t=1}, f^k_{target}) + \int_0^1 c_{o^k}(h^k)$$

with for each k, $o^k_{t=0} = o_{temp}$ and $\dot{o}^k = \xi_{o^k} \circ \zeta_{o^k}(h^k)$. Note that the initial value of geometrical descriptor o_{temp} is common to all subjects but that the trajectory o^k is specific to each subject as it is transported thanks to the trajectory of control h^k. As in the matching case, optimal trajectories of controls can be parametrized by initial momenta attached to o_{temp}. Then computing an atlas corresponds to estimate one *template* $o_{temp} \in \mathcal{O}$ and P initial momenta $\eta^k_0 \in T^*_{o_{temp}}\mathcal{O}$ so that the corresponding trajectories of controls minimize J. We emphasize here that the template o_{temp} belongs to $\mathcal{O} = \tilde{\mathcal{O}} \times \mathcal{F}$. Its $\mathcal{F}$-component intuitively corresponds to the idea of mean shape while the $\tilde{\mathcal{O}}$-component corresponds to a geometrical characterization of the variability amongst this population of shapes.

2.4.5 A "plug-and-play" algorithm

The algorithm enabling to compute matching and atlases of shapes will be presented in Chapter 6. We use an object-oriented implementation as our framework is very well adapted to it. An abstract class called ABSTRACTMODULE defines deformation modules and all kinds of base-module inherit from this class. Methods of this class implement functions ζ, ξ and c defining the deformation module. Then a meta class called MODULARDIFFEO has as attribute a list of ABSTRACTMODULE. Its methods implement the simple combination rules presented in Section 2.4.1. Thanks to this class one can compute geodesic trajectories. The important idea is that this class uses only *abstract methods* of class ABSTRACTMODULE and is implemented only once. Then if one wants to use a new type of deformation module, one only has to define its functions ζ, ξ and c and then it is directly incorporated in the global deformation model. In practice a large class of base-modules are implemented and we only need to choose which ones we want to use to study a particular problem. If we have a particular prior knowledge on the population of shapes that we want to study, we can use a combination of the corresponding base-deformation modules. On the opposite, if we do not have such a prior knowledge, it is possible to use for example a combination of deformation modules generating local translations, scalings and rotations at different scales (an example will be given in Section 7.4). We integrated this algorithm in the software Deformetrica [DPC+14], so that the data attachment term $\mu(f_k(t=1), f_k^{target})$, and its derivatives, can be automatically computed for a large collection of shapes: point clouds, curve and surface meshes in 2D and 3D. In Chapter 7 we present different problems of matching and atlas computations that we have studied thanks to this modular algorithm.

2.5 Notation

Throughout this thesis we will use the following notations:

- $C_0^\ell(\mathbb{R}^d)$: space of vector fields of class C^ℓ on $\mathbb{R}^d$ whose derivatives of order less than or equal to ℓ converge to zero at infinity (see 3.2).
- $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$: space of $C^\ell-$diffeomorphisms of $\mathbb{R}^d$ that converge to identity at infinity (see 3.2).
- A^{-T}: transpose of the inverse of matrix A.
- K_H: Reproducing kernel of a RKHS H (see 3.1).
- K_σ: scalar kernel of the scalar Gaussian RKHS of scale $\sigma > 0$ (see 3.1).
- X^*: dual of a vector space X.
- $(\delta|x)_{X^*}$: application of $\delta \in X^*$ to $x \in X$ (for X a vector space).
- $\langle \cdot, \cdot \rangle_X$: inner product of vector space X.
- φ^v: flow of v (see 3.2).
- $\underline{X}$: implementation in the algorithm of the mathematical notion X (Chapter 6).

Chapter 3

Mathematical background

Contents

3.1 Reproducing Kernel Hilbert Space

We present here the notion of Reproducing Kernel Hilbert Space [Aro50] that we will use in this thesis. We set an integer $d > 0$.

3.1.1 Definition

Definition 10. *Let $(H, \langle \cdot, \cdot \rangle_H)$ be a Hilbert space of functions $\mathbb{R}^d \longrightarrow \mathbb{R}^d$. We say that H is a Reproducing Kernel Hilbert Space (RKHS) if for each $x \in \mathbb{R}^d$, $\delta_x : f \in H \mapsto f(x) \in \mathbb{R}^d$ is a continuous function.*

Let us set a RKHS $(H, \langle \cdot, \cdot \rangle_H)$. For $x \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^d$ we denote the function $\delta_x^\alpha : f \in H \mapsto \langle f(x), \alpha \rangle_{\mathbb{R}^d}$. Functions δ_x^α are continuous linear forms and then belong to H^*. Then, using Riesz representation theorem we can define the **reproducing kernel** of H:

Definition 11. *The **reproducing kernel** K_H of a RKHS H is the unique function $K_H : H^* \longrightarrow H$ such that for all $h \in H^*$, $\forall f \in H$,*

$$(h|f)_{H^*} = \langle K_H h, f \rangle_H.$$

For $x, \alpha \in \mathbb{R}^d$, one gets $K_H \delta_x^\alpha \in H$ so that for all f in H, $\langle K_H \delta_x^\alpha, f \rangle_H = \langle f(x), \alpha \rangle_{\mathbb{R}^d}$. Then for x and y in $\mathbb{R}^d$, one can define the matrix $K(x, y)$ of $\mathbb{R}^{d \times d}$ so that for all α and β in $\mathbb{R}^d$,

$$\alpha^T K(x, y) \beta = \langle K_H \delta_x^\alpha, K_H \delta_y^\beta \rangle_H.$$

Remark 7. *With a slight abuse of notation, we will identify the operator K_H and the function $K : (x, y) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto K(x, y) \in \mathbb{R}^{d \times d}$.*

Proposition 13. *For all x, y in $\mathbb{R}^d$, $K(x, y)^T = K(y, x)$ and for all finite families $(x_i)_{1 \leq i \leq n}$, $(\alpha_i)_{1 \leq i \leq n}$ of $\mathbb{R}^d$, $\sum_{1 \leq i,j \leq n} \alpha_i^T K(x_i, x_j) \alpha_j \geq 0$.*

3.1.2 Building a RKHS

In the previous section we saw that from a RKHS H, could be built a unique kernel K satisfying Proposition 13. The reverse is also true:

Proposition 14. *Let $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^{d \times d}$ such that for all x, y in $\mathbb{R}^d$, $K(x, y)^T = K(y, x)$ and for all finite families $(x_i)_{1 \leq i \leq n}$, $(\alpha_i)_{1 \leq i \leq n}$ of $\mathbb{R}^d$, $\sum_{1 \leq i,j \leq n} \alpha_i^T K(x_i, x_j) \alpha_j \geq 0$. We say that K is a **positive kernel**. Then there exists a unique RKHS H such that K is its kernel. Besides $\{v = \sum_{i=1}^N K(x, \cdot)\alpha_i | N > 0, x_i \in \mathbb{R}^d, \alpha_i \in \mathbb{R}^d\}$ is a dense subspace of H.*

Then in order to build a RKHS, one only needs to define a positive kernel. In the case where for all values of x and y in $\mathbb{R}^d$ $K(x, y) = k(x, y) I_d$ with $k(x, y) \in \mathbb{R}$ and I_d the Identity matrix, we say that the kernel is **scalar** and we identify the kernel with k. In this thesis we will consider **Gaussian scalar kernels**:

Definition 12. *We say that H is the **scalar Gaussian RKHS of scale** $\sigma \in \mathbb{R}_+^*$ if it is built thanks to the scalar positive kernel $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto \exp -\frac{|x-y|^2}{\sigma^2}$ with $|\cdot|$ the Euclidean norm in $\mathbb{R}^d$.*

Remark 8. *In the case of a scalar Gaussian RKHS, vector fields $K(x, \cdot)\alpha$ (with x and α in $\mathbb{R}^d$) are local translations and then thanks to Proposition 13 we see that the RKHS is generated thanks to sums of local translations.*

3.2 Shape spaces

We present here definitions and propositions related to the notion of shape spaces, they are all proved in [Arg14].

3.2.1 Trajectory of diffeomorphisms

Let d and ℓ be non-zero integers, we define $C_0^\ell(\mathbb{R}^d)$ the space of vector fields of class C^ℓ on $\mathbb{R}^d$ whose derivatives of order less than or equal to ℓ converge to zero at infinity and we equip it with norm $|v|_\ell = \sup\{|\frac{\partial^{\ell_1+\cdots+\ell_d} v(x)}{\partial x_1^{\ell_1} \cdots x_d^{\ell_d}}| \mid x \in \mathbb{R}^d, (\ell_1, \cdots, \ell_d) \in \mathbb{N}^d, \ell_1 + \cdot + \ell_d \leq \ell\}$ such that it is a Banach space. We define $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ the space of $C^\ell-$diffeomorphisms of $\mathbb{R}^d$ that converge to identity at infinity. It is an open set of the affine Banach space $Id + C_0^\ell(\mathbb{R}^d)$ and as such equipped with a natural smooth differential structure. We will consider particular trajectories of $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ defined as flow of particular trajectories of $C_0^\ell(\mathbb{R}^d)$.

Proposition 15. *Let v be an element of $L^1([0,1], C_0^\ell(\mathbb{R}^d))$, ie a time-dependent vector field such that $t \in [0,1] \mapsto |v(t)|_\ell$ is integrable. Then there exists a unique absolutely continuous solution φ^v, called **the flow** of v, to the system*

$$\begin{cases} \dot{\varphi}^v(t) &= v(t) \circ \varphi^v(t) \\ \varphi(0) &= Id \end{cases}$$

where $\varphi^v(t) \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$ for any $t \in [0,1]$.

This proposition allows us to consider flows of time-dependent vector fields, where "time" t refers here to a variable of integration. As we are interested in how these flows can deform a certain shape, we need to precise what is a shape, and how a diffeomorphism of $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ can act on it.

3.2.2 Shape spaces

Definition 13. *Let $\mathcal{O}$ be an open space of a Banach space and let $k \in \mathbb{N}^*$. Assume that the group $\mathrm{Diff}_0^\ell(\mathbb{R}^d)$ continuously acts on $\mathcal{O}$, according to an action*

$$\begin{array}{rcl} \mathrm{Diff}_0^\ell(\mathbb{R}^d) \times \mathcal{O} & \to & \mathcal{O} \\ (\varphi, o) & \mapsto & \varphi \cdot o \end{array} \tag{3.1}$$

*We say that $\mathcal{O}$ is a C^k-**shape space of order** ℓ on $\mathbb{R}^d$ if the following conditions are satisfied:*

1. *for each $o \in \mathcal{O}$, $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto \phi \cdot o$ is Lipschitz with respect to the norm $|\cdot|_\ell$ and is differentiable at $Id_{\mathbb{R}^d}$. This differential is denoted ξ_o and is called the **infinitesimal action** of $C_0^\ell(\mathbb{R}^d)$.*
2. *The mapping $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto \xi_o v$ is continuous and its restriction to $\mathcal{O} \times C_0^{\ell+k}(\mathbb{R}^d)$ is of class C^k.*

*An element o of $\mathcal{O}$ is called a **shape**, and $\mathbb{R}^d$ will be referred to as the **ambient space**.*

The simplest example of shape space is the one of landmarks where shapes are given as a collection of a fixed number of points: $\mathcal{O} = \{o = (x_1, \cdots, x_n) \in (\mathbb{R}^d)^n \mid x_i \neq x_j \texttt{ if } i \neq j\}$. The action of $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$ on $o = (x_1, \cdots, x_n) \in \mathcal{O}$ is given by $\phi \cdot o = \Big(\phi(x_1), \cdots, \phi(x_n)\Big)$ (application to each point), and then the infinitesimal action of $v \in C_0^\ell(\mathbb{R}^d)$ on $o = (x_1, \cdots, x_n) \in \mathcal{O}$ is $\xi_o(v) = \Big(v(x_1), \cdots, v(x_n)\Big)$, namely the velocity of the trajectories of shape points. These actions make $\mathcal{O}$ a shape space of order ℓ for any $\ell \geq 1$.

Remark 9. *If $\mathcal{O}_1$ and $\mathcal{O}_2$ are two shape spaces of order ℓ on $\mathbb{R}^d$, then $\mathcal{O} = \mathcal{O}_1 \times \mathcal{O}_2$ is also a shape space of order ℓ on $\mathbb{R}^d$.*

The following proposition shows that a shape transported by a flow of diffeomorphisms (see Proposition 15) satisfies a differential equation.

Proposition 16. *For every $o \in \mathcal{O}$, the mapping $\phi \mapsto \phi \cdot o$ is of class C^1. Moreover, if $v \in L^1([0,1], C_0^\ell(\mathbb{R}^d))$, for $a \in \mathcal{O}$, the curve $o : t \in [0,1] \mapsto o(t) = \varphi^v(t) \cdot a$ is absolutely continuous and satisfies for almost every $t \in [0,1]$, $\dot{o} = \xi_o(v)$.*

3.2.3 Geodesics on a shape space

Let us set $\mathcal{O}$ a C^k-shape space of order ℓ on $\mathbb{R}^d$, ξ its infinitesimal action, V a RKHS (see Section 3.1) of vector fields at least $C_0^{\ell+1}$ and K its kernel.

Proposition 17. *[ATTY15] Let $a \in \mathcal{O}$, let $g : \mathcal{O} \longrightarrow \mathbb{R}$ a C^1 function. For $v \in L^2([0,1], V)$ we define*

$$J(v) = \int_0^1 |v(t)|_V^2 \mathrm{d}t + g(o(1))$$

where o is the unique absolutely continuous trajectory of $\mathcal{O}$ such that $o(t=0) = a$ and for almost every $t \in [0,1]$, $\dot{o}(t) = \xi_o(v(t))$. If $v \in L^2([0,1], V)$ minimizes J then there exists a measurable function $\eta : [0,1] \mapsto T^*_{o(t)}\mathcal{O}$ such that $v = K\xi_o^*(\eta)$.

This proposition shows that the nature of optimal vector fields is determined by the nature of the shape.

3.3 Currents and varifolds

We will present in this section particular shape spaces. Until we considered only geometrical shapes that were landmarks, therefore with correspondence of points. However in general we do not know corresponding points between shapes that we consider: they are curves or surfaces (without a canonical parametrization).

Let us set an integer d of $\mathbb{N}^*$ (in general d equals 2 or 3). In this thesis we will only consider curves when the ambient space is of dimension 2, and surfaces when the ambient space is of dimension 3 so we will present here only these particular cases. The shape space that we consider is then the shape space of sub-manifolds of dimension $d-1$ and the action of diffeomorphisms is given by the application of them to each point of the sub-manifold. We need a distance which enables to quantify how close are two sub-manifolds in order to measure how well a given diffeomorphism transports one shape close to an other. In order to do so the idea is to embed this shape space in a normed vector space as the dual of a norm space. We briefly present here two possible frameworks: currents where sub-manifolds are supposed to be oriented, and varifolds where they are not. These two frameworks are used in numerical experiments of Chapter 7.

3.3.1 Currents

We present here the notion of currents based on the work of [Gla05].

3.3.1.1 Shape space of currents

We will denote $\Lambda^{d-1}\mathbb{R}^d$ the space of $d-1$-forms on $\mathbb{R}^d$, *i.e.* the space of multi-linear and alternating maps $(\mathbb{R}^d)^{d-1} \longrightarrow \mathbb{R}$. This space is equipped with the following euclidean norm: for $\nu \in \Lambda^{d-1}\mathbb{R}^d$, we set $|\nu| = \sup\{|\nu(\alpha_1, \cdots, \alpha_{d-1})| \text{ for } |\alpha_i| \leq 1\}$. We also define $C_0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ the set of continuous $d-1$-differential forms (*i.e.* applications $\mathbb{R}^d \mapsto \Lambda^{d-1}\mathbb{R}^d$) which tends to zero at infinity. We equip it with the norm $|\cdot| : w \in C^0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d) \mapsto \sup_{x\in\mathbb{R}^d} |w(x)|$ so that is a complete space. We now define its dual $M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$, it is a normed space. Elements of $M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ will be called **currents**. The action of diffeomorphisms on currents is given thanks to the **pull back of differential forms**:

Definition 14. *[Gla05] Let $\phi \in \mathrm{Diff}^\ell_0(\mathbb{R}^d)$, and let $w \in C^0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$, the **pull back** of w by ϕ is $\phi^\# w \in C^0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ defined by*

$$\phi^\# w : (x, \alpha_1, \cdots, \alpha_{d-1}) \in (\mathbb{R}^d)^d \mapsto w_{\phi(x))}(\mathrm{d}\phi_x\alpha_1, \cdots, \mathrm{d}\phi_x\alpha_{d-1}).$$

Definition 15. *[Gla05] Let $S \in M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ be a current, and let $\phi \in \mathrm{Diff}^\ell_0(\mathbb{R}^d)$, the group action of ϕ on S is defined by*

$$\phi \cdot S \doteq \phi_\# S : w \in C^0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d) \mapsto S(\phi^\# w).$$

Thanks to the following proposition it can be shown that $M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ is a shape space:

Proposition 18. *[Gla05] Let* $S \in M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$, $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$ *then* $\phi_\# S \in M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ *and* $|\phi_\# S| \leq |S||\mathrm{d}\phi|_\infty^{d-1}$.

3.3.1.2 Oriented sub manifolds seen as currents

Let S be an oriented C^1 sub-manifold of dimension $d-1$ in $\mathbb{R}^d$ and let $\mathrm{d}\mu$ be its canonical volume form. The shape S can be canonically associated with the following current

$$S : w \in C^0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d) \mapsto S(w) \doteq \int_S w(x)(u_x^1, \cdots, u_x^{d-1})\mathrm{d}\mu$$

where for each $x \in S$, $(u_x^1, \cdots, u_x^{d-1})$ is an orthonormal basis of the tangent space $T_x S$. In the following we identify sub-manifolds S and their currents and we will refer to them as *shapes*. Besides we have the following consistency of transport by diffeomorphisms:

Lemma 1. *[Gla05] Let* S *be an oriented sub-manifold and let* $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$,

$$\phi(S) = \phi_\# S$$

where $\phi(S)$ *is the transport of* S *by* ϕ *by applying* ϕ *to points of* S *and* $\phi_\# S$ *is the transport of* S *as a current.*

The dual norm on $M_s(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ allows then to measure the difference between a transported shape and an other shape. In practice one sets a RKHS W continuously embedded in $C_0(\mathbb{R}^d, \Lambda^{d-1}\mathbb{R}^d)$ and shapes are considered as elements of the dual of W. In our examples we use a RKHS defined thanks to scalar Gaussian kernels. The only parameter that needs to be specified is then its scale, it will corresponds to the scale at which we want to study differences between shapes. Besides, in the practical computation we consider shapes which are discretized sub-manifolds. One can show that these finite-dimensional shapes can still be considered as currents of W^* and that there exists an explicit expression of the distance between two such shapes in W^* (see [Gla05]).

3.3.2 Varifolds

We present here the notion of varifolds based on the work of [Cha13]. Varifolds will be also be defined as a dual of a space of functions, but a different one than for currents so that we no longer need to suppose that shapes are *oriented* sub manifolds.

3.3.2.1 Shape space of varifolds

Definition 16. *[Cha13] The Grassmann manifold of dimension* m *in* $\mathbb{R}^d$, *denoted* $G_m(\mathbb{R}^d)$, *is the set of all* $m-$*dimensional subspaces of* $\mathbb{R}^d$. *It can be identified to the quotient space of all families of* d *independent vectors of* $\mathbb{R}^d$ *by the equivalence relation obtained by identifying families that generate the same subspace.*

In this thesis we consider only the case where $m = d-1$ and in fact G_{d-1} is the real projective space of $\mathbb{R}^d$.

Definition 17. *[Cha13] A **varifold** is an element of $C_0(\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d))'$ (i.e. a Borel finite measure on $\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d)$). For $\mu \in C_0(\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d))'$ and $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$, we define the action of ϕ on μ by*

$$\phi \cdot \mu \doteq \phi_*(\mu) : w \in C_0(\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d)) \mapsto \mu(\phi^* w)$$

where for $w \in C_0(\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d))$, $x \in \mathbb{R}^d$, $V \in G_{d-1}(\mathbb{R}^d)$ and $(u_1, \cdots, u_{d-1})$ an orthonormal basis of V,

$$\mu(\phi^* w)(x, V) \doteq |\mathrm{d}_x\phi(u_1) \wedge \cdots \wedge \mathrm{d}_x\phi(u_{d-1})| w(\phi(x), \mathrm{d}\phi_x V).$$

3.3.2.2 Sub manifolds seen as varifolds

Let S be a *non oriented* C^1 sub manifold. We can associate S with the following varifold

$$\mu_S : w \in C_0(\mathbb{R}^d \times G_{d-1}(\mathbb{R}^d)) \mapsto \int_S w(x, V) \mathrm{d}\mathcal{H}^{d-1}(x, T_x S)$$

where $\mathcal{H}^{d-1}$ is the $d-1$-dimensional Hausdorff measure on $\mathbb{R}^d$. We have the following consistency of transport by diffeomorphisms:

Lemma 2. *[Cha13] Let S be a sub manifold, let $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$,*

$$\phi_* \mu_S = \mu_{\phi(S)}.$$

Similarly to currents, in practice the varifold norm is computed through the introduction of a scalar Gaussian RKHS, and the choice of its scale is an important parameter. Once it is setted, when shapes are discretized sub manifolds, it is possible to obtain an explicit expression of the distance between to of them.

Chapter 4

Building modular large deformations

Contents

4.1 Deformation model

As explained in Section 2.3, the object of this thesis is the development of a generic framework to build *modular* deformations. These deformations would be modular in the sense that they are defined thanks to a few generators of deformations, each one generating a particular type of local and interpretable transformation. One goal is to be able to easily incorporate constraints on the deformation model. This constraints could correspond to an additional knowledge one would have on the shapes under study, or to a point of view thanks to which one would want to study these shapes. A second goal is then to use this modular framework to understand the variability between different shapes.

4.1.1 Deformation modules

4.1.1.1 Definition

The framework we propose relies on the new notion of **deformation module**. Intuitively, a deformation module is a structure that embeds a vector field generation mechanism. This mechanism depends on some geometrical descriptors, which specify the local structure of the induced vector fields (think about the positioning of some actuators), and a finite numbers of control parameters commanding the actuators. For a given positioning, a cost is specified for any possible values of the control parameters, so that optimal policies can be defined. For each resulting vector fields, a feedback mechanism is defined to update the positioning of the geometrical descriptors. Desirable consistency properties lead to consider the geometrical descriptors as defining a shape on which one has a diffeomorphic action. Hence following [You12], the positioning of the geometrical descriptors will be represented as a shape in a shape space.

In the following we set $d \in \mathbb{N}^*$ the dimension of the ambient space.

Definition 18. *Let $k, \ell \in \mathbb{N}^*$. We define $M = (\mathcal{O}, H, \zeta, \xi, c)$ as a C^k-**deformation module of order** ℓ with **geometrical descriptors** in $\mathcal{O}$, **controls** in H, **infinitesimal action** ξ, **field generator** ζ and **cost** c, if*

- *$\mathcal{O}$ is a C^k-shape space of $\mathbb{R}^d$ of order ℓ of finite dimension with infinitesimal action $\xi : C_0^\ell(\mathbb{R}^d) \times \mathcal{O} \longrightarrow T\mathcal{O}$,*
- *H is a finite dimensional Euclidean space,*
- *$\zeta : (o, h) \in \mathcal{O} \times H \to (o, \zeta_o(h)) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d)$ is continuous, with $h \mapsto \zeta_o(h)$ linear and $o \mapsto \zeta_o$ of class C^k,*
- *$c : (o, h) \in \mathcal{O} \times H \to c_o(h) \in \mathbb{R}^+$ is a continuous mapping such that $o \mapsto c_o$ is smooth and for all $o \in \mathcal{O}$, $h \mapsto c_o(h)$ is a positive definite quadratic form on H, thus defining smooth metric on $\mathcal{O} \times H$.*

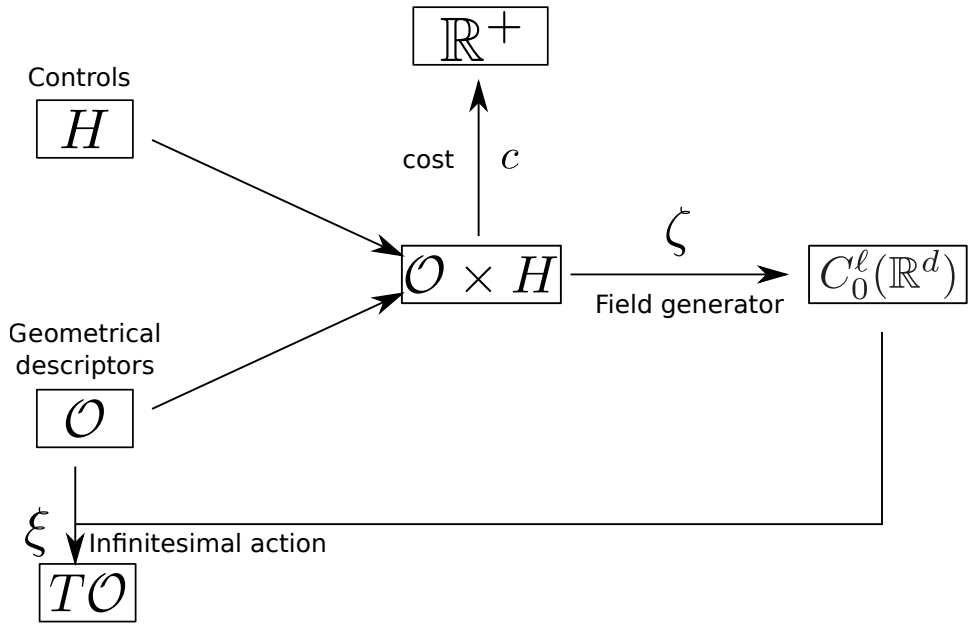


Figure 4.1: Schematic view of a deformation module.

A deformation module is then defined by the way it can generate a vector field, which is given by the field generator ζ, and by the feedback action on vector fields, which is given by the infinitesimal action ξ. A schematic view of the construction of a module can be seen in figure 4.1.

Remark 10. *By definition, the geometrical descriptors of a deformation module are considered as a "shape" in the shape space $\mathcal{O}$. In the following examples, these "shapes" are the centers of the scaling, of the rotation or the base points of translation vectors. The fact that they are "shapes" in the sense of Definition 13 allows us to use the global deformation under construction to transport them via the infinitesimal action ξ. These "shapes" may or may not coincide with points from the input shape data set. For example, it is sensible not to locate the center of a scaling on the boundaries of an object. We will see in the following that the input shape data may be considered as geometrical descriptors of a particular module, called a silent module (see Section 4.2.8), which is deformed thanks to its infinitesimal action, but which does not contribute to the global velocity field.*

4.1.1.2 First examples

We will now present two examples of simple deformation modules, a richer presentation of different deformation modules will be done in Section 4.2

4.1.1.2.1 First example: sum of local translations This first example explains how the construction of [DPGJ11] can be seen as a C^k-deformation module of order ℓ for any $k, \ell \in \mathbb{N}^*$. We want to build a deformation module M that generates vector

fields that are a sum of D local translations acting at scale σ in the ambient space $\mathbb{R}^d$ ($d \in \mathbb{N}^*$). We set V_σ the scalar Gaussian RKHS of scale σ (see Section 3.1, its kernel will be denoted K_σ), $\mathcal{O} \doteq (\mathbb{R}^d)^D$ the shape space of D landmarks with infinitesimal action $\xi_o : v \in C_0^\ell(\mathbb{R}^d) \mapsto (v(z_i))_i$, where $o = (z_i) \in \mathcal{O}$ (application of the vector field at each point), and $H \doteq (\mathbb{R}^d)^D$ (families of D vectors). For $o = (z_i) \in \mathcal{O}$, we define $\zeta_o : h = (\alpha_i) \in H \mapsto \sum_{i=1}^D K_\sigma(z_i, \cdot)\alpha_i$ and $c_o : h = (\alpha_i) \in H \mapsto |\sum_i K_\sigma(z_i, \cdot)\alpha_i|^2_{V_\sigma}$. It is easy to show that $M = (\mathcal{O}, H, \zeta, \xi, c)$ defines a deformation module of order ℓ.

4.1.1.2.2 Second example: local scaling or rotations Let $\ell, k \in \mathbb{N}^*$. The second example is a C^k-deformation module of order ℓ which generates vector fields that are local scalings of fixed scale $\sigma \in \mathbb{R}^+$ in the ambient space $\mathbb{R}^2$. A local scaling is parametrized by its center $o \in \mathbb{R}^2$ and its scaling factor $h \in \mathbb{R}$. The point o plays the role of the geometrical descriptor, and the factor h the control parameter. From o, we build 3 points $z_j(o)$ and 3 unit vectors d_j as described in Figure 4.2 (they also depend on the fixed parameter σ). The idea is to build the vector field generated by the geometrical descriptor o and the control h as an interpolation of the values at these points $z_j(o)$ thanks to vectors d_j: $\zeta_o(h) \doteq h \sum_{j=1}^3 K_\sigma(z_j(o), \cdot)d_j$, where K_σ is the kernel of the scalar gaussian RKHS of scale σ. We emphasize here that points $z_j(o)$ and vectors d_j are *intermediate tools* used to build the vector field but that the latter is only parametrized by o and h. We then define the deformation module M by the following spaces : $\mathcal{O} \doteq \mathbb{R}^2$, $H = \mathbb{R}$ and the following mappings : for $o \in \mathcal{O}$, $\zeta_o : h \in H \mapsto \zeta_o(h)$ as given above, $\xi_o : v \in C_0^\ell(\mathbb{R}^2) \mapsto v(o)$, the velocity field at the scaling center, and $c_o : h \in H \mapsto |\zeta_o(h)|^2_{V_\sigma} = h^2 \sum_{j,j'} K_\sigma(z_j, z_{j'}) d_j^T d_{j'}$, the squared norm of the generated velocity field in the RKHS V_σ. Then $M = (\mathcal{O}, H, \zeta, \xi, c)$ defines a C^k-deformation module of order ℓ. Indeed space $\mathcal{O}$ is a C^k-shape space (of one landmark) of order ℓ and $\zeta : o \in \mathcal{O} \mapsto \zeta_o \in L(H, C_0^\ell(\mathbb{R}^2))$ is locally Lipschitz because distances between $z_i(o)$ and $z_i(o')$ $(1 \leq i \leq 3)$ are the same as the distance between o and o'. For the same reason ζ is globally continuous. Other properties can be easily verified. This construction can be generalised to any other affine deformation by changing the rule to build vectors d_j, as for example in Figure 4.3 where the generated vector field is a local rotation.

4.1.1.3 Uniform Embedding Condition

A key point in the design of a consistent approach for deformation modules is the possibility to deduce existence of optimal deformations between shapes as solutions of the optimal control problem associated with the choice of a cost for the control parameters. This requirement imposes constraints on the choice of costs. For instance, it seems important for the cost to be related with some metric of the induced vector field. This is the spirit of the following embedding condition:

Definition 19. *Let $M = (\mathcal{O}, H, \zeta, \xi, c)$ be a C^k-deformation module of order ℓ. We say that M satisfies the **Uniform Embedding Condition (UEC)** if there exists a Hilbert space of vector fields V continuously embedded in $C_0^{\ell+k}(\mathbb{R}^d)$ and a constant $C > 0$ such that for all $o \in \mathcal{O}$ and for all $h \in H$, $\zeta_o(h) \in V$ and*

$$|\zeta_o(h)|^2_V \leq C c_o(h).$$

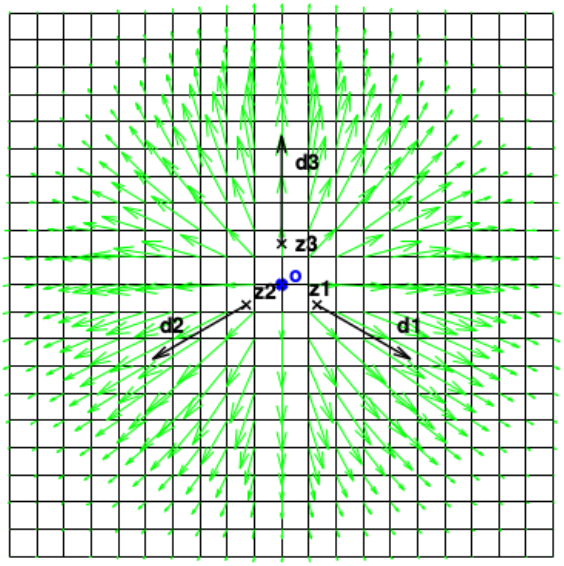


Figure 4.2: Local scaling. Geometrical descriptor o (in blue) and intermediate tools (in black). Plot of the resulting vector field in green.

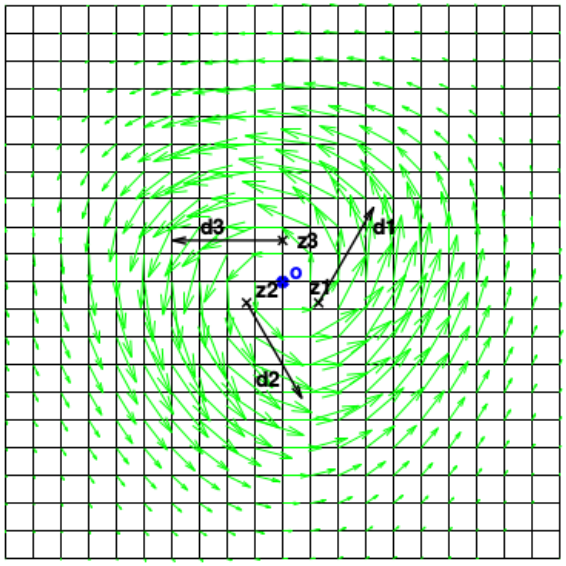


Figure 4.3: Local rotation. Geometrical descriptor o (in blue) and intermediate tools (in black). Plot of the resulting vector field in green.

Remark 11. *Examples of deformation modules presented previously satisfy this UEC. Indeed in these two examples the infinitesimal action ξ takes values in the Hilbert space V_σ which is continuously embedded in $C_0^{\ell+k}(\mathbb{R}^d)$ (see [Gla05]), and the cost is defined by the squared-norm of the generated vector field in V_σ.*

4.1.1.4 Combination

We have presented in the previous section examples of simple, base deformation modules generating *simple* vectors fields. However if one wants to build more complex vector fields, building the corresponding deformation module might not be straightforward. In order to build such complex deformation module, we will define them as the *combination* of simpler ones.

Definition 20. *Let $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, be L C^k-deformation modules of order ℓ. We define the* ***compound module*** *of modules M^l by $\mathcal{C}(M^l, l = 1 \cdots L) = (\mathcal{O}, H, \zeta, \xi, c)$ where $\mathcal{O} \doteq \prod_l \mathcal{O}^l$, $H \doteq \prod_l H^l$ and for $o = (o^l)_l \in \mathcal{O}$, $\zeta_o : h = (h^l) \in H \mapsto \sum_l \zeta^l_{o^l}(h^l)$, $\xi_o : v \in C_0^\ell(\mathbb{R}^d) \mapsto (\xi^l_{o^l}(v))_l \in T_o\mathcal{O}$ and $c_o : h = (h^l) \in H \mapsto \sum_l c^l_{o^l}(h^l)$.*

A schematic view of this combination can be seen in Figure 4.4. A key point of our framework is the following stability result under combination:

Proposition 19. *If $M^l = (\mathcal{O}^l, H^l, \zeta^l, \xi^l, c^l)$, $l = 1 \cdots L$, are C^k-deformation modules of order ℓ, then $\mathcal{C}(M^l, l = 1 \cdots L)$ is a C^k-deformation module of order ℓ. Furthermore, if each M^l satisfies UEC, then $\mathcal{C}(M^l, l = 1 \cdots L)$ also satisfies UEC.*

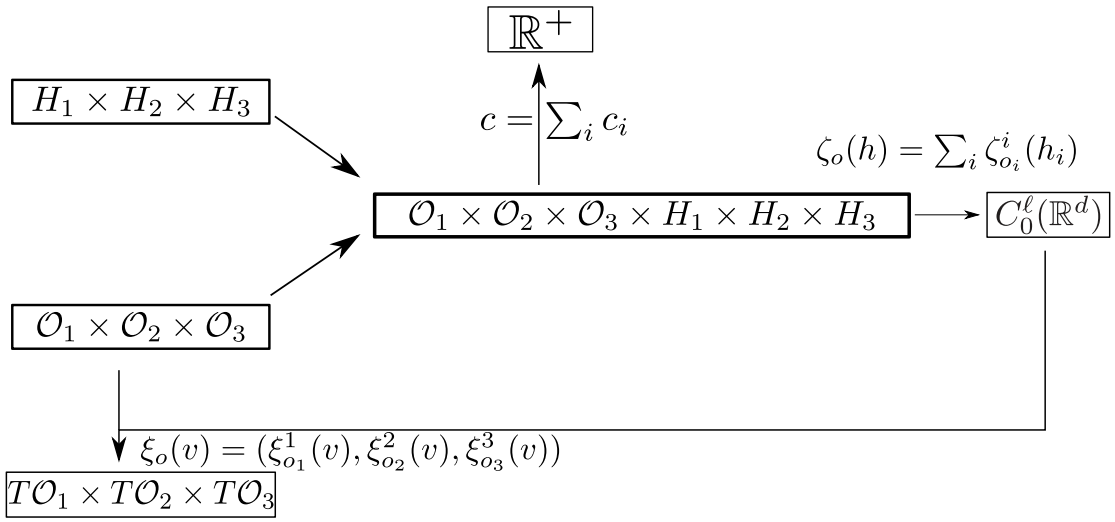


Figure 4.4: Schematic view of a combination of three deformation modules.

Proof. It is clear that $\mathcal{C}(M^l, l = 1 \cdots L)$ is a C^k-deformation module of order ℓ. Let suppose that each M^l satisfies UEC. We define

$$\pi : w = (w_1, ..., w_L) \in W \doteq \prod_l V^l \mapsto \sum_i w_i \in C^\ell_0(\mathbb{R}^d).$$

Then space $V \doteq \pi(W)$ can be equipped with the following norm: for $v \in V$, $|v|^2_V = \inf\{\sum_l |v_l|^2_{V^l} \mid \pi((v_l)_l) = v\}$, such that it is a Hilbert space continuously embedded in $C^\ell_0(\mathbb{R}^d)$. For any $o = (o^l) \in \mathcal{O}$ and $h = (h^l) \in H$ we have

$$|\zeta_o(h)|^2_V \leq \sum_{l=1}^{L} |\zeta^l_{o^l}(h^l)|^2_{V_l} \leq \sum_{l=1}^{L} C_l c^l_{o^l}(h^l) \leq (\max_{1 \leq l \leq L} C_l) c_o(h)$$

and then $\mathcal{C}(M^l, l = 1 \cdots L)$ satisfies UEC. □

Remark 12. *Note that even if costs of elementary modules M^l are given by $c^l_{o^l}(h^l) = |\zeta^l_{o^l}(h^l)|^2_{V^l}$ as in our previous examples, in general (when π is not one to one) the cost of the compound module is not the squared norm of the compound velocity field $\zeta_o(h) = \sum_l \zeta^l_{o^l}$ in the global embedding RKHS V, i.e. $c_o(h) = \sum_l |\zeta^l_{o^l}(h^l)|^2_{V^l} \neq |\zeta_o(h)|^2_V$. Then in general $C > 1$ and c is not the pullback metric on $\mathcal{O} \times H$ of the metric on $\mathcal{O} \times V$. The cost $c_o(h)$ does not directly depend on the norm of the generated vector field $\zeta_o(h)$ but on its specific decomposition as a sum of elementary vector fields $\zeta^l_{o^l}(H^l)$. As we will consider configurations (o, h) that minimize the cost $c_o(h)$ (for a given action $\xi_o(\zeta_o(h))$ on the geometrical descriptor), different choices of cost c^l can favour different decompositions for the same resulting vector field. Moreover in practice one can compute easily the cost of the compound module c from the elementary costs c^l.*

Remark 13. *Note that the combination of deformation module is an associative and commutative law.*

4.1.2 Large deformations

In this section, we show how large deformations can be generated from a given deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfying UEC (see Definition 19, we denote V the corresponding Hilbert space). These large deformations are obtained by the integration of a trajectory of vector fields $v : t \in [0,1] \mapsto v_t \in V$ that are modular, meaning that at each time t one can write $v_t = \zeta_{o_t}(h_t)$ with $(o_t, h_t) \in \mathcal{O} \times H$. During the integration of the trajectory we want the geometrical descriptor of the module to be transported by the flow and therefore, denoting φ^v the flow of v (see Proposition 15), that v_t belongs to $\zeta_{o_t}(H)$, with $o_t = \varphi_t^v(o_{t=0})$.

Definition 21. *Let $a, b \in \mathcal{O}$. We denote $\Omega_{a,b}$ the set of mesurable curves $t \mapsto (o_t, h_t) \in \mathcal{O} \times H$ where o_t is absolutely continuous (a.c.), starting from a and ending at b, such that, for almost every $t \in [0,1]$, $\dot{o}_t = \xi_{o_t}(v_t)$, where $v_t \doteq \zeta_{o_t}(h_t)$, and*

$$E(o,h) \doteq \int_0^1 c_{o_t}(h_t)dt < \infty.$$

*The quantity $E(o,h)$ is called the **energy** of (o,h) and $\Omega_{a,b}$ is the set of **controlled paths of finite energy** starting at a and ending at b.*

If UEC is satisfied, we can build large deformations from these trajectories :

Proposition 20. *Let us suppose that M satisfies UEC. Let $(o,h) \in \Omega_{a,b}$ and for each t, $v_t = \zeta_{o_t}(h_t)$. Then $v \in L^2([0,1], V) \subset L^1$, the flow φ^v exists, $h \in L^2([0,1], H)$ and for each $t \in [0,1]$, $o_t = \phi_t^v.o_0$. The final diffeomorphism $\varphi_{t=1}^v$ is called a **modular large deformation generated by** a.*

Proof. Let $(o,h) \in \Omega_{a,b}$ and $v : t \in [0,1] \mapsto v_t \doteq \zeta_{o_t}(h_t)$. From the UEC we get: $\int_0^1 |v(t)|_V^2 dt \leq C \int_0^1 c_{o_t}(h_t)dt < \infty$ since $(o,h) \in \Omega_{a,b}$. Then $v \in L^2([0,1], V) \subset L^1([0,1], V) \subset L^1([0,1], C_0^\ell(\mathbb{R}^d))$ and its flow φ^v can be defined. As explained in Proposition 16 we can deduce that $o(t) = \varphi_t^v \cdot a$ for each t.
Let us now define, for $o \in \mathcal{O}$, $||c_o^{-1}|| := \sup_{|h|_H=1} c_o(h)^{-1}$ (well defined as H is of finite dimension). As $o \mapsto c_o$ and $t \mapsto o(t) = \varphi_t^v \cdot a$ are continuous, $\sup_t ||c_{o_t}^{-1}||$ is finite and then $\int_0^1 |h_t|_H^2 dt \leq (\sup_t ||c_{o_t}^{-1}||) \int_0^1 c_{o_t}(h_t)dt < \infty$ so $h \in L^2([0,1], H)$.

□

Remark 14. *A modular large deformation is parametrized by an initial value of geometrical descriptor $o_{t=0}$ and a trajectory of control $h \in L^2([0,1], H)$.*

4.2 Collection of deformation modules

We present here a list of possible deformation modules. Most of these modules will be used in the numerical experiments in Chapter 7. Some modules will be presented only in dimension 2, but may be easily generalized in higher dimensions. In the first section we present the different shape spaces of geometrical descriptors (and their infinitesimal action) that will be used. In next sections will be presented different examples of deformation modules that can be built thanks to these geometrical descriptors.

4.2.1 Different spaces of geometrical descriptors

4.2.1.1 Points

A widely used case will be when the geometrical descriptor is made of one point: the space of geometrical descriptors is $\mathcal{O} \doteq \mathbb{R}^d$.

4.2.1.1.1 Fixed points A first possibility is that the geometrical descriptor belongs to a fixed background and then is not transported by diffeomorphisms. In this case the action of diffeomorphisms is $(o, \phi) \in \mathcal{O} \times \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto o$ and then the infinitesimal action is $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto 0$.

4.2.1.1.2 Transported point However in general one wants for points generating vector fields to be transported with the ambient space. The natural corresponding action of diffeomorphisms is $(o, \phi) \in \mathcal{O} \times \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto \phi(o)$ and then the infinitesimal action is $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto v(o)$.

4.2.1.2 Vectors attached to points

We will now study the case where the geometrical descriptor is made of a vector attached to a point: here the geometrical prior on the generated vector field will depend on a direction and not only a localisation. We give different possible rules for updating this direction during the integration process.

4.2.1.2.1 Fixed direction A first possibility is for the vector to be kept constant during integration, meaning that it is linked to a fixed background and not to the shape that is deformed. The shape space is $\mathcal{O} \doteq \mathbb{R}^d \times \mathbb{R}^d$, the action of diffeomorphisms is $(o, \phi) \in \mathcal{O} \times \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto (\phi(z), u)$ and then the infinitesimal action is $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), 0)$, with $o = (z, u)$.

4.2.1.2.2 Direction updated by the differential If one wants the direction attached to a point to be transported, a natural way is to update it thanks to the derivative of the diffeomorphism. The shape space is still $\mathcal{O} \doteq \mathbb{R}^d \times \mathbb{R}^d$, the action of diffeomorphisms becomes here $(o, \phi) \in \mathcal{O} \times \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto (\phi(z), \mathrm{d}\phi_z \cdot u)$ and we deduce the infinitesimal action $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), \mathrm{d}v_z \cdot u)$, with $o = (z, u)$.

4.2.1.2.3 Direction of constant norm In the previous action of diffeomorphisms, the norm of the vector changes while it is transported by the diffeomorphism whereas sometimes it may be interesting for it to be of constant norm. In this framework we need to ensure that this norm is not null and then we define a new shape space $\mathcal{O} \doteq \mathbb{R}^d \times (\mathbb{R}^d - \{0\})$ associated with the action $(o, \phi) \in \mathcal{O} \times \mathrm{Diff}_0^\ell(\mathbb{R}^d) \mapsto (\phi(z), \frac{\mathrm{d}\phi_z \cdot u}{|\mathrm{d}\phi_z \cdot u|})$ where $o = (z, u)$. Then the infinitesimal action is given by $(o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), \frac{\mathrm{d}v_z \cdot u}{|u|} - \frac{\langle u, \mathrm{d}v_z \cdot u \rangle}{|u|^3} u)$.

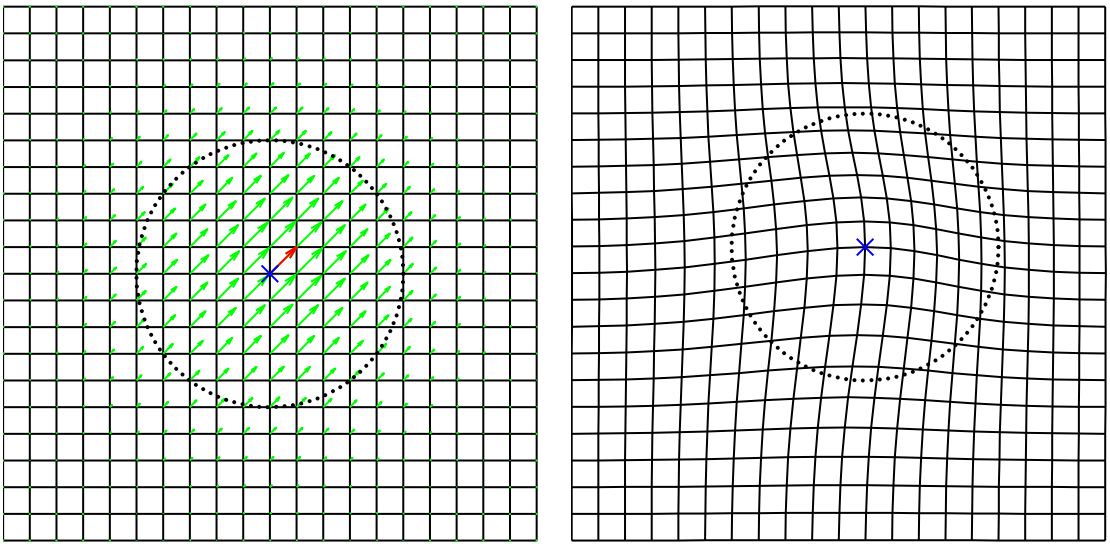


Figure 4.5: Example of vector field generated by the deformation module generating one unconstrained translation at scale σ. Geometrical descriptor (centre of translation): blue cross; Control (vector of translation): red arrow; the dotted circle is centred at geometrical descriptor and its radius is σ. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field.

4.2.1.2.4 Direction updated by adjoint action

A last possibility is for the vector to be updated by the adjoint action. Here there is no need to ensure that the norm of the vector is not null and then the shape space is $\mathcal{O} \doteq \mathbb{R}^d \times \mathbb{R}^d$. The action of the deformation ϕ on the geometrical descriptors o is $(\phi, o = (z, u)) \mapsto (\phi(z), (\mathrm{d}\phi^{-1}_{\phi(x)})^T u)$, where the second part is the so-called adjoint action. If one considers the direction u as the normal to an infinitesimal surface patch, then $(\mathrm{d}\phi^{-1}_{\phi(x)})^T u$ is the direction of the normal of the deformed surface patch. The infinitesimal action on geometrical descriptors writes now $\xi_o(v) = (v(z), -dv_z^T u)$ (with $o = (z, u)$).

4.2.2 Deformation module generating a sum of local translations

4.2.2.1 Unconstrained local translations

This deformation module enables us to see the construction of [DPGJ11] as an instance of a deformation module and was detailed in Section 4.1.1.2.1. The cost of this deformation module is $c : (o = (z_i), h = (\alpha_i)) \mapsto |\zeta_o(h)|^2_{V_\sigma} = \sum_{i,j} K_\sigma(z_i, z_j)\alpha_i^T \alpha_j$. Another possibility to generate sums of translations would be to consider the combination of mod-

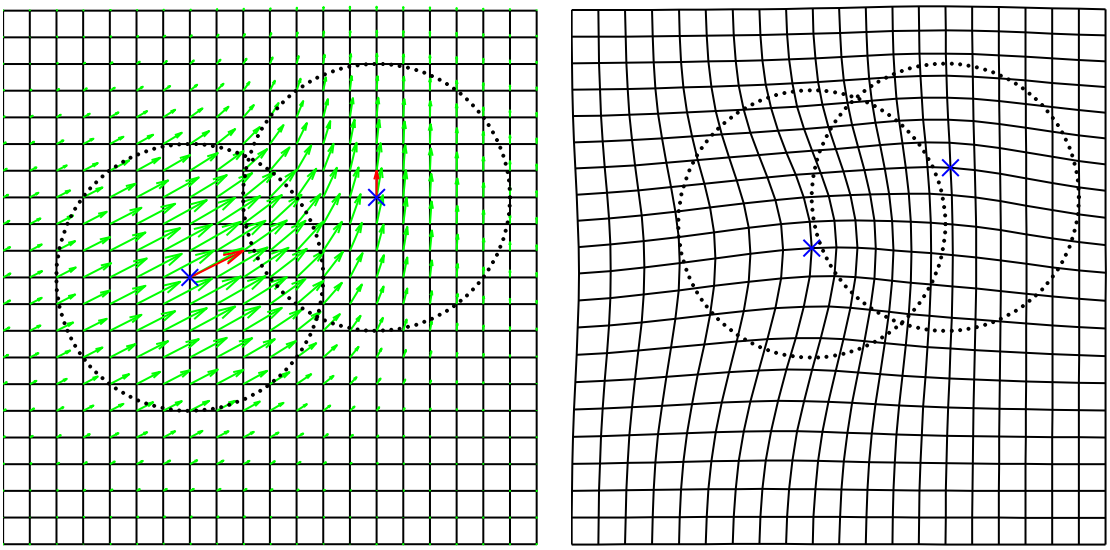


Figure 4.6: Example of vector field generated by the deformation module generating a sum of two unconstrained translations at scale σ. Geometrical descriptor (centres of translations): blue crosses; Control (vectors of translation): red arrows; the dotted circles are centred at geometrical descriptor (the two centres of translations) and of radius σ. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field.

ules creating one translation, such that the cost would be $\sum_i |K_\sigma(\cdot, z_i)\alpha_i|^2_{V_\sigma} = \sum_i |\alpha_i|^2$. The first case is more interesting because minimizing the cost tends to separate centres z_i and then forces local translations of the same scale to act everywhere it is needed, while in the second case several could converge to the same location, making it redundant. In Figures 4.5 and 4.6 we present two examples of vector fields that can be generated by such deformation modules. We also present the deformed grid and the transported geometrical descriptors under this infinitesimal action. In Figure 4.5 we present the case of the deformation module generating one translation ($\mathcal{O} = \mathbb{R}^2$), and in Figure 4.6 we present the case of the deformation module generating a sum of two translations ($\mathcal{O} = \mathbb{R}^2 \times \mathbb{R}^2$).

4.2.2.2 Example of combination: a multi-scale sum of local translations

Let us set P scales σ_l and for each l a number of translations D_l. We present here the construction of the deformation module that generates vector fields that are a sum of multi-scale local translations, with D_l translations for each scale σ_l. For each l can be built, as defined in Section 4.1.1.2.1, a module M^l generating vector fields

that are a sum of D_l local translations acting at scale σ_l. The multi-scale module M is then the combination of these modules M^l (see Section 4.1.1.4). In particular the vector field created by geometrical descriptor $o = (z_j^l) \in \mathcal{O} \doteq \Pi_l \left((\mathbb{R}^d)^{D_l}\right)$ and control $h = (\alpha_j^l) \in H \doteq \Pi_l \left((\mathbb{R}^d)^{D_l}\right)$ is $\zeta_o(h) = \sum_l \sum_j K_{\sigma_l}(z_j^l, \cdot)\alpha_j^l$. It is clear here that, unlike in the framework of [RVW+11] where vector fields are generated thanks to a sum of gaussian kernels, centres of local translations are different for each scale. This property is kept for optimal trajectories (minimising the cost) as we *force* these points to be different at initial time. The cost is, for $o = (z_j^l) \in \mathcal{O}$ and $h = (\alpha_j^l) \in H$, $c_o(h) = \sum_l \sum_j K_{\sigma_l}(z_j^l, z_{j'}^l)\alpha_j^{lT}\alpha_{j'}^l$. Even though this cost is similar to the framework of vector bundle presented in [SNLP11], the control variables which are optimised in our framework are only the vectors of local translations $h = (\alpha_j^l) \in H$ and not the global vector field. Therefore, optimal trajectories are different from the vector bundle framework and the decomposition of the vector field in a sum of local translations acting at different scales and centres at different points is preserved.

Remark 15. *This example corresponds to one of the models that we first developed which was presented in Section 2.3.2.*

4.2.2.3 Deformation module generating a sum of local translations with a prior on the directions

In the previous examples of deformation modules based on local translations (Sections 4.2.2.1 and 4.2.2.2), the direction and magnitude of the translation vector were considered as control parameters. Therefore, during the integration of the flow, the direction of the translation at each time needs to be determined as a optimal solution for a given criterion. By contrast, we may want to update the direction of the translation during deformation by using a prior rule, thus considering the direction no more as a control parameter but as a geometrical descriptor instead. In this case, only the magnitude of the translation vector is considered as a control parameter and the geometrical parameter is formed of a vector attached to a point. The shape space of geometrical descriptors $\mathcal{O}$ and the infinitesimal action ξ on the geometrical descriptor can then be chosen between the four possibilities presented in Section 4.2.1.2. We set a scale $\sigma \in \mathbb{R}_+^*$. The space of control is $H \doteq \mathbb{R}$ and we define the field generator $\zeta : (o, h) \in \mathcal{O} \times H \mapsto hK_\sigma(z, \cdot)u$ with $o = (z, u)$. The cost is given by $c_o(h) = |\zeta_o(h)|_{V_\sigma}^2 = |u|^2h^2$ (with $o = (z, u)$). Then $M \doteq (\mathcal{O}, H, \zeta, \xi, c)$ is a deformation module satisfying UEC. If we consider a compound module combining this deformation module with others, the direction of the translation at integration time t will only depend of the generated global diffeomorphism ϕ and the direction u at time $t = 0$. This deformation module M can be generalized to the deformation module $\tilde{M} \doteq (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ generating vector fields that are a sum of P local translations at scale σ by setting $\tilde{\mathcal{O}} \doteq \mathcal{O}^P$, $\tilde{H} \doteq \mathbb{R}^P$ (the control is made of P scalars), and for $o = (o_i)_{1 \leq i \leq P} \in \tilde{\mathcal{O}}$, $h = (h_i) \in H$, $v \in C_0^\ell(\mathbb{R}^d)$, $\tilde{\zeta}_o(h) \doteq \sum_i \zeta_{o_i}(h_i)$, $\tilde{\xi}_o(v) = (\xi_{o_i}(v))_i$ and $c_o(h) = |\tilde{\zeta}_o(h)|_{V_\sigma}^2$. Such deformation modules will be used in examples in Section 7.3 and 7.4. In Figures 4.7 and 4.8 we present two examples of vector fields that can be generated by deformation modules generating a sum of two transla-

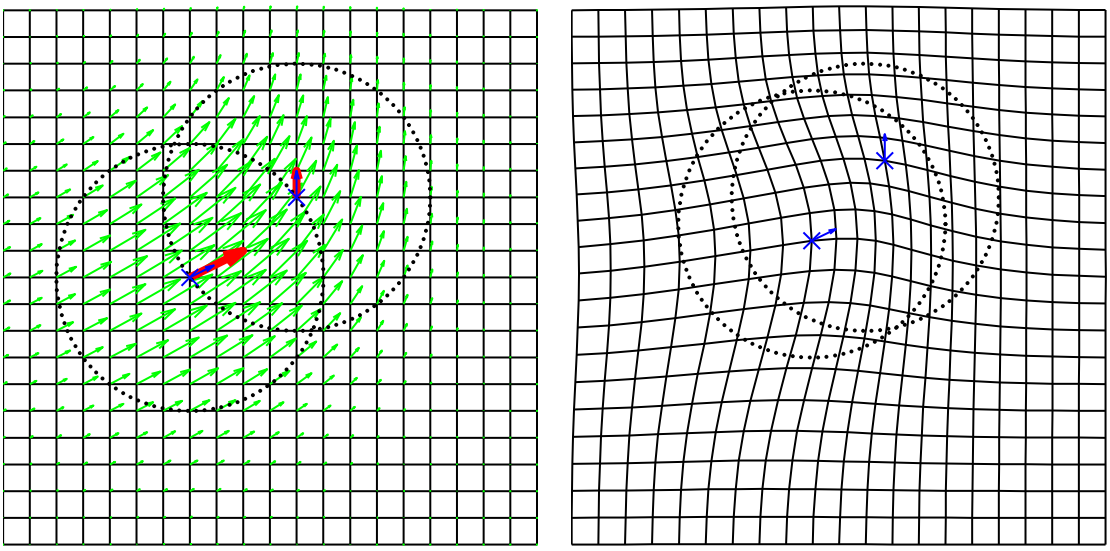


Figure 4.7: Example of vector field generated by the deformation module generating a sum of two translations at scale σ with fixed directions. Geometrical descriptor (2 points and 2 vectors): blue crosses for centres of translations and blue arrows for directions of translations; control (2 scalars): lengths and directions of the two red arrows; the dotted circles are centred at centres of translations and of radius σ. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field (centres of translations are transported but directions remain constant).

tions with priors on directions, as well as the displacement of geometrical descriptors under this infinitesimal action. In Figure 4.7 we present the case where directions are fixed (the infinitesimal action on them is null: they stay steady under actions on vector fields) while in Figure 4.8 we present the case where directions are updated by the differential and are of constant norm.

4.2.3 Constrained local transformations

In Section 4.1.1.2 we presented the example of a deformation module generating vector fields in dimension 2 that are always a local scaling at a fixed scale σ. Such a local vector field, centred at a point $o \in \mathbb{R}^2$ and acting at a scale σ, is generated thanks to local translations carried by 3 points z_i close to o with respect to σ. Points z_i are chosen in a fixed manner from o, at a distance from o that we set equal to $\delta \doteq \frac{1}{3}\sigma$. Then in dimension 2, in order to build a local vector field at scale σ around $o \in \mathbb{R}^2$ we define: $z_1(o,\sigma) = o + \delta(1,0)$, $z_2(o,\sigma) = o + \delta(\frac{1}{2}\sqrt{3}, -\frac{1}{2})$ and $z_3(o,\sigma) = o + \delta(-\frac{1}{2}\sqrt{3}, -\frac{1}{2})$. This setting can be generalized in dimension 3: in order to build a local vector

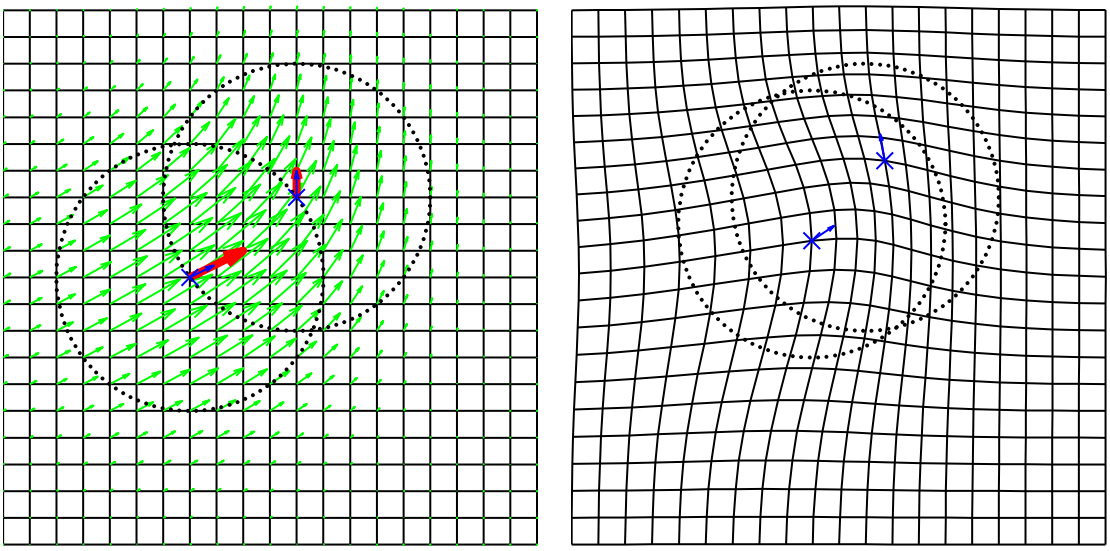


Figure 4.8: Example of vector field generated by the deformation module generating a sum of two translations at scale σ with directions of constant norm. Geometrical descriptor (2 points and 2 vectors): blue crosses for centres of translations and blue arrows for directions of translations; control (2 scalars): lengths of the two red arrows; the dotted circles are centred at centres of translations and of radius σ. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field (centres and directions of translations are transported).

field at scale σ around $o \in \mathbb{R}^3$ we build 4 points z_1 as vertices of a regular polygon centred at o, *i.e.* we set $z_1(o,\sigma) = o + \frac{\delta}{\sqrt{3}}(-1,-1,1)$, $z_2(o,\sigma) = o + \frac{\delta}{\sqrt{3}}(-1,1,-1)$, $z_3(o,\sigma) = o + \frac{\delta}{\sqrt{3}}(1,-1,-1)$ and $z_4(o,\sigma) = o + \frac{\delta}{\sqrt{3}}(1,1,1)$.

Let us set a point $o \in \mathbb{R}^d$ ($d = 2$ or 3) and build a local vector field v centred at o and equal to 0 at o. Similarly to the example presented in Section 4.1.1.2, we build it as the interpolation of local translations centred at points $z_i(o,\sigma)$. Vectors of these translations are taken as the values of the wanted vector field at these points so that the built vector field is their interpolation. The control associated to this deformation module is a scalar, and the generated vector field is its multiplication with the sum of these $d+1$ translations. We emphasize here that points $z_i(o,\sigma)$ and vectors of the translations they carry are intermediate tools used to compute the vector field but that this latter is totally defined by the value of the geometrical descriptor and the control.

In a more general setting, we can build the deformation module that generates vector fields that are a sum of P replications of the same local vector field v at P different locations. We set $\mathcal{O} \doteq (\mathbb{R}^d)^P$, $H \doteq \mathbb{R}^P$ and, for $o = (o_i)_i \in \mathcal{O}$,

$\zeta_o : h = (h_i) \in H \mapsto \sum_{i=1}^{P} h_i \sum_{j=1}^{d+1} K_\sigma(z_j(o_i,\sigma),\cdot) d_j(o_i,\sigma)$ (with $d_j(o_i,\sigma) = v(z_j(o_i,\sigma))$), $\xi_o : v \in C_0^\ell(\mathbb{R}^d) \mapsto (v(o_i))_i \in T\mathcal{O}$ (application of the vector field to each point), $c_o : h = (h_i)_i \in H \mapsto |\zeta_o(h)|^2_{V_\sigma}$. As for the deformation module generating one local scaling, it can be easily shown that $M = (\mathcal{O}, H, \zeta, \xi, c)$ is a deformation module satisfying UEC.

We introduce here two additional example of this construction.

4.2.3.1 Example of combination: a local scaling and a local rotation

We have presented in Section 4.1.1.2 how one could build deformation modules generating locally a scaling or a rotation. We consider here the deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ obtained by the combination of two deformation modules generating constrained local transforms in dimension 2: the first one $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ generates a scaling at scale σ_1 and the second one $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ generates a rotation at scale $\sigma_2 = \sigma_1/3$. Let us detail elements defining M. The space of geometrical descriptor is the space couples of points (centres of a scaling and a rotation): $\mathcal{O} = \mathcal{O}^1 \times \mathcal{O}^2 = \mathbb{R}^2 \times \mathbb{R}^2$. The space of controls is the space of couples of two scalars: $H = H^1 \times H^2 = \mathbb{R} \times \mathbb{R}$. Let $o = (o^1, o^2) \in \mathcal{O}$ and let $h = (h^1, h^2) \in H$, we define now how the vector field $\zeta_o(h)$ is built. We use the definition of points $z_j(o,,\sigma)$ from a centre point of $\mathbb{R}^2$ and a scale σ as given previously. These points will carry local translations enabling to build the local scaling centred at o^1 and the local rotation centred at o^2. We need to define vectors of these translations: they are the value that we want the vector field to take at these points. Then for the scaling we define $d_1^1 = (0,1)$, $d_2^1 = (\frac{1}{2}\sqrt{3}, -\frac{1}{2})$, $d_3^1 = (-\frac{1}{2}\sqrt{3}, -\frac{1}{2})$ and for the rotation we define $d_1^2 = (-1,0)$, $d_2^2 = (\frac{1}{2}, \frac{1}{2}\sqrt{3})$, $d_3^2 = (\frac{1}{2}, -\frac{1}{2}\sqrt{3})$. These points and these vectors are the needed intermediate tools and can be seen in Figure 4.9. Then the vector field generated by o and h is

$$\zeta_o(h) = h^1 \sum_{j=1}^{3} K_{\sigma_1}(z_j(o^1,\sigma_1),\cdot) d_j^1 + h^2 \sum_{j=1}^{3} K_{\sigma_2}(z_j(o^2,\sigma_2),\cdot) d_j^1.$$

The infinitesimal action is given by the application to each point of the geometrical descriptor: $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(o^1), v(o^2))$ where $o = (o^1, o^2)$. Last, the cost is given by the sum of the two costs: for $o = (o^1, o^2) \in \mathcal{O}$ and $h = (h^1, h^2) \in H$,

$$\begin{aligned} c_o(h) &= c^1_{o^1}(h^1) + c^2_{o^2}(h^2) \\ &= (h^1)^2 \textstyle\sum_{j,j'} K_{\sigma_1}(z_j(o^1,\sigma_1), z_{j'}(o^1,\sigma_1)) d_j^{1T} d_{j'}^1 \\ &\quad + (h^2)^2 \textstyle\sum_{j,j'} K_{\sigma_2}(z_j(o^2,\sigma_2), z_{j'}(o^2,\sigma_2)) d_j^{2T} d_{j'}^2 \end{aligned}$$

In Figure 4.9 we present an example of large deformation (see Proposition 20) generated by M. The deformation at each time is represented by the deformation of the grid. We can see that geometrical descriptors are transported by the global flow created by the two deformation modules. As its scale is smaller, the area of action of the local rotation is smaller than the one of the scaling. Then the area of influence of the local rotation is transported by the vector field created by the local scaling, while the

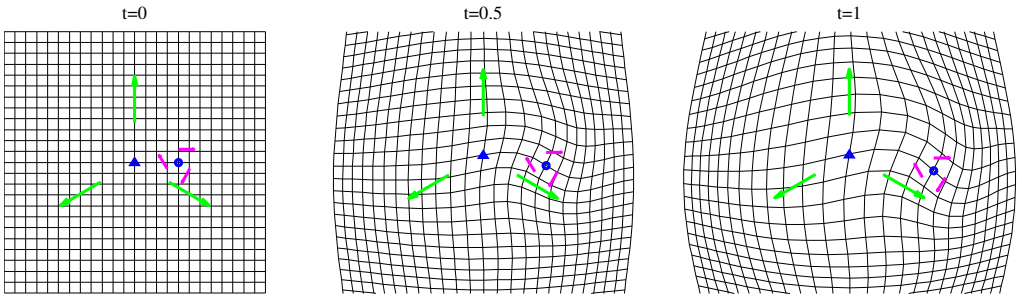


Figure 4.9: Example of a large deformation. Geometrical descriptors are in blue (triangle for scaling, circle for rotation). Vectors are intermediate tools used to build vector fields (magenta for rotation and green for scaling)

geometrical descriptor of the local scaling is almost constant. We represent also in this figure intermediate tools d_j, which are recomputed at each time from the geometrical descriptors (centres of the scaling and rotation) and therefore are not transported by the flow. This example shows how complex modular large deformations can be naturally built from simple base modules, and how their mutual interaction during the integration of the trajectory is encoded in the compound deformation module. We emphasize here that in order to build such a complex deformation module, one only needs to define a deformation module generating a local scaling, a deformation module generating a local rotation, and then to apply the simple combination rules presented in 4.1.1.4.

4.2.3.2 Spreading

We introduce here a deformation module that generates a local uni-directional spreading along a direction. The geometrical descriptors are made of one point z and one vector u and then the shape space $\mathcal{O}$ and the infinitesimal action ξ can be chosen between the four possibilities presented in Section 4.2.1.2. The vector field generated by $o = (z, u)$ and a control $h \in \mathbb{R}$ is a uni-directional spreading along the direction u. In order to build this in practice, we set a scale σ and to a geometrical descriptor $o = (z, u) \in \mathcal{O}$ we associate points $z_i(o, \sigma) \doteq z_i(z, \sigma)$ as defined previously. These points $z_i(o, \sigma)$ carry translations enabling to generate the vector field, and vectors defining these translations are, for each i, $d_i(o, \sigma) \doteq (y_i(o, \sigma)^T u)u$, with $o = (z, u)$ and $y_i(o, \sigma) \doteq z_i(o, \sigma) - z$. Then $\zeta_o(h) = h \sum_i K_\sigma(z_i(o, \sigma), \cdot) d_i(o, \sigma)$, see Figure 4.10 for an example.

4.2.4 Anisotropic deformation

Let us set a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ such that each geometrical descriptor $o \in \mathcal{O}$ is composed of at least one point, *ie* o can be written $o = (z, o^1)$ with $z \in \mathbb{R}^d$. We also supposed that vector fields generated by $o = (z, o^1) \in \mathcal{O}$ are localized at a scale σ in an isotropic area centred at o^1. M can for instance be one of the deformation modules generating one local translation (see Section 4.2.2) or one constrained affine

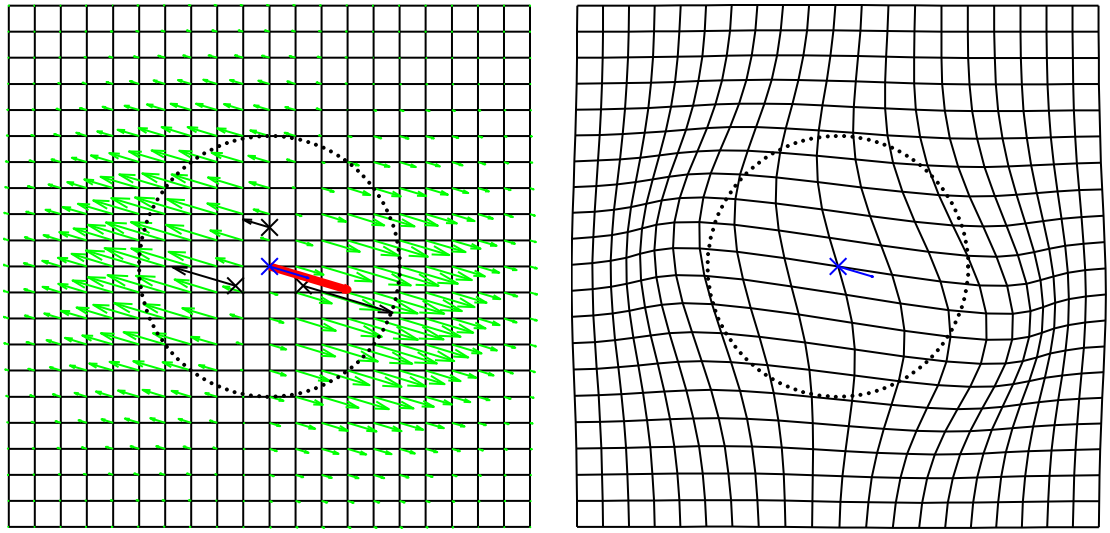


Figure 4.10: Example of vector field generated by the deformation module generating a local spreading at scale σ. Geometrical descriptor (one point and one vector): blue cross for centre of spreading and blue arrow for the direction of spreading; control (one scalar): length and direction of the red arrow; the dotted circle is centred at the centre of spreading and of radius σ; black crosses and black arrows are intermediate tools used to build vector fields (z_i and d_i). **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field.

deformation (see Section 4.2.3). We present here a concrete generic method to build a deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ generating vector fields of the same type and which is localized in an anisotropic area. The geometrical descriptor will be formed of two components, the first belonging to $\mathcal{O}$, and the second made of $d-1$ vectors (with d the dimension of the ambient space) which will define directions of anisotropy: $\tilde{\mathcal{O}} \doteq \mathcal{O} \times (\mathbb{R}^d)^{d-1}$. From a geometrical descriptor $\tilde{o} = (o, u_1, \cdots, u_{d-1}) \in \tilde{\mathcal{O}} = \mathcal{O} \times (\mathbb{R}^d)^{d-1}$ we compute a set of points $\Lambda_{\tilde{o}} \doteq \{z + \sum_{k=1}^{d-1} \sigma j_k u_k \mid \forall k \in [|1, d-1|], -r_k \leq j_k \leq r_k\}$ where $o = (z, o^1)$ and for each k, the *coefficient of anisotropy* r_k is a fixed positive integer. We define $\tilde{H} \doteq H$. We can now define the field generator as the sum of the vector fields that would be created by the $\mathcal{O}$ component of the geometrical descriptor, but centred at points of $\Lambda_{\tilde{o}}$:

$$\zeta : (\tilde{o}, h) \in \tilde{\mathcal{O}} \times \tilde{H} \mapsto \sum_{x \in \Lambda_{\tilde{o}}} \zeta_{(x, o^1)}(h)$$

where $\tilde{o} = (z, o^1, u_1, \cdots, u_{d-1})$. The cost $\tilde{c}$ is defined thanks to the norm of the generated vector field in the RKHS V_σ: $\tilde{c}_{\tilde{o}}(\tilde{h}) \doteq |\tilde{\zeta}_{\tilde{o}}(\tilde{h})|^2_{V_\sigma}$. Last we need to define the infinitesimal action: we set $\xi : (\tilde{o}, v) \in \tilde{\mathcal{O}} \times C^1_0(\mathbb{R}^d) \mapsto (\xi_o(v), \xi^1_{(z,u_1)}(v), \cdots, \xi^{d-1}_{(z,u_{d-1})}(v))$ with $\tilde{o} =$

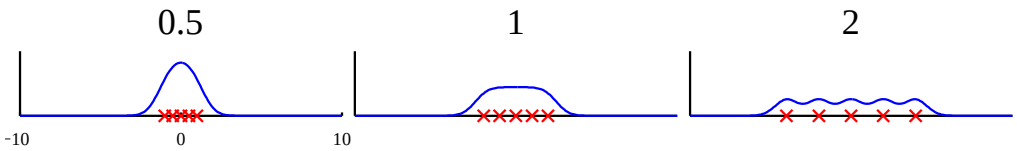


Figure 4.11: Sum of 5 gaussian functions of scale 1, centred at red crosses. Distance between these centres is 0.5 (left), 1 (middle) and 2 (right).

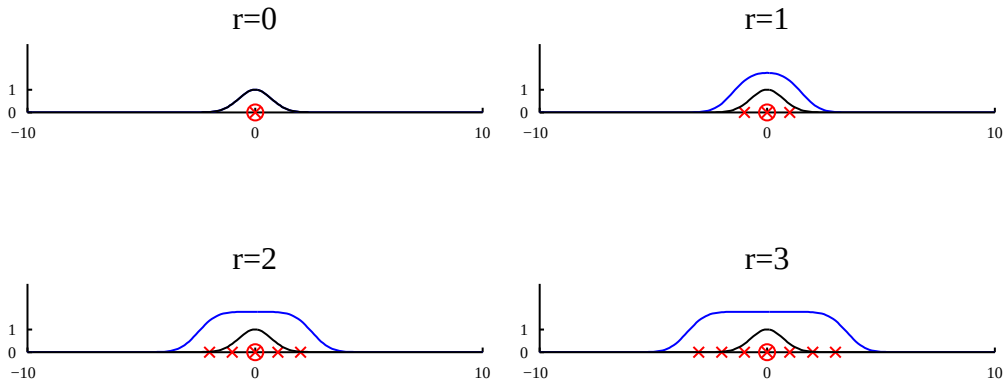


Figure 4.12: Examples of the resulting localisation function in the direction of the anisotropy. In black is plotted the gaussian function centres at the circled point and of scale 1. In blue are plotted the sums of $2r+1$ gaussian functions of scale 1 centred at red crosses (r equals successively 0, 1, 2 and 3). Distance between these centres is 1.

$(o, u_1, \cdots, u_{d-1})$, $o = (z, o^1)$ and for each k, ξ^k one of the four possible infinitesimal actions on couples of one point and one vector (see Section 4.2.1.2). One may want to favour actions which keep the vector or its length constant as the length of vectors u_i plays an important role here. Indeed, in practice the localization of the vector field is often performed thanks to a Gaussian kernel, we present in Figure 4.11 the plot of sums of such Gaussian functions centred at points regularly spaced, for different values of distances between these points. One can see that there are three modes depending on the distance: when it is to small, the sum acts like one Gaussian kernel, on the opposite when it is too big, the resulting sum is different Gaussian kernel. The last mode is the one which we are interested in: in the intermediate range, the sum of kernels results in a new local function of bigger scale. In Figure 4.12 we present the resulting localisation function in the direction of the anisotropy for different values of coefficient of anisotropy r.

We introduce here two examples of this construction in dimension 2 and one in dimension 3.

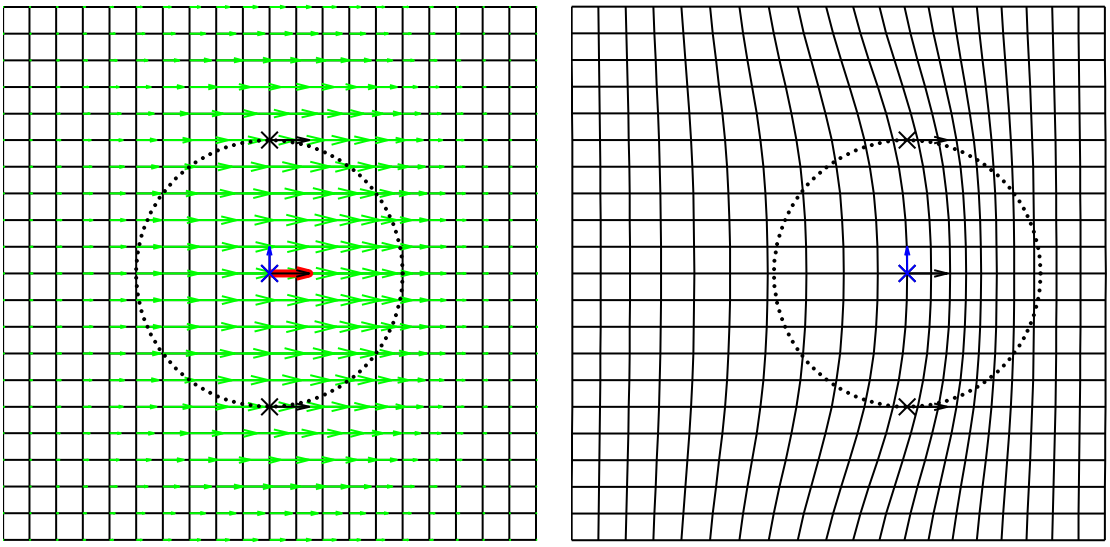


Figure 4.13: Example of vector field generated by the deformation module generating an anisotropique unconstrained local translation at scale σ. Geometrical descriptor: blue cross for centre of spreading and blue arrow for the direction of anisotropy; control: red arrow; the dotted circle is centred at the centre of spreading and of radius σ; intermediate tools used to build the vector fields: black crosses are the points $o+\sigma ju$ and black arrows are equal to the control, attached to these points. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field.

4.2.4.1 Anisotropic local translation in dimension 2

We will present here the construction of the deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ generating an unconstrained local translation localized in an anisotropic area. We first set $M = (\mathcal{O}, H, \zeta, \xi, c)$ the deformation module generating an unconstrained isotropic local translation as defined in Section 4.2.2.1 of scale $\sigma \in \mathbb{R}_+^*$ in dimension 2: $\mathcal{O} = \mathbb{R}^2$, $H = \mathbb{R}^2$, $\zeta : (o,h) \in \mathcal{O} \times H \mapsto K_\sigma(o,\cdot)h$, $\xi : (o,h) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto v(o)$ and $c : (o,h) \in \mathcal{O} \times H \mapsto |h|^2$. Let us set a coefficient of anisotropy $r \in \mathbb{N}$. In this example we will impose that the direction of anisotropy remains steady during the integration of the diffeomorphism: $\tilde{\mathcal{O}} = \mathcal{O} \times \mathbb{R}^2$ and $\xi : (\tilde{o}, v) \in \tilde{\mathcal{O}} \times C_0^\ell(\mathbb{R}^d) \mapsto (\xi_o(v), 0) = (v(o), 0)$ with $\tilde{o} = (o, u)$. The field generator is

$$\tilde{\zeta} : (\tilde{o}, h) \mapsto \sum_{j=-r}^{j=r} K_\sigma(o + \sigma ju, \cdot)h$$

where $\tilde{o} = (o, u)$. See Figure 4.13 for an example of generated vector field for $r = 2$.

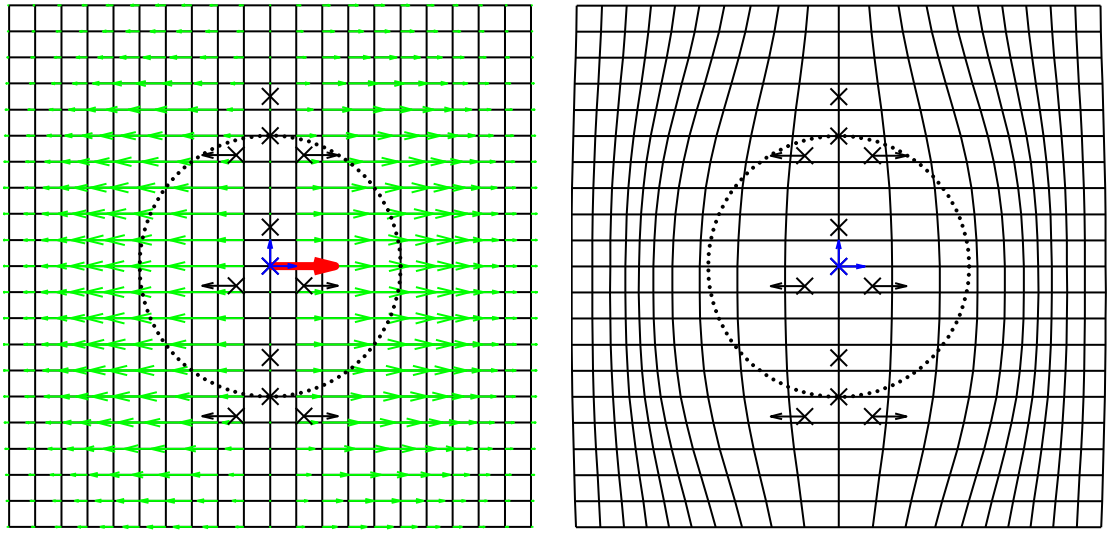


Figure 4.14: Example of vector field generated by the deformation module generating an anisotropique local spreading at scale σ. Geometrical descriptor: blue cross for centre of spreading, horizontal blue arrow for the direction of spreading and vertical blue arrow for the direction of anisotropy; control: length and direction of the red arrow; the dotted circle is centred at the centre of spreading and of radius σ; intermediate tools used to build the vector fields: black crosses the points $z_i(z + \sigma ju, \sigma)$ and black arrows vectors $h(y_i(z+\sigma ju, \sigma)^T w)w$. **Left:** Vector field generated by this geometrical descriptor and this control in green. **Right**: Deformation of the ambient space and geometrical descriptor under the action of the vector field.

4.2.4.2 Anisotropic local spreading in dimension 2

Let us set $\sigma \in \mathbb{R}_+^*$ and $M = (\mathcal{O}, H, \zeta, \xi, c)$ the deformation module generating one local Spreading at scale σ in dimension 2 (see Section 4.2.3.2). We choose the update by the adjoint action for infinitesimal action: $\mathcal{O} = \mathbb{R}^2 \times \mathbb{R}^2$ and $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), -\mathrm{d}v_z^T w)$ where $o = (z, w)$. We set an anisotropic coefficient $r \in \mathbb{N}$, we will now define the deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ generating a local spreading localized in an anisotropic area. We will consider the action of diffeomorphisms on the vector of anisotropy such that its norm remains constant. Then $\tilde{\mathcal{O}} \doteq \mathcal{O} \times (\mathbb{R}^2 - \{0\}) = \mathbb{R}^2 \times \mathbb{R}^2 \times (\mathbb{R}^2 - \{0\})$ and the infinitesimal action is $\tilde{\xi} : (\tilde{o}, v) \in \tilde{\mathcal{O}} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), -\mathrm{d}v_z^T w, \frac{\mathrm{d}v_z \cdot u}{|u|} - \frac{<u, \mathrm{d}v_z \cdot u>}{|u|^3} u)$ where $\tilde{o} = (z, w, u)$. We can define the space of control $\tilde{H} \doteq H = \mathbb{R}$ and the field generator:

$$\tilde{\zeta} : (\tilde{o}, \tilde{h}) \in \tilde{\mathcal{O}} \times \tilde{H} \mapsto \tilde{h} \sum_{j=-r}^{j=r} \sum_{i=1}^{3} K_\sigma(z_i(z + \sigma ju, \sigma), \cdot)(y_i(z + \sigma ju, \sigma)^T w)w$$

where $\tilde{o} = (z, w, u)$ and for any point x, points $z_i(x, \sigma)$ are defined in Section 4.2.3 and $y_i(x, \sigma) \doteq z_i(x, \sigma) - x$. See Figure 4.14 for an example.

4.2.4.3 Anisotropic local translation in dimension 3

In this last example we will present the example of a deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ generating a translation with prior on the direction of translation in dimension 2 and localized in an anisotropic area. Let us first set $\sigma \in \mathbb{R}_+^*$ and $M = (\mathcal{O}, H, \zeta, \xi, c)$ the deformation module generating one local isotropic translation with prior on the direction. We update the vector of translation thanks to differentials: $\mathcal{O} = \mathbb{R}^3 \times \mathbb{R}^3$ and $\xi : (o, v) \in \mathcal{O} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), \mathrm{d}v_z w)$ with $o = (z, w)$. We can now define $\tilde{M}$. As we are now in dimension 3, the anisotropy will be defined thanks to 2 vectors. Let us set r_1 and r_2 two coefficients of anisotropy. We will take the action on vectors of anisotropy so that they remain constant: $\tilde{\mathcal{O}} \doteq \mathcal{O} \times \mathbb{R}^3 \times \mathbb{R}^3$ and $\tilde{\xi} : (\tilde{o}, v) \in \tilde{\mathcal{O}} \times C_0^\ell(\mathbb{R}^d) \mapsto (v(z), \mathrm{d}v_z w, 0, 0)$ with $\tilde{o} = (z, w, u_1, u_2)$. We set $\tilde{H} \doteq H = \mathbb{R}$ and the field generator is then

$$\tilde{\zeta} : (\tilde{o}, \tilde{h}) \in \tilde{\mathcal{O}} \times \tilde{H} \mapsto \tilde{h} \sum_{k_1=-r_1}^{r_1} \sum_{k_2=-r_2}^{r_2} K_\sigma(z + \sigma k_1 u_1 + \sigma k_2 u_2, \cdot) w$$

where $\tilde{o} = (z, w, u_1, u_2)$.

4.2.5 Unconstrained local affine transformations

We present here the deformation module generating vector fields that are locally an affine deformation (at a fixed scale σ), without any other prior on the local deformation pattern. We explain here the construction in dimension 2 without loss of generality. For $o \in \mathbb{R}^2$ we define points $(z_j(o, \sigma))_j$ as in the previous case. We define spaces $\mathcal{O} \doteq \mathbb{R}^2$, $H \doteq (\mathbb{R}^2)^3$ (groups of 3 vectors of $\mathbb{R}^2$) and applications $\zeta : (o, h = (\alpha_j)) \in \mathcal{O} \times H \mapsto \sum_j K_\sigma(z_j(o), \cdot)\alpha_j$, $\xi_o : v \in C_0^\ell(\mathbb{R}^2) \mapsto v(o)$ and $c_o : h = (\alpha_j) \in H \mapsto |\zeta_o(h)|^2_{V_\sigma}$. Then $M = (\mathcal{O}, H, \zeta, \xi, c)$ is a C^k-deformation module that satisfies UEC and generates vector fields that are unconstrained local transformations at scale σ. This example differs from the sum of translation in Section 4.1.1.1 (example 1) as the 3 centres of translations here are glued together. This example differs also from Section 4.2.3 as here the directions d_j are free to generate any local transform. These directions become then control parameters, whereas they were a function of the geometrical descriptor in the previous example. Similarly to previous examples local affine transforms may be combined into a deformation module creating a superimposition of P unconstrained local transforms at different locations. This module class differs from the poly-affine framework [SPR12] in that the neighbourhood which is affected by the local affine transformation is transported by the global deformation via the infinitesimal feedback action ξ.

4.2.6 External force

In some cases it might be useful to consider a deformation module whose geometrical descriptors are attached to a fixed background and then are not transported during the integration of the flow. Any of the previous deformation modules can be adapted to this case by setting the infinitesimal action ξ to zero.

4.2.7 Boundary motion (image geometrical descriptor)

We give here an example of deformation module where geometrical descriptors are not only geometrical shapes but composed of N landmarks and an image (N being fixed). In the deformation module that we present here, landmarks are a discretisation of an area of the image but one could define them as points of interest. The generated vector field will be defined thanks to the gradient of the image at this points, allowing motion orthogonal to boundaries. We define $\mathcal{O} = (\mathbb{R}^2)^N \times C^\infty(\mathbb{R}^2, \mathbb{R})$, it is a shape space associated with the following action of diffeomorphisms: for $\phi \in \mathrm{Diff}_0^\infty(\mathbb{R}^d)$ and $o = (x_1, \cdots, x_N, I) \in \mathcal{O}$, $\phi \cdot o = (\phi(x_1), \cdots, \phi(x_N), I \circ \phi^{-1})$. Then the infinitesimal action is given by $\xi_o : v \in C_0^\infty(\mathbb{R}^2) \mapsto (v(x_1), \cdots, v(x_N), -\mathrm{d}I(v))$ where $o = (x_1, \cdots, x_N, I) \in \mathcal{O}$. We set a scale $\sigma > 0$ and we define $H = \mathbb{R}$ and

$$\zeta : (o, h) \in \mathcal{O} \times H \mapsto \sum_{i=1}^{N} hK_\sigma(x_i, \cdot)\nabla I(x_i)$$

with $o = (x_1, \cdots, c_N, I)$. The cost is defined by $c : (o, h) \in \mathcal{O} \times H \mapsto |\zeta_o(h)|^2_{V_\sigma}$. In Figure 4.15 we present an example of geometrical descriptor for this deformation module (we fix $N = 28$), as well as its transport by the vector field it generates itself (for a control h equal to 3).

4.2.8 Silent modules

We present here a last example of deformation modules: modules generating a vector field that is always null. For a choice a C^k-shape space $\mathcal{O}$ of order ℓ with infinitesimal action ξ, we set $H \doteq \{0\}$ (null space for the controls) and for $o \in \mathcal{O}$, $\zeta_o : h \in H \mapsto 0$, $c_o : h \in H \mapsto 0$. Then $M = (\mathcal{O}, H, \zeta, \xi, c)$ defines a C^k-deformation module of order ℓ satisfying UEC, which will be referred to as **the silent deformation module induced by shape space** $\mathcal{O}$. For instance, if $\mathcal{O}$ is the shape space made of a collection of points as in the examples of landmarks (Section 4.1.1.1) and we combine M with other (active) deformation modules, then points of $\mathcal{O}$ will feel the velocity field generated by active modules located around them, and move accordingly, but will not contribute to this velocity field. The introduction of silent modules is necessary if one wants to use active deformation modules whose geometric descriptors do not derive from the input shape data, for instance for scaling module whose center is not forced to be a vertex or point of the input shape data, or for the direction of a translation which is not forced to be normal to a surface mesh.

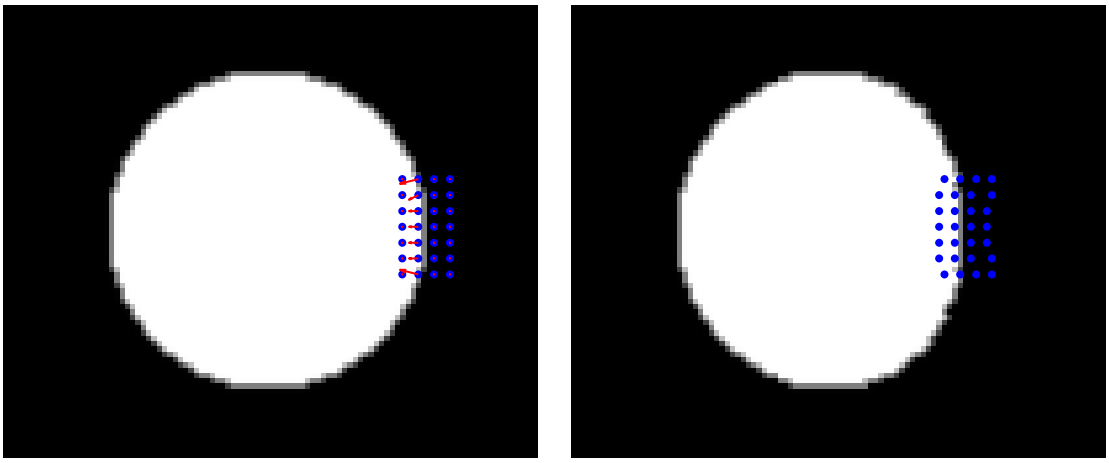


Figure 4.15: Example of geometrical descriptor and control of deformation module generating boundary motion. **Left:** The geometrical descriptor is composed of the image and the blue points representing landmarks x_i. In red are the vectors $h\nabla I(x_i)$ generating the vector field $\zeta_o(h)$ (here $h = 3$). **Right**: Deformation of the geometrical descriptor (image and landmarks) under the action of the vector field they generate.

Chapter 5

From a geometrical model to the study of shapes

Contents

5.1 Sub-Riemannian setting

We explain here how modular large deformations may be used to provide shape spaces with a sub-Riemannian metric, and therefore define a distance between shapes.

We consider a C^k-deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ of order $\ell \in \mathbb{N}^*$. A geometrical descriptor of the module is a "shape" in the shape space $\mathcal{O}$ and, intuitively, the orbit of this geometrical descriptor under the action of a regular group of diffeomorphisms forms a Riemannian manifold. If one considers only the diffeomorphisms which result from the integration of a modular velocity field (generated by the module M), then one provides this Riemannian manifold with a sub-Riemannian structure (Section 5.1.1). This construction allows the definition of a sub-Riemannian distance and of optimal trajectories between two geometrical descriptors in the shape space $\mathcal{O}$ (Section 5.1.2).

5.1.1 A sub-Riemannian structure on $\mathcal{O}$

We suppose that M satisfies UEC, and we set a space of vector fields V and a constant $C > 0$ such that V is continuously embedded in $C_0^{\ell+k}(\mathbb{R}^d)$ and for all o in $\mathcal{O}$, for all h in H, $\zeta_o(h)$ is in V and $|\zeta_o(h)|_V^2 \leq C c_o(h)$. We use now the notion of continuous sub-Riemannian structure on a manifold, following the definition given in [Arg14].

Definition 22. *Let $\mathcal{M}$ be a manifold of finite dimension. We define a* ***continuous sub-Riemannian structure on*** *$\mathcal{M}$ as a triple $(\mathcal{E}, g, \rho)$, where $\mathcal{E} \to \mathcal{M}$ is a smooth vector bundle on $\mathcal{M}$ endowed with a smooth, Riemannian metric g, and $\rho : \mathcal{E} \to TM$ is a continuous vector bundle morphism.*

The composition of the field generator ζ with the infinitesimal action ξ yields a continuous vector bundle morphism $\rho : (o, h) \in \mathcal{O} \times H \mapsto (o, \rho_o(h) = \xi_o \circ \zeta_o(h)) \in T\mathcal{O}$. Moreover, the cost c induces a smooth Riemannian metric g on the vector bundle $\mathcal{O} \times H$. Then $(\mathcal{O} \times H, g, \rho)$ defines a continuous sub-Riemannian structure on $\mathcal{O}$, which we will denote $\mathcal{O}_H$. This structure is the key to define trajectories of modular deformations. At each shape $o \in \mathcal{O}$ is attached the space $\zeta_o(H)$ of vector fields that can be generated by o. The **horizontal space** $\rho_o(H) = \xi_o(\zeta_o(H))$ is the set of tangent vectors of $T_o\mathcal{O}$ that can be obtained by the action of the geometrical descriptor o on itself.

Remark 16. *Note that the dimension of the horizontal space at $o \in \mathcal{O}$, namely the rank of $\rho_o(H)$, may depend on o. For instance, let us consider the deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ obtained by combining a module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ generating a local scaling in $\mathbb{R}^2$ at scale σ (see Section 4.1.1.2.2, geometrical descriptors are points, controls are scalars) and a module $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ generating a sum of three local translations in $\mathbb{R}^2$ at scale σ (see Section 4.1.1.2.1, geometrical descriptors are triplets of points, controls are triplets of vectors). For each geometrical descriptor $o = (o^1, o^2 = (z_i)_{1\leq i\leq 3}) \in \mathcal{O}^1 \times \mathcal{O}^2 = \mathbb{R}^2 \times (\mathbb{R}^2)^3$ and each control $h = (h^1, h^2 = (\alpha_i)_{1\leq i\leq 3}) \in H^1 \times H^2 = \mathbb{R} \times (\mathbb{R}^2)^3$, the generated vector field is $\zeta(o, h) = h^1 \sum_j K(z_j(o^1), \cdot) d_j(o^1) + \sum_i K(z_i, \cdot)\alpha_i$ with $z_j(o^1)$ and $d_j(o^1)$ defined in Section 4.1.1.2.2. Then in the particular case where o^1 and $o^2 = (z_i)_i$ are such that $z_i = z_i(o^1)$ for $i \in \{1, 2, 3\}$, $\dim(\xi_o(\zeta_o(H))) = 2 \times 3 = 6$ while in other cases $\dim(\xi_o(\zeta_o(H))) = 2 \times 3 + 1 = 7$.*

The cost c equips the vector bundle $\mathcal{O} \times H$ with a metric g, which may be used to derive a **metric $g^{\mathcal{O}}$ on $\mathcal{O}_H$**: if $\delta o,\ \delta o' \in \rho_o(H)$, let $h,\ h' \in Ker(\rho_o)^{\perp}$ (orthogonal for the metric g_o on H) such that $\rho_o(h) = \delta o$ and $\rho_o(h') = \delta o'$, then $g_o^{\mathcal{O}}(\delta o, \delta o') \doteq g_o(h, h')$. Note that for $\delta o \in \rho_o(H)$, $|\delta o|_o^2 \doteq g_o^{\mathcal{O}}(\delta o, \delta o) = \inf\{c_o(h) \mid \rho_o(h) = \delta o,\ h \in H\}$. This sub-Riemannian metric $g^{\mathcal{O}}$ will be the one used to build the new distance on $\mathcal{O}$.

We will now present some definitions and results given in [AB14], which allow the definition of a sub-Riemannian distance on $\mathcal{O}$.

Definition 23. *[AB14] Let $\mathcal{M}$ be a smooth manifold of finite dimension equipped with a continuous sub-Riemannian structure $(\mathcal{E}, g, \rho)$. A **horizontal system** is a curve $t \in [0,1] \mapsto (q(t), u(t)) \in \mathcal{E}$ such that $t \in [0,1] \mapsto u(t) \in \mathcal{E}_{q(t)}$ is of class L^2 (ie $\int_0^1 g_{q(t)}(u(t), u(t))\mathrm{d}t < \infty$), and its projection $t \mapsto q(t)$ is absolutely continuous and satisfies for almost every $t \in [0,1]$, $\dot{q}(t) = \rho_{q(t)}u(t)$. A **horizontal curve** is the projection q to $\mathcal{M}$ of a horizontal system (q, u).*

Remark 17. *For $a, b \in \mathcal{O}$, space $\Omega_{a,b}$ (see Definition 21) is exactly the set of horizontal system connecting a and b.*

Definition 24. *[AB14] Let $o : [0,1] \longrightarrow \mathcal{O}$ be a horizontal curve, we define its **sub-Riemannian length**:*

$$l(o) = \int_0^1 |\dot{o}_t|_{o_t}\mathrm{d}t = \int_0^1 \sqrt{g_{o_t}^{\mathcal{O}}(\dot{o}_t, \dot{o}_t)}\mathrm{d}t.$$

*For $a, b \in \mathcal{O}$ we can then define the **sub-Riemannian distance***

$$D(a,b) = inf\{l(o) \mid \exists h : (o,h) \in \Omega_{a,b}\}.$$

Lemma 3. *[AB14] Let o be a horizontal curve of positive length. There exists a horizontal curve $\tilde{o}$ and a Lipschitz bijective reparametrization $\gamma : [0,1] \longrightarrow [0,1]$ such that $\tilde{o} = o \circ \gamma$ and for almost every t, $|\dot{\tilde{o}}(t)|_{\tilde{o}(t)} = l(o)$.*

Definition 25. *[AB14] Let $a, b \in \mathcal{O}$ and $(o,h) \in \Omega_{a,b}$. We define the **length of** (o,h) by*

$$l(o,h) = \int_0^1 \sqrt{c_{o_t}(h_t)}\mathrm{d}t$$

and we set

$$D_H(a,b) = \inf\{l(o,h) \mid (o,h) \in \Omega_{a,b}\}.$$

Remark 18. *If $\Omega_{a,b}$ is empty, both $D(a,b)$ and $D_H(a,b)$ have an infinite value.*

We will now show that studying D and D_H amounts to considering same trajectories, and that D is a real distance on $\mathcal{O}$.

Definition 26. *[AB14] Let $o : [0,1] \longrightarrow \mathcal{O}$ be a horizontal curve, for each $t \in [0,1]$, let $h^\star(t) \in H$ be the only element of H such that $c_o(h^\star(t)) = |\dot{o}_t|_o^2$. We say that $h^\star : t \mapsto h^\star(t)$ is the **minimal control associated with** o.*

Lemma 4. *[AB14] Let $o : [0,1] \longrightarrow \mathcal{O}$ be a horizontal curve, its minimal control $h^\star$ is measurable and of class L^2.*

Thanks to the previous lemma, we can deduce the following proposition:

Proposition 21. *[AB14] Let $a, b \in \mathcal{O}$, $D(a,b) = D_H(a,b)$.*

We can now prove that D defines a distance on $\mathcal{O}$.

Proposition 22. *As M satisfies UEC, the sub-Riemanian distance D is a true distance on $\mathcal{O}$.*

Proof. It is clear that D is a pseudo-distance, we need to show that if $D(a,b) = 0$ then $a = b$. We will use a result proved in [ATTY15]: if we set for $\phi \in \mathrm{Diff}_0^\ell(\mathbb{R}^d)$, $d(Id, \phi) \doteq \inf\{\int_0^1 |v_t|_V \mathrm{d}t \mid \forall t\ v_t \in V \text{ and } \varphi_{t=1}^v = \phi\}$ (we remind that V is the Hilbert space of vector fields defined UEC) and for $a, b \in \mathcal{O}$, $dist(a,b) = \inf\{d(Id,\phi) \mid \phi \cdot a = b\}$ then $dist$ defines a distance (taking its value in $[0, +\infty]$).
Let $a, b \in \mathcal{O}$ such that $D(a,b) = 0$, then there exists $(o^n)_{n\in\mathbb{N}}$ such that for each n there exists h^n such that $(o^n, h^n) \in \Omega_{a,b}$ and $l(o^n) \longrightarrow 0$. By choosing h^n the minimal control of o^n, we also have $l(o^n, h^n) \longrightarrow 0$. Yet, thanks to the UEC, for each n, $dist(a,b) \leq \int_0^1 |\zeta_{o^n}(h^n)|_V \leq \sqrt{C}\int_0^1 \sqrt{c_{o^n}(h^n)} = \sqrt{C} l(o^n, h^n) \longrightarrow 0$. Then $a = b$.

□

5.1.2 Optimal trajectories

Thanks to Proposition 21, the minimum of l in $\Omega_{a,b}$ (for $a, b \in \mathcal{O}$) is equal to the distance between a and b. However the quantity $l(o,h) = \int_0^1 \sqrt{c_o(h)}$ is hard to study and then it is necessary to link it with the energy $E(o,h) = \int_0^1 c_o(h)$.

Proposition 23. *[AB14] Let $a, b \in \mathcal{O}$ and let (o,h) be in $\Omega_{a,b}$. Then (o,h) minimizes E in $\Omega_{a,b}$ if and only if it minimizes l in $\Omega_{a,b}$ and its cost $c_o(h)$ is constant.*

Remark 19. *We deduce that along minimizers (if they exist), $D_H(a,b)^2 = l(o,h)^2 = E(o,h)$.*

Propositions 21 and 23 show that calculating the distance $D(a,b)$ between two elements a, b of $\mathcal{O}$ amounts to looking for horizontal systems minimizing the energy E, which is easiest to study. Therefore, in the next paragraph we characterize horizontal systems minimizing E. We need here to restrict ourselves to a certain type of shapes, obtained through an adaptation of a definition given by S. Arguillère in [ATTY15].

Definition 27. *An element o of $\mathcal{O}$, is said to be of **compact support** if there exists a compact set K of $\mathbb{R}^d$ such that for all $\phi \in D_0^\ell(\mathbb{R}^d)$, $\phi \cdot o$ only depends on $\phi_{|K}$ and $\phi \in D_K^\ell \mapsto \phi \cdot o$ is continuous with $D_K^\ell := \{\phi_{|K} \mid \phi \in D_0^\ell(\mathbb{R}^d)\}$ equipped with the distance deduced from the norm on $\{v_{|K} \mid v \in C_0^\ell(\mathbb{R}^d)\}$: $|v|_{\ell,K} = \sup\{|\frac{\partial^{\ell_1+\cdots+\ell_d} v(x)}{\partial x_1^{\ell_1}\cdots x_d^{\ell_d}}| \mid x \in K, (\ell_1, \cdots, \ell_d) \in \mathbb{N}^d, \ell_1 + \cdot + \ell_d \leq \ell\}$.*

Remark 20. *Examples presented in Section 4.2 correspond to geometrical descriptors of compact support.*

Lemma 5. *Let $o \in \mathcal{O}$ be of compact support and let K be a compact set of $\mathbb{R}^d$ such that for all ϕ of $D_0^\ell(\mathbb{R}^d)$ $\phi \cdot o$ only depends on K and $\phi \in D_K^\ell \mapsto \phi \cdot o$ is continuous. Then for each $\phi \in D_0^\ell(\mathbb{R}^d)$, $\phi \cdot o$ is of compact support and $\psi \in D_0^\ell(\mathbb{R}^d) \mapsto \psi \cdot (\phi \cdot o)$ only depends on the compact set $\phi(K)$.*

Proof. Let ϕ, ψ and ψ' be elements of $D_0^\ell(\mathbb{R}^d)$ such that $\psi_{|\phi(K)} = \psi'_{|\phi(K)}$. Then $\psi \cdot (\phi \cdot o) = (\psi \circ \phi) \cdot o = (\psi' \circ \phi) \cdot o$ because $\psi \circ \phi_{|K} = \psi' \circ \phi_{|K}$. Then $\psi \cdot (\phi \cdot o) = \psi' \cdot (\phi \cdot o)$ and we conclude that for all $\psi \in D_0^\ell(\mathbb{R}^d)$, $\psi \cdot (\phi \cdot o)$ only depends on $\phi(K)$. Besides from Faà di Bruno's formula one gets that for ϕ an element of $D_0^\ell(\mathbb{R}^d)$, there exists a constant $C_{|\phi|_{\ell,K}}$ such that for each ψ, ψ' in $D_0^\ell(\mathbb{R}^d)$, $|\psi \circ \phi - \psi' \circ \phi|_{\ell,K} \leq C_{|\phi|_{\ell,K}} |\psi - \psi'|_{\ell,\phi(K)}$. Then for $\phi \in D_0^\ell(\mathbb{R}^d)$, $\psi \in D_0^\ell(\mathbb{R}^d) \mapsto \psi \cdot (\phi \cdot o)$ is continuous and therefore $\phi \cdot o$ is of compact support.

□

Let a and b be two geometrical descriptors of compact support, we prove here the existence of minimizers of the energy defined in Definition 21 (if $\Omega_{a,b}$ is non-empty), and therefore of trajectories reaching sub-Riemannian distance D.

Theorem 5. *We recall that the deformation module M satisfies UEC. If $\Omega_{a,b}$ is non-empty, the energy E reaches its minimum on $\Omega_{a,b}$.*

Proof. Let (o^n, h^n) be a minimizing sequence of E in $\Omega_{a,b}$ and let, for each n, φ^n be the flow associated to (o^n, h^n) as defined in Proposition 20: $\varphi^n = \varphi^{v_n}$ with $v_n = \zeta_{o^n}(h^n)$. Since $\int_0^1 |v_t^n|_V^2 \mathrm{d}t \leq CE(o^n, h^n)$ (from UEC), the sequence $(v_n)_n$ is bounded in $L^2([0,1], \mathbb{R}^d)$ so up to the extraction of a subsequence we can assume that v^n converges weakly to $v^\infty \in L^2([0,1], \mathbb{R}^d)$. Let us define φ^∞ the flow of v^∞. As a is of compact support, there exists K compact of $\mathbb{R}^d$ such that for all $\phi \in D_0^\ell(\mathbb{R}^d)$, $\phi \cdot a$ only depends on $\phi_{|K}$. Moreover, as K is compact, [Gla05] shows that $\sup_{(t,x)\in[0,1]\times K} |\varphi_t^n(x) - \varphi_t^\infty(x)| \longrightarrow 0$ so that, as $\phi \in D_K^\ell \mapsto \phi \cdot a$ is continuous, $o_\cdot^n = \phi_\cdot^n \cdot a$ converges to $o_\cdot^\infty = \phi_\cdot^\infty \cdot a$ uniformly on $[0,1]$. Therefore there exists a compact set L of $\mathcal{O}$ such that for all t, $o_t^\infty \in L$ and for all n, $o_t^n \in L$. Then $\sup_{o\in L} ||c_o^{-1}||$ (see proof of Proposition 20) is finite and for each n: $\int_0^1 |h^n(t)|_H^2 \mathrm{d}t \leq \sup_{o\in L} ||c_o^{-1}|| E(o^n, h^n)$. Therefore h^n is bounded in $L^2([0,1], H)$ so up to the extraction of a subsequence we can assume that h^n converges weakly to $h^\infty \in L^2([0,1], H)$. Let us show that $(o^\infty, h^\infty) \in \Omega_{a,b}$. Let $w \in L^2([0,1], V)$, we have

$$\begin{aligned} |\int_0^1 \langle v_t^\infty - \zeta_{o_t^\infty}(h_t^\infty), w_t\rangle_V \mathrm{d}t| \quad \leq \quad & |\int_0 \langle v_t^\infty - \zeta_{o_t^n}(h_t^n), w_t\rangle_V \mathrm{d}t| \\ & + |\int_0^1 \langle \zeta_{o_t^n}(h_t^n) - \zeta_{o_t^\infty}(h_t^n), w_t\rangle_V \mathrm{d}t| \\ & + |\int_0^1 \langle \zeta_{o_t^\infty}(h_t^n) - \zeta_{o_t^\infty}(h_t^\infty), w_t\rangle_V \mathrm{d}t. \end{aligned}$$

As $\zeta_{o^n}(h^n)$ converges weakly to v^∞ the first term tends to 0. In the same way, $h \in L^2([0,1], H) \mapsto \int_0^1 \langle \zeta_{o_t^\infty}(h_t), w_t\rangle_V \mathrm{d}t$ is continuous since, as ζ is of class at least C^1 with respect to o, $o \in \mathcal{O} \mapsto \zeta_o \in L(H, V)$ is bounded on L (which contains o_t^∞ for all t). Then as h^n weakly converges to h^∞, $\int_0^1 \langle \zeta_{o_t^\infty}(h_t^n) - \zeta_{o_t^\infty}(h_t^\infty), w_t\rangle_V \mathrm{d}t$ tends to 0. Therefore

$$\begin{aligned} |\int_0^1 \langle v_t^\infty - \zeta_{o_t^\infty}(h_t^\infty), w_t\rangle_V \mathrm{d}t| \quad & \leq \quad \limsup |\int_0^1 \langle (\zeta_{o_t^n} - \zeta_{o_t^\infty})(h_t^n), w_t\rangle \mathrm{d}t| \\ & \leq \quad \limsup (\int_0 |w_t|_V^2 \mathrm{d}t \int_0^1 ||\zeta_{o_t^n} - \zeta_{o_t^\infty}||^2 |h_t^n|_H^2 \mathrm{d}t)^{1/2} \\ & = \quad 0 \end{aligned}$$

since h^n is bounded in $L^2([0,1],H)$ and, as o^n uniformly converges to o^∞, $||\zeta_{o_t^n} - \zeta_{o_t^\infty}|| \longrightarrow 0$. Since w is arbitrary, $v^\infty = \zeta_{o^\infty}(h^\infty)$ and $\dot{o}_t^\infty = \xi_{o_t^\infty}(v_t^\infty)$ so that $(o^\infty, h^\infty) \in \Omega_{a,b}$. We now need to show that $E(o^\infty, h^\infty) = \lim E(o^n, h^n)$. Since $h \mapsto \int_0^1 c_{o_t^\infty}(h)\mathrm{d}t$ is continuous and convex $\int_0^1 c_{o_t^\infty}(h_t^\infty)\mathrm{d}t \leq \liminf \int_0^1 c_{o_t^\infty}(h_t^n)\mathrm{d}t$. Moreover since c is a continuous metric, there exists $C : o \mapsto C_o$ continuous such that $c_o(h) = (C_o h|h)_{H^\star}$, so that $|\int_0^1 (c_{o_t^\infty}(h_t^n) - c_{o_t^n}(h_t^n))\mathrm{d}t| \leq (\sup_t ||C_{o_t^\infty} - C_{o_t^n}||) \int_0^1 |h_t^n|_H^2 \mathrm{d}t \longrightarrow 0$. Then we obtain $\int_0^1 c_{o_t^\infty}(h_t^\infty)\mathrm{d}t \leq \liminf \int_0^1 c_{o_t^n}(h_t^n)\mathrm{d}t$ and $E(o^\infty, h^\infty) \leq \liminf E(o^n, h^n) = \lim E(o^n, h^n)$. We conclude that $E(o^\infty, h^\infty) = \lim E(o^n, h^n)$ since (o^∞, h^∞) belongs to $\Omega_{a,b}$. □

5.2 Modular comparison of shapes

In this section we study the computation of optimal trajectories in practice, by studying the *inexact matching problem* between two shapes of a common shape space. In the first section we will present the case where these two shapes are in the shape space of geometrical descriptors of a deformation module. In the general case (which will be studied in Section 5.2.2), M is a compound module built from a silent module induced by the shape space of input shape data and user-defined deformation modules.

5.2.1 A first matching problem

We set a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfying UEC and we will now explain how optimal trajectories can be computed. Let a and b be two shapes of $\mathcal{O}$ of compact support. We consider the practical case where the target shape b does not derive from the source shape a by the action of a modular diffeomorphism. We propose to find the "optimal" shape $\hat{b}$ in the orbit of a, i.e. such that $\hat{b} = \phi.a$, so that the deformed shape $\hat{b} = \phi.a$ falls as close as possible to b in the sense of a measure μ which is supposed given with $\mathcal{O}$ (it is typically the Euclidean distance when $\mathcal{O}$ is formed of landmarks or can be more sophisticated in case of currents or varifolds). Simultaneously we derive the differential equations to compute the optimal trajectory between a and $\hat{b}$.

This construction may be seen from a statistical point of view using the following generative model: we suppose that the probability density function (with respect to a given reference measure) of the transformation $\hat{b}$ of a through a modular large deformation is proportional to $\exp(-D(a,\hat{b})^2/\sigma_0^2)$ where D is the sub-Riemannian distance built on $\mathcal{O}$ (see Section 5.1). We also suppose that, the density of the conditionnal distribution of b given $\hat{b}$ is proportional to $\exp -(\mu(\hat{b},b)/\sigma_1^2)$. Eventually, we want to estimate the most probable $\hat{b}$, given a and b: the likelihood of $\hat{b}$ knowing a and b is proportional to $\exp(-D(a,\hat{b})^2/\sigma_0^2) \times \exp(-\mu(\hat{b},b)/\sigma_1^2)$ and its maximisation amounts to the minimisation of:

$$\mu(\hat{b},b)/\sigma_1^2 + D(a,\hat{b})^2/\sigma_0^2 \tag{5.1}$$

As shown in Theorem 5, for each $\hat{b}$ such that $D(a,\hat{b})$ is finite, there exists a trajectory $h^\star$ of controls such that $D(a,\hat{b}) = \int \sqrt{c_o(h^\star)}$, with o satisfying $o_{t=0} = a$ and $\dot{o} = \xi_o \circ \zeta_o(h^\star)$.

Then minimizing (5.1) with respect to $\hat{b}$ amounts to minimizing the following quantity with respect to h

$$\frac{1}{\sigma_1^2}\mu(o_{t=1}, b) + \frac{1}{\sigma_0^2}\Big(\int \sqrt{c_{o_t}(h_t)}dt\Big)^2 \tag{5.2}$$

where $o_{t=0} = a$ and $\dot{o} = \xi_o \circ \zeta_o(h)$.

Besides, thanks to Proposition 23, we know that along minimizing trajectories one gets $\Big(\int \sqrt{c_{o_t}(h_t)}dt\Big)^2 = \int c_{o_t}(h_t)dt$. Then maximizing (5.2) amounts to minimizing the following quantity with respect to h

$$J_{a,b}(h) = \frac{1}{\sigma^2}\mu(o_{t=1}, b) + \int c_{o_t}(h_t)dt \tag{5.3}$$

with $\sigma = \frac{\sigma_1}{\sigma_0}$, $o_{t=0} = a$ and $\dot{o} = \xi_o \circ \zeta_o(h)$. The first term of this sum will be referred to as the **Data term** while the second one will be called the **Regularity term**.

A trajectory o of $\mathcal{O}$ starting at a such that there exists $h \in L^2([0,1], H)$ so that (o, h) is a horizontal system (see Definition 23) and h minimizes $J_{a,b}$, will be called an **optimal trajectory starting at** a, or a **geodesic**. These trajectories can be well characterized thanks to the next result, which we prove following the idea of the proof of [Arg14].

Theorem 6. *We recall that $M = (\mathcal{O}, H, \zeta, \xi, c)$ is a C^k-deformation module of order C^ℓ satisfying UEC. We suppose $k, l \geq 2$ and that μ is C^1. If $h \in L^2([0,1], H)$ minimizes functional $J_{a,b}$ then, with $o : [0,1] \longrightarrow \mathcal{O}$ starting at a and satisfying $\dot{o} = \xi_o \circ \zeta_o(h)$, there exists a path $\eta : t \in [0,1] \longrightarrow \eta_t \in T^*_{o_t}\mathcal{O}$ such that with*

$$\mathcal{H} : (o, \eta, h) \in T^*\mathcal{O} \times H \mapsto \Big(\eta | \xi_o(\zeta_o(h))\Big) - \frac{1}{2}c_o(h)$$

*the **Hamiltonian** of the system, $\eta_{t=1} = -\partial_1\mu(o_{t=1}, b)$ and (in a local chart)*

$$\left\{\begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \xi_o \circ \zeta_o(h) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o} \\ \frac{\partial \mathcal{H}}{\partial h} & = & 0 \end{array}\right. \tag{5.4}$$

Proof. In this proof we suppose that $\mathcal{O}$ is an open subset of $\mathbb{R}^N$. As previously we associate to each $h \in L^2([0,1], H)$ the absolutely continuous trajectory o^h of $\mathcal{O}$ such that $o_{t=0} = a$ and $\dot{o}^h_t = \xi_{o_t} \circ \zeta_{o_t}(h_t)$. For each $h \in L^2([0,1], H)$, o^h is absolutely continuous on $[0,1]$ and then belongs to $H^1_a([0,1], \mathcal{O})$ (elements in $H^1([0,1], \mathcal{O})$ starting at a). We define the new functional $\tilde{J} : (o, h) \in H^1([0,1], \mathcal{O}) \times L^2([0,1], H) \mapsto \int_0^1 c_o(h) + \mu(o_{t=1}, b)$. Then $h \in L^2([0,1], H)$ minimizes J if and only if (o^h, h) minimizes $\tilde{J}$ under the constraint $0 = \Gamma(o, h) \doteq \dot{o} - \xi_o \circ \zeta_o(h)$. Functions $\tilde{J}$ and Γ are of class C^1 and $\partial_o\Gamma$ is an isomorphism for each $(o, h) \in H^1([0,1], \mathcal{O}) \times L^2([0,1], H)$. Indeed let $o \in H^1([0,1], \mathcal{O})$, $h \in L^2([0,1], H)$ and $\alpha \in L^2([0,1], \mathbb{R}^N)$, we can define $\delta o \in H^1_0([0,1], \mathbb{R}^N)$ by $\delta o(t = 0) = 0$ and $\dot{\delta o} = \partial_o(\xi_o \circ \zeta_o(h)) + \alpha$ (the solution is well defined). Then $\alpha = \partial_o\Gamma(o, h) \cdot \delta o$ and therefore $\partial_o\Gamma(o, h)$ is surjective. Moreover $\partial_o\Gamma(o, h)$ in injective as for $\delta o \in H^1_0([0,1], \mathbb{R}^N)$, $\partial_o\Gamma(o, h)\delta o = 0$ implies $\delta o = 0$ by Cauchy uniqueness.

We therefore deduce (thanks to the implicit function theorem) that $\Gamma^{-1}(\{0\})$ is a manifold.
Let (o,h) be a minimizer of $\tilde{J}$ over the set $\Gamma^{-1}(\{0\})$, from [Kur76] (Theorem 4.1) can be shown that there exists a non trivial Lagrange multiplier $\eta \in L^2(0,1,\mathbb{R}^N)^* = L^2(0,1,\mathbb{R}^{N,*})$ such that $\mathrm{d}\tilde{J}_{(o,h)} + (d\Gamma_{(o,h)})^\star(\eta) = 0$. It is shown in [ATTY15] that $\dot{\eta} = -\partial_o \mathcal{H}$ and $\partial_h \mathcal{H} = 0$. □

Remark 21. *As c_o is positive definite, there exists an invertible symmetric operator $C : o \in \mathcal{O} \mapsto C_o \in L(H, H^*)$ such that for all $(o,h) \in \mathcal{O} \times H$, $c_o(h) = (C_o h|h)$. Then the third equality in (5.4) allows to compute h: $h = C_o^{-1}\rho_o^*\eta$ with $\rho_o = \xi_o \circ \zeta_o$ and ρ_o^* such that $(\eta|\rho_o(h))_{T_o^*\mathcal{O}} = (\rho_o^*\eta|h)_H$.*

We define the **reduced Hamiltonian**

$$\mathcal{H}_r(o,\eta) \doteq \mathcal{H}(o,\eta,C_o^{-1}\rho_o^*(\eta)) = \frac{1}{2}(\rho_o^*(\eta)|C_o^{-1}\rho_o^*(\eta)) = \frac{1}{2}c_o(C_o^{-1}\rho_o^*(\eta)) \tag{5.5}$$

and as $\nabla_h \mathcal{H} = 0$, the system of equations (5.4) can be written:

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \frac{\partial \mathcal{H}_r}{\partial \eta} = \rho_o(C_o^{-1}\rho_o^*(\eta)) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}_r}{\partial o} \end{array} \right. \tag{5.6}$$

Proposition 24. *If the module M is C^j of order ℓ with $j, \ell \geq 2$ then solutions of Equation (5.6) exist for any $(o_{t=0}, \eta_{t=0})$ and are totally defined by these initial values.*

Proof. In this case $(o,\eta) \mapsto H_r(o,\eta)$ is of class at least C^2 so $(o,\eta) \mapsto (\frac{\partial \mathcal{H}_r}{\partial \eta}, -\frac{\partial \mathcal{H}_r}{\partial o})$ is at least C^1.

□

Then by choosing an initial momentum $\eta \in T_o^*\mathcal{O}$ one can generate an optimal trajectory starting at a. Optimal trajectories are parametrized by initial values of geometrical descriptor and momentum, so in dimension $2 \times \dim(\mathcal{O})$.

Remark 22. *From Equation (5.4) we re-deduce that cost $c_o(h)$ is constant along minimizing trajectories. Indeed let (o,h) be such an optimal trajectory and let η be the trajectory of momenta, such that (o,η,h) satisfies Equation (5.4). Then, as $\rho_o(h) = \xi_o \circ \zeta_o(h) = \nabla_\eta \mathcal{H}$, one gets: $\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} = \partial_o \mathcal{H}\frac{\mathrm{d}o}{\mathrm{d}t} + \partial_\eta \mathcal{H}\frac{\mathrm{d}\eta}{\mathrm{d}t} + \partial_h \mathcal{H}\frac{\mathrm{d}h}{\mathrm{d}t} = (\nabla_o \mathcal{H}, \nabla_\eta \mathcal{H}) - (\nabla_\eta \mathcal{H}, \nabla_o \mathcal{H}) = 0$. So the Hamiltonian is constant along optimal trajectories and as it is equal to half of the cost, $\int_0^1 c_o(h) = c_{o_{t=0}}(h_{t=0})$ along minimizing trajectories.*

Remark 23. *Let $(o,\eta) \in T^*\mathcal{O}$. If $(h_k)_k$ is an orthonormal basis of H for g_o, then let $(\alpha_k)_k$ be the coefficients of $C_o^{-1}\rho_o^*(\eta)$ in this basis: $C_o^{-1}\rho_o^*(\eta) \doteq \sum_k \alpha_k h_k$. Then for each k:*

$$\begin{array}{rcl} \alpha_k & = & g_o(C_o^{-1}\rho_o^*(\eta), h_k) \\ & = & (\rho_o^*(\eta)|h_k)_{H^*} \\ & = & (\eta|\rho_o(h_k))_{T_o^*\mathcal{O}}. \end{array}$$

Then we see that the $k-$th coefficient α_k of $C_o^{-1}\rho_o^*(\eta)$ is equal to the application of η to the speed of o generated thanks to $k-$th element of the orthonormal basis.

5.2.2 The matching problem in practice

We will now consider the practical case when one wants to study the differences between two shapes f^1 and f^2 belonging to a common shape space $\mathcal{F}$, thanks to a deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ independent from the shape space $\mathcal{F}$ (and satisfying UEC). We first build the silent deformation module $M^2 = (\mathcal{F}, H^2, \zeta^2, \xi^2, c^2)$ induced by shape space $\mathcal{F}$ (see Section 4.2.8) and its combination $M = (\mathcal{O}, H, \zeta, \xi, c)$ with M^1 (see Section 4.1.1.4). Then M is a deformation module satisfying UEC and for $a = (o, f) \in \mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$, $h = (h^1, 0) \in H = H^1 \times \{0\}$, the application to a of the vector field generated by a and the control h is

$$\xi_o \circ \zeta_o(h) = \xi_o \circ \zeta^1_{o^1}(h^1) = (\xi^1_{o^1} \circ \zeta_{o^1}(h^1), \xi^2_f \circ \zeta_{o^1}(h^1)).$$

The sub-Riemannian distance is now defined for the augmented data set containing the input data and the geometrical descriptors of the user-defined modules. In this case, this distance cannot be used to compare directly the input shape with another input shape data, since first, one may not assume that the second shape derives from the input shape by a modular deformation, and second, one does not know what would be the corresponding component in $\mathcal{O}^1$ of the geometrical descriptor for the second shape. The strategy will then be to determine optimal trajectories for a given value of initial geometrical descriptor, and then to estimate the component in $\mathcal{O}^1$ for this geometrical descriptor which is most appropriate (the component in $\mathcal{F}$ being the fixed source shape).

Similarly to the previous section the inexact matching problem between two shapes a, b of $\mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$ amounts to minimizing

$$J_{a,b}(h) = \frac{1}{\sigma^2}\mu(o_{t=1}, b) + \int c_{o_t}(h_t)dt \tag{5.7}$$

with $o_{t=0} = a$ and $\dot{o} = \xi_o \circ \zeta_o(h)$. Besides as in practice we only have information about the second component of geometrical descriptors (belonging to $\mathcal{F}$), the attachment term function μ will only depend on this second component so that $J_{a,b}$ does not depend on the first component of b and will be written:

$$J_{o^1_0, f^1, f^2}(h) = \frac{1}{\sigma^2}\mu(f_{t=1}, f^2) + \int c^1_{o^1}(h^1)$$

with $o = (o^1, f)$ starting at $o_{t=0} = (o^1_0, f^1)$, $h = (h^1, 0)$ and $\dot{o} = (\dot{o}^1, \dot{f}) = \Big(\xi^1_{o^1} \circ \zeta^1_{o^1}(h^1), \xi^2_f \circ \zeta^1_{o^1}(h^1)\Big)$.

As previously we define the Hamiltonian $\mathcal{H} : (o, \eta, h) \in T\mathcal{O} \times H \mapsto (\eta | \xi_o \circ \zeta_o(h)) - \frac{1}{2}c_o(h)$ and we can show that optimal trajectories (o, h) are such that there exists a path $\eta : t \in [0, 1] \mapsto \eta_t \in T^*_{o_t}\mathcal{O}$ so that

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \xi_o \circ \zeta_o(h) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o} \\ \frac{\partial \mathcal{H}}{\partial h} & = & 0 \end{array} \right.$$

Remark 24. *It is important to note that even though* $h = (h^1, 0)$, $\rho_o^*(\eta) \neq \rho_{o^1}^{1,*}(\eta^1)$ *(with* $\eta = (\eta_1, \eta^2) \in T^*_{o^1}\mathcal{O}^1 \times T^*_f\mathcal{F}$ *and* $\rho^1_{o^1} = \xi^1_{o^1} \circ \zeta^1_{o^1}$*). Then, even though the component of geometrical descriptor belonging to* $\mathcal{F}$ *generates only a null vector field, its initial momentum has an influence on the trajectory.*

Optimal trajectories are parametrized by initial values of geometrical descriptor and momentum, so now in dimension $2 \times (\dim(\mathcal{O}^1) + \dim(\mathcal{F}))$. In order to solve the inexact matching problem, the strategy is to minimizing $J_{o^1_0, f^1, f^2}$ by searching only amongst minimizing trajectories. Therefore we define

$$J(o^1_{t=0}, \eta_{t=0}, f^1, f^2) = \frac{1}{\sigma^2}\mu(f_{t=1}, f^2) + \int c^1_{o^1}(h^1) \tag{5.8}$$

where $o = (o^1_{t=0}, f)$ starts at $(o^1_{t=0}, f^1)$, $h = (h^1, 0)$ satisfies $h = C_o^{-1}\rho_o^*(\eta)$ (with C_o defined in Remark 21) and (o, η) satisfies the following differential equation

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o}{\mathrm{d}t} & = & \frac{\partial \mathcal{H}_r}{\partial \eta} = \rho_o(C_o^{-1}\rho_o^*(\eta)) \\ \frac{\mathrm{d}\eta}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}_r}{\partial o} \end{array} \right.$$

As the cost $c_o(h)$ is constant along optimal trajectories, it amounts to minimizing

$$J(o^1_{t=0}, \eta_{t=0}, f^1, f^2) = \frac{1}{\sigma^2}\mu(f_{t=1}, f^2) + c^1_{o^1}(h^1) \tag{5.9}$$

The study of an optimal large deformation generated by deformation module M^1 in order to match f^1 onto f^2 is now reduced to the estimation of best initial variable $o^1_{t=0} \in \mathcal{O}^1$ and $\eta \in T^*_{o^1_{t=0}}\mathcal{O}^1 \times T^*_{f^1}\mathcal{F}$. In practice this estimation is performed thanks to a gradient descent process.

5.2.3 Interpretation of the initial momentum ?

In the previous section we saw that geodesic trajectories are parametrized by an initial value of geometrical descriptor and momentum. In this section we study the link between the momentum and the corresponding vector field.

5.2.3.1 $\zeta_o(H)$ is a RKHS

We consider here a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfying the Uniform Embedding Condition. Let V a Hilbert space continuously embedded in $\mathcal{C}_0^{\ell+k}(\mathbb{R}^d)$ and $C > 0$ so that for all $(o, h) \in \mathcal{O} \times H$, $\zeta_o(h) \in V$ and $|v|_V^2 \leq C c_o(h)$. For each $o \in \mathcal{O}$, $\zeta_o(H)$ is of finite dimension (since H is of finite dimension) then it is a RKHS. We suppose now that V is a RKHS with $K : V^* \mapsto V$ its kernel, let us express the kernel of $\zeta_o(H)$ in function of K. Let us set $o \in \mathcal{O}$ and study $\zeta_o(H)$ equipped with the inner product $\langle v, v' \rangle_{\zeta_o(H)} \doteq g_o(h, h)$ with $h, h' \in (\ker \zeta_o)^\perp$ such that $\zeta_o(h) = v$, $\zeta_o(h') = v'$ (see Section 5.1.1). We define $\pi_o : V \longrightarrow \zeta_o(H)$ the orthogonal projection on $\zeta_o(H)$ in V, and its dual $\pi_o^* : \zeta_o(H)^* \mapsto V^*$ such that for $\gamma \in \zeta_o(H)^*$ and $v \in V$, $(\gamma | \pi_o(v))_{\zeta_o(H)^*} = (\pi_o^*(\gamma) | v)_{V^*}$. Then if γ is in $\zeta_o(H)^*$, $\pi_o^*\gamma$ is in V^* and we can show that $K\pi_o^*\gamma$ is in $\zeta_o(H)$: let v be in the orthogonal of $\zeta_o(H)$ in V, then $\langle K\pi_o^*\gamma, v \rangle_V = (\pi_o^*\gamma | v)_{V^*} = (\gamma | \pi_o(v))_{\zeta_o(H)^*} = 0$

because $\pi_o(v) = 0$. We deduce that $K\pi_o^*\gamma$ is in $\zeta_o(H)$.
We now define $\Lambda : \zeta_o(H) \longrightarrow \zeta_o(H)$ such that for v and v' in $\zeta_o(H)$, $\langle \Lambda(v), v' \rangle_o = \langle v, v' \rangle_V$: let $(w_i)_i$ be an orthonormal basis of $\zeta_o(H)$ for $\langle \cdot, \cdot \rangle_V$ and let $(w_i')_i$ be an orthonormal basis of $\zeta_o(H)$ for $\langle \cdot, \cdot \rangle_{\zeta_o(H)}$. For $v = \sum_i \alpha_i w_i$ we define $\Lambda(v) = \sum_i \langle v, w_i' \rangle_V w_i'$. Then let $v' = \sum_i \beta_i w_i' \in \zeta_o(H)$ and let $v = \sum_i \alpha_i w_i$, one gets

$$\begin{aligned} \langle \Lambda(v), v' \rangle_o &= \langle \textstyle\sum_i \langle v, w_i' \rangle_V w_i', \sum_k \beta_k w_k' \rangle_o \\ &= \langle \textstyle\sum_i \langle v, w_i' \rangle_V w_i', \sum_k \beta_k w_k' \rangle_o \\ &= \textstyle\sum_i \langle v, w_i' \rangle_V \beta_i \end{aligned}$$

We conclude

$$\langle \Lambda(v), v' \rangle_o = \langle v, v' \rangle_V .$$

Then for $\gamma \in \zeta_o(H)^*$ and $v \in \zeta_o(H)$, $\Lambda K \pi_o^*(\gamma)$ is in $\zeta_o(H)$ and

$$\begin{aligned} \langle \Lambda K \pi_o^*(\gamma), v \rangle_o &= \langle K \pi_o^*(\gamma), v \rangle_V \\ &= (\pi_o^*(\gamma) | v)_{V^*} \\ &= (\gamma | \pi_o(v))_{\zeta_o(H)^*} \\ &= (\gamma | v)_{\zeta_o(H)^*} \end{aligned}$$

This shows that the kernel of $\zeta_o(H)$, equipped with $\langle \cdot, \cdot \rangle_{\zeta_o(H)}$, is $\Lambda K \pi_o^*$. This expression links the kernel of $\zeta_o(H)$ with the kernel of V. However this dependence in V is *artificial* since one could choose another space of vector fields V allowing M to satisfy UEC. This is why we will now use a more constructive approach in order to give a new expression of this kernel.
Let $(e_i)_i$ be an orthonormal basis of $\ker(\zeta_o(H))^\perp \subset H$ (for the metric g_o given by the cost c). For each i we set $f_i = \zeta_o(e_i) \in \zeta_o(H)$. Then (f_i) is an orthonormal basis of $\zeta_o(H)$ for $\langle \cdot, \cdot \rangle_o$. We can associate to this basis of $\zeta_o(H)$ the dual basis $(f_i^*)_i$ of $\zeta_o(H)^*$ defined by: for i, j, $(f_i^* | f_j)_{\zeta_o(H)^*} = \delta_{i,j}$. We can now define $K_o : \sum_i \alpha_i f_i^* \in \zeta_o(H)^* \mapsto \sum_i \alpha_i f_i \in \zeta_o(H)$, let us show that it is the kernel of $\zeta_o(H)$. Let $\gamma = \sum_i \alpha_i f_i^*$ in $\zeta_o(H)^*$ and $v = \sum_i \beta_i f_i \in \zeta_o(H)$, then

$$\begin{aligned} \langle K_o \gamma, v \rangle_{\zeta_o(H)} &= \langle \textstyle\sum_i \alpha_i K_o f_i^*, \sum_j \beta_j f_j \rangle_{\zeta_o(H)} \\ &= \langle \textstyle\sum_i \alpha_i f_i, \sum_j \beta_j f_j \rangle_{\zeta_o(H)} \\ &= \textstyle\sum_i \alpha_i \beta_i \\ &= (\textstyle\sum_i \alpha_i f_i^* | \sum_j \beta_j f_j)_{\zeta_o(H)^*} \\ &= (\gamma | v)_{\zeta_o(H)^*} \end{aligned}$$

Remark 25. *Thanks to this expression of the kernel K_o, we can easily recover the expression we had given in Chapter 2, Section 2.3.2 for the particular case of the deformation module generating a multi-scale sum of translations. Indeed, let $\sigma_1, \cdots, \sigma_N$ be N elements of $\mathbb{R}_+^*$ and for each i let $D_i \in \mathbb{N}^*$. We set for each i, $M^i = (\mathcal{O}^i, H^i, \zeta^i, \xi^i, c^i)$ the deformation module generating a sum of D_i translations at scale σ_i in dimension d, and then we define $M = (\mathcal{O}, H, \zeta, \xi, c)$ the combination of deformation modules M^i. In particular, for each i, $\mathcal{O}^i = \mathbb{R}^{dD_i}$ and $H^i = \mathbb{R}^{dD_i}$. Let us set $o = (z_j^i)_{1 \leq i \leq N, 1 \leq j \leq D_i}$ in $\mathcal{O} = \prod_i \mathcal{O}^i$. For each i we define the matrix $K_i(z^i)$ of size $D_i \times D_i$ such that is*

entry (j, j') *equals* $K_{\sigma_i}(z_j^i, z_{j'}^i)$, *and* $C_i \doteq K_i(z^i)^{-1/2}$ ($K_i(z^i)$ *is a positive definite matrix). If we follow the same reasoning as previously, the orthonormal basis of* $\zeta_o(H)$ *which we obtain is* $(\sum_k K_{\sigma_i}(z_k^i, \cdot)C_i(k,j)\alpha_u)_{1\leq i\leq N, 1\leq j\leq D_i, 1\leq u\leq d}$ *where* $(\alpha_u)_u$ *is the canonical basis of* $\mathbb{R}^d$. *Let us denote* $(\gamma_{i,j,u})_{1\leq i\leq N, 1\leq j\leq D_i, 1\leq u\leq d}$ *the dual basis of* $\zeta_o(H)^*$. *Then let* $x \in \mathbb{R}^d$ *and* $\beta \in \mathbb{R}$, *we set* $\delta_x^\beta : v \in \zeta_o(H) \mapsto \langle v(x), \beta\rangle_{\mathbb{R}^d}$. *It is easy to see that* $\delta_x^\beta = \sum_{i,j,u}\sum_l K_{\sigma_i}(z_l^i, x)C_i(l,j)\beta(u)\gamma_{i,j,u}$. *Then, by definition,*

$$\begin{aligned} K_o\delta_x^\beta &= \textstyle\sum_{i,j,u}\sum_l K_{\sigma_i}(z_l^i, x)C_i(l,j)\beta(u)\sum_k K_{\sigma_i}(z_k^i, \cdot)C_i(k,j)\alpha_u \\ &= \textstyle\sum_{i,k} K_{\sigma_i}(z_k^i, \cdot)\sum_l K_{\sigma_i}(z_l^i, x)\sum_j C_i(l,j)C_i(j,k)\beta \\ &= \textstyle\sum_{i,k} K_{\sigma_i}(z_k^i, \cdot)\sum_l K_{\sigma_i}(z_l^i, x)(C_i^2)_{l,k}\beta \end{aligned}$$

because C_i *is symmetric by construction. Then, as* $C_i^2 = K_{\sigma_i}(z^i)^{-1}$, *we obtain*

$$K_o\delta_x^\beta = \sum_{i,k} K_{\sigma_i}(z_k^i, \cdot)(K_{\sigma_i}(z_l^i, x)K_{\sigma_i}(z^i)^{-1})_k\beta$$

with $K_{\sigma_i}(x, z^i) = (K_{\sigma_i}(x, z_1^i), \cdots, K_{\sigma_i}(x, z_{D_i}^i))$. *We recognize here the expression given Section 2.3.2.*

5.2.3.2 Link between initial vector field and initial momentum

In this section we also consider a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ satisfying the Uniform Embedding Condition and we define V the Hilbert space continuously embedded in $\mathcal{C}_0^{\ell+k}(\mathbb{R}^d)$ and $C > 0$ so that for all $(o, h) \in \mathcal{O} \times H$, $\zeta_o(h) \in V$ and $|v|_V^2 \leq c_o(h)$.

We set now $(o, \eta) \in T^*\mathcal{O}$, the corresponding geodesic control is $h \doteq C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)$ and the generated vector field is $v^{o,\eta} \doteq \zeta_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta))$. We define $\xi_o^* \in V^*$ so that for all $\delta o \in T_o\mathcal{O}$, for all $v \in V$, $(\eta|\xi_o(v))_{T_o^*\mathcal{O}} = (\xi_o^*(\eta)|v)_{V^*}$. We also set F the orthogonal of $\zeta_o(H)$ in V and, as previously, we consider $\pi_o : V \mapsto \zeta_o(H)$ the orthogonal projection on $\zeta_o(H)$. Then we can decompose V^* in $V^* = \pi_o^*(\zeta_o(H)^*) + (Id - \pi_o)^*(F^*)$. Note that for $\gamma \in V^*$, there exists a unique $\gamma_1 \in \zeta_o(H)^*$ and a unique $\gamma_2 \in F^*$ such that $\gamma = \pi_o^*(\gamma_1) + (Id - \pi_o)^*(\gamma_2)$. We denote $P_o : \gamma \in V^* \mapsto P_o(\gamma) \in \pi_o^*(\zeta_o(H)^*)$ the projection on $\pi_o^*(\zeta_o(H)^*)$.

Proposition 25. *If we set* $\xi_o^*(\eta) = \pi_o^*(\gamma_1) + (Id - \pi_o)^*(\gamma_2)$ *with* $\gamma_1 \in \zeta_o(H)^*$, $\gamma_2 \in F^*$ *then*

$$\zeta_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)) = K_o\gamma_1.$$

This property can be represented by the following commutative diagram (we recall that

$K_o = \Lambda K \pi_o^*$):

$$\begin{array}{ccccc}
\xi_o^*(\eta) \in V^* & \xrightarrow{P_o} & \pi_o^*(\gamma_1) \in V^* & \xrightarrow{K} & K\pi_o^*(\gamma_1) \in \zeta_o(H) \subset V \\
\downarrow{\scriptstyle \zeta_o^*} & & & & \\
(\xi_o \circ \zeta_o)^*(\eta) \in H^* & & & & \downarrow{\scriptstyle \Lambda} \\
\downarrow{\scriptstyle C_o^{-1}} & & & & \\
C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta) \in H & \xrightarrow{\zeta_o} & \zeta_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)) & = & \Lambda K \pi_o^*(\gamma_1) \in \zeta_o(H) \subset V
\end{array}$$

Proof. Let $v \in \zeta_o(H)$ and let $h_1 \in \ker(\zeta_o)^\perp$ so that $\zeta_o(h_1) = v$ (with $Ker(\zeta_o)^\perp = \{\tilde{h} \in H \mid \forall \bar{h} \in \ker(\zeta_o), g_o(\bar{h}, \tilde{h}) = 0\}$). Then, as $C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)$ belongs to $Ker(\zeta_o)^\perp$,

$$\begin{array}{rcl}
\langle \zeta_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)), v \rangle_{\zeta_o(H)} & = & g_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta), h_1) \\
& = & ((\xi_o \circ \zeta_o)^*(\eta) | h_1)_{H^*} \\
& = & (\xi_o^*(\eta) | \zeta_o(h_1))_{V^*} \\
& = & (\xi_o^*(\eta) | v)_{V^*} \\
& = & (\gamma_1 | v)_{\zeta_o(H)^*} \\
& = & \langle K_o \gamma_1, v \rangle_{\zeta_o(H)}
\end{array}$$

□

This proposition shows that the vector field generated by $(o, \eta) \in T_o^*\mathcal{O}$ is obtained through a projection on $\zeta_o(H)^*$ of the dual vector field $\xi_o^*(\eta)$ which encodes η as a function on speeds of o.

The following proposition gives another intuition of the link between η and the speeds it confers on o.

Proposition 26. *For all $o, \eta \in T_o^*\mathcal{O}$, we have*

$$\eta \perp \{\delta o \in \xi_o \circ \zeta_o(H) \mid g_o^{\mathcal{O}}(\delta o, \xi_o(v^{o,\eta})) = 0\}.$$

Proof. Let $\delta o \in \xi_o \circ \zeta_o(H)$ such that $g_o^{\mathcal{O}}(\delta o, \xi_o(v^{o,\eta}) = 0$. Then $h = C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)$ belongs to $\ker(\xi_o \circ \zeta_o)^\perp$. Indeed, let $\tilde{h} \in \ker(\xi_o \circ \zeta_o)$, then

$$g_o(h, \tilde{h}) = g_o(C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta), \tilde{h}) = ((\xi_o \circ \zeta_o)^*(\eta) | \tilde{h})_{H^*} = (\eta | \xi_o \circ \zeta_o(\tilde{h}))_{T_o^*\mathcal{O}} = 0.$$

We can write $\delta o = \xi_o \circ \zeta_o(h_1)$ with $h_1 \in \ker(\xi_o \circ \zeta_o)^\perp$. Then :

$$0 = g_o^{\mathcal{O}}(\delta o, \xi_o(v^{o,\eta})) = g_o(h_1, C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)) = (\eta | \xi_o \circ \zeta_o(h_1))_{T_o^*\mathcal{O}}.$$

□

This proposition shows that η is orthogonal (for the co-metric) to all horizontal speeds (*i.e.* in $\xi_o \circ \zeta_o(H)$) which are orthogonal (for the sub-Riemannian metric) to the geodesic speed generated by η.

5.2.3.3 Changing the metric

We briefly study here the influence of the choice of the cost on geodesic controls. Let us consider a deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ obtained thanks to the combination of N deformation modules $M^i = (\mathcal{O}^i, H^i, \zeta^i, \xi^i, c^i)$ satisfying UEC. Let us consider another deformation module $\tilde{M} = (\mathcal{O}, H, \zeta, \xi, \tilde{c})$, similar to M except in the definition of the cost, and let us study how different are geodesic controls for M and $\tilde{M}$.
As explained in Remark 21, we can define operators $C : \mathcal{O} \times H \mapsto H^*$ and $\tilde{C} : \mathcal{O} \times H \mapsto H^*$ such that for $(o, h) \in \mathcal{O} \times H$, $(C_o(h)|h)_{H^*} = c_o(h)$ and $(\tilde{C}_o(h)|h)_{H^*} = \tilde{c}_o(h)$. The difference between costs c and $\tilde{c}$ leads to a difference between C and $\tilde{C}$. Then for a given (o, η) in $T^*\mathcal{O}$, the corresponding geodesic control for M is $C_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)$ while for $\tilde{M}$ it is $\tilde{C}_o^{-1}(\xi_o \circ \zeta_o)^*(\eta)$. We see that a same element of H^* is associated to (o, η) with M and $\tilde{M}$ but that C_o and $\tilde{C}_o$ do not associate the same control to this element of H^*.
Let us consider the particular case where $\tilde{c}$ is defined by, for $o = (o^i)_i \in \mathcal{O} = \prod \mathcal{O}^i$ and $h = (h^i)_i \in H = \prod H^i$, $\tilde{c}_o(h) = \alpha_1 c^1_{o^1}(h^1) + \sum_{i \geq 2} c^i_{o^i}(h^i)$ with $\alpha_1 > 0$ fixed. We recall that by definition, for $o = (o^i)_i \in \mathcal{O} = \prod \mathcal{O}^i$ and $h = (h^i)_i \in H = \prod H^i$, $c_o(h) = \sum_i c^i_{o^i}(h^i)$. Let us set $o = (o^i)_i \in \mathcal{O}$, we denote g_o and $\tilde{g}_o$ the metrics on H which come respectively from c_o and $\tilde{c}_o$. We will also denote $g^i_{o^i}$ the metric on H^i which comes from $c^i_{o^i}$ for each $i \in [\![1, N]\!]$. Let $h_1 = (h^i_1)_i$ be in H and let $\gamma \in H^*$, we denote $h_2 = (h^i_2)_i \doteq C_o^{-1}\gamma$ and $\tilde{h}_2 = (\tilde{h}^i_2)_i \doteq \tilde{C}_o^{-1}\gamma$. Then $(\gamma|h_1)_{H^*} = g_o(C_o^{-1}\gamma, h_1) = g_o(h_2, h_1) = \sum_i g^i_{o^i}(h^i_2, h^i_1)$. Similarly one gets $(\gamma|h_1)_{H^*} = \tilde{g}_o(\tilde{C}_o^{-1}\gamma, h_1) = \tilde{g}_o(\tilde{h}_2, h_1) = \alpha_1 g^1_{o^1}(\tilde{h}^1_2, h^1_1) + \sum_{i \geq 2} g^i_{o^i}(\tilde{h}^i_2, h^i_1)$. Then

$$\sum_i g^i_{o^i}(h^i_2, h^i_1) = \alpha_1 g^1_{o^1}(\tilde{h}^1_2, h^1_1) + \sum_{i \geq 2} g^i_{o^i}(\tilde{h}^i_2, h^i_1)\,.$$

This equality is true for all controls h_1, so we deduce that $h^1_2 = \alpha_1 \tilde{h}^1_2$ and for all $i \geq 2$, $h^i_2 = \tilde{h}^i_2$. As a consequence, the geodesic controls $h = (h^i)_i$ and $\tilde{h} = (\tilde{h}^i)_i$ associated to a value (o, η) in $T^*\mathcal{O}$ with respectively M and $\tilde{M}$, will have all their components equal except for the one of H^1 which will satisfy

$$\tilde{h}^1 = \frac{1}{\alpha_1} h^1\,.$$

This suggests that one can *favour* the use of the deformations generated by M^1 in its combination with deformation modules M^i by using a weighted sum where the weight of cost c^1 is smaller than one and all other weights are equal to one. This will be studied in Sections 7.2.5 and 7.3.5.

5.3 Modular analysis of shape variability

5.3.1 Principle

In the previous section we have shown how modular large deformations can enable to study differences between two shapes. We will now study how this geometrical framework may be used to infer statistical properties from a series of P shapes data f^k_{target} belonging to a common shape space $\mathcal{F}$. We compute an *atlas* of these shapes which

means that we build one shape, corresponding to a *mean shape*, and P deformations, each one transporting the template as close as possible to one shape. In our framework we set a deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ and we want the deformations to be modular large deformations. Then computing an atlas will correspond to computing one mean shape, one initial value of the geometrical descriptor, and modular large deformations generated by it transporting the mean shape as close as possible to target shapes. Following the idea presented in Section 5.2.2, we will build a deformation module M by combining $\tilde{M}$ and the silent deformation module induced by $\mathcal{F}$. Then computing an atlas of the data set will correspond to computing one initial value of the (compound) geometrical descriptor, and modular large deformations generated by it, bringing its silent component as close as possible to target shapes. Geometrical descriptors of the silent deformation module play the role of an "average" of the data set. The active deformation module is a dictionary, in which the shape variations seen in the data may be decomposed. The choice of the types of modules is left to the user, whereas parameters such as the initial position of their geometrical descriptors are to be optimised for a given training shape data set. In the spirit of [DPC+14], the user provides an example of template shape with the desired topology (i.e. number of points, edges between points, number of connected components, etc..), and the atlas construction method will optimize its shape (i.e. the position of the points or vertices) to be at the "center" of the training samples. The method needs to optimize at the same time the other geometrical descriptors and find the P modular large deformations ϕ^k, so that each warped template resembles as much as possible to one of the target shapes f^k_{target}. Given the initial position of geometrical descriptors, modular large deformations that need to be estimated are parametrized by initial momenta. So finally the method will estimate one initial position of geometrical descriptors and P initial momenta. These momenta may be used subsequently for other statistical tasks, such as clustering, classification or regression with covariates. We will now detail this construction.

5.3.2 Building an atlas of shapes

5.3.2.1 Framework

Let $\mathcal{F}$ be a shape space of order $\ell \geq 2$, let f^k_{target} be a series of P different shapes of $\mathcal{F}$, and let $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ be the deformation module thanks to which we want to study this series of shapes. Similarly to Section 5.2.2, we build $M = (\mathcal{O}, H, \zeta, \xi, c)$ the compound module of $\tilde{M}$ and the silent deformation module induced from $\mathcal{F}$. We suppose that $\tilde{M}$ (and then M) is C^j of order ℓ and satisfies UEC. The estimation of the atlas may be done in a coherent Bayesian framework. However, the derivation of a Bayesian approach as in [AAT07, AKT+10] is out of the scope of this thesis, and we propose here a more straightforward extension of the geometrical construction of Section 5.2.1. It is the analogue of the concept of Fréchet mean, which has been used intensively in the field of Computational Anatomy [LDJ05, JDJG04, DPC+14]. This approach amounts here to minimising the following quantity with respect to the geometrical descriptors $o_{temp} \in \mathcal{O} = \tilde{\mathcal{O}} \times \mathcal{F}$ and P trajectories $h^k \in L^2([0,1], H)$:

$$\frac{1}{\sigma^2}\sum_k \mu(f^k_{t=1}, f^k_{target}) + \int_0^1 c_{o^k}(h^k) \tag{5.10}$$

with for each k, $o^k_{t=0} = o_{temp}$ and $\dot{o}^k = \xi_{o^k} \circ \zeta_{o^k}(h^k)$.

Like in Section 5.2.1, can be shown that for a fixed value of o_{temp}, the minimizer $(h^k)_k$ of (5.10) is such that there exist P trajectories $\eta^k : t \in [0,1] \mapsto \eta^k_t \in T^*_{o^k_t}\mathcal{O}$ so that, with $\mathcal{H} : (o, \eta, h) \in T^*\mathcal{O} \times H \mapsto (\eta|\xi_o(\zeta_o(h))) - \frac{1}{2}c_o(h)$,

$$\left\{\begin{array}{rcl} h^k & = & h^{o^k,\eta^k} \doteq C^{-1}_{o^k}\rho^*_{o^k}\eta^k \\ \frac{\mathrm{d}o^k}{\mathrm{d}t} & = & \xi_{o^k} \circ \zeta_{o^k}(h^k) \\ \frac{\mathrm{d}\eta^k}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o}(o^k, \eta^k, h^k) \end{array}\right. \tag{5.11}$$

where $C : o \in \mathcal{O} \mapsto C_o$ is smooth and satisfies for each $h \in H$, $c_o(h) = (C_o h|h)_{H^*}$, and ρ^*_o is such that, for all $h \in H$, $\eta \in T^*_o\mathcal{O}$, $(\eta|\rho_o(h))_{T^*_o\mathcal{O}} = (\rho^*_o(\eta), h)_{H^*}$. Then whole trajectories (o^k, η^k, h^k) are defined by initial values of o^k and η^k. Moreover, thanks to Remark 22, the cost is constant along optimal trajectories and so for each k: $c_{o_{temp}}(h^{o_{temp},\eta^k_0}) = \int_0^1 c_{o^k}(h^{o^k,\eta^k})$. Therefore computing the atlas of shapes f^k_{target}, $k = 1 \cdots P$ thanks to the deformation module $\tilde{M}$ amounts to minimizing:

$$E(o_{temp}, (\eta^k_0)_k) = \sum_{k=1}^{P} c_{o_{temp}}(h^{o_{temp},\eta^k_0}) + \frac{1}{\sigma^2}\mu\left(\phi^{o^k,\eta^k} \cdot f_{temp}, f^k_{target}\right) \tag{5.12}$$

where for each k, (o^k, η^k) starts at (o_{temp}, η^k_0) and satisfies Equation (5.11), and $\phi^{o^k,h^k} = \varphi^{v^k}_{t=1}$ with $v^k = \zeta_{o^k}(h^k)$. As previously, the first term of this sum will be referred to as the Regularity term while the second one will be called the Data term.

Note that the initial value of geometrical descriptor o_{temp} is common to all subjects but that the trajectory o^k (starting at o_{temp}) obtained by integrating Equation (5.11) is specific to each subject as it depends on the initial momentum η^k_0, which is specific to each subject.

Remark 26. *A matching problem corresponds to an atlas problem where the template shape is fixed (equal to the source shape) and the series of data shape only has one element (equal to the target shape).*

We emphasize here that the estimated $o_{temp} = (\tilde{o}_{temp}, f_{temp})$ has two components. The second one f_{temp} is the template and corresponds to an average of the data set, while the first one $\tilde{o}_{temp}$ is a common geometric characterization of the variability among the population of shapes.

5.3.2.2 Parameters

The computation of an atlas depends on several parameters. In this thesis we suppose that they are known but we give here some ideas to estimate them in future works.

In practice the deformation module that is used is obtained through combination of base-deformation modules $M_i = (\mathcal{O}^i, H^i, \zeta^i, \xi^i, c^i)$. The compound cost is defined as the

sum of costs c^i but in a more general setting one we could consider the compound cost associated to $o = (o^i) \in \prod_i \mathcal{O}^i$ and $h = (h^i)_i \in \prod_i H^i$ defined as $c_o(h) = \sum_i \alpha_i c^i_{o^i}(h^i)$ where, for each i, $\alpha_i > 0$. By choosing different values for coefficients α_i, one can favour certain deformations over others, and especially the way these deformations are generated by our deformation modules. These coefficients could then be learnt from a population by developing a Bayesian-framework along the lines of [AAT07, GCW+13]. Similarly, initial momenta could be considered as following a centred gaussian distribution of covariance matrix Σ so that $\eta^T \Sigma \eta = c_o(h^{o,\eta})$ where $h^{o,\eta} = C_o^{-1} \rho_o^*(\eta)$. The matrix Σ depends on the geometrical descriptor o and on the coefficients α_i which may therefore be estimated through an E-M algorithm.

A Bayesian framework could also be developed in order to optimize the balance coefficient $1/\sigma^2$ between data and regularity terms.

5.4 Discussion about the model

In the framework presented in Sections 5.2 and 5.3, the dimension of geodesics is at least equal to the dimension of the shape which is transported. As this dimension is very high in practice, one may want to develop a smaller-dimension framework. We will introduce such a possible framework. Let us set a deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ and a data shape space $\mathcal{F}$.

5.4.1 A smaller-dimension framework

5.4.1.1 Presentation of the SIGS framework

We explore here a smaller-dimension version of our framework based on the idea developed in [DPC+14]. The idea is to consider, amongst large deformations that can be generated by deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$, those that are a geodesic for this deformation module (and not for the combination of M^1 and the silent module induced by $\mathcal{F}$). We know that these geodesics are parametrized by initial variables $(o_0^1, \eta_0^1) \in T\mathcal{O}^1$ then in dimension $2 \times \dim(\mathcal{O}^1)$: to each initial value $(o_0^1, \eta_0^1) \in T^*\mathcal{O}^1$ we associate an optimal trajectory (o^1, h^1) following the approach presented in 5.2.1. This trajectory then enables to build a trajectory of vector fields v which gives a trajectory of diffeomorphisms φ^v thanks to which we can transport shapes of $\mathcal{F}$. Therefore in order to study the matching of two shapes f^1 and f^2 thanks to M^1, in this framework we will minimize the following functional with respect to $(o_0^1, \eta_0^1) \in T^*\mathcal{O}^1$:

$$\tilde{J}(o_0^1, \eta_0^1, f^1, f^2) = \frac{1}{\sigma^2} \mu\left(\varphi_{t=1}^{\zeta^1_{o^1}(h^1)} \cdot f^1, f^2\right) + c^1_{o^1}(h^1) \tag{5.13}$$

where (o^1, η^1) starts at (o_0^1, η_0^1) , and satisfies ,with $\rho = \xi \circ \zeta$, $H_r^1 : (o^1, \eta^1) \in T\mathcal{O}^1 \mapsto \frac{1}{2} c_{o^1}(C_{o^1}^{-1} \rho_{o^1}^{1,*}(\eta^1))$ the reduced Hamiltonian,

$$\begin{cases} \frac{\mathrm{d}o^1}{\mathrm{d}t} &= \frac{\partial \mathcal{H}_r^1}{\partial \eta^1} = \rho_{o^1}(C_{o^1}^{-1} \rho_{o^1}^{1,*}(\eta^1)) \\ \frac{\mathrm{d}\eta^1}{\mathrm{d}t} &= -\frac{\partial \mathcal{H}_r^1}{\partial o^1} \end{cases}$$

and $\varphi^{\zeta^1_{o^1}(h^1)}$ is the flow of the trajectory of vector fields $\zeta^1_{o^1}(h^1)$.

In this framework we restrict ourselves to a set of geodesics which is *independent* from the data shapes, therefore we will call it the **shape-independent geodesic set (SIGS) framework**. On the contrary, the framework which we have developed in Sections 5.2 and 5.3 will be denoted the **shape-dependent geodesic set (SDGS) framework**.

Remark 27. *In the case where we take for M^1 a deformation modules generating sums of local translations at a fixed scale σ, this framework corresponds exactly to the framework developed in [DPC+14].*

Proposition 27. *Trajectories that can be generated by the SIGS framework are exactly trajectories of the SDGS framework parametrized by an initial momentum $\eta = (\eta^1, \eta^2)$ such that the second component η^2 is null.*

Proof. Indeed let $(o^1_0, f^1) \in \mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$, $\eta^1_0 \in T^*_{o^1_0}\mathcal{O}^1$ and let us consider the trajectory (o^1, f, η^1) that can be built from (o^1_0, f^1, η^1_0) in the SIGS framework. The trajectory (o^1, η^1) starts at (o^1_0, η^1_0) and satisfies, with $\mathcal{H}^1_r(o^1, \eta^1) = \frac{1}{2} c^1_{o^1}(C^{1,-1}_{o^1} \rho^{1,\star}_{o^1}(\eta^1))$,

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}o^1}{\mathrm{d}t} & = & \frac{\partial \mathcal{H}^1_r}{\partial \eta^1} = \rho_{o^1}(C^{1,-1}_{o^1} \rho^{1,\star}_{o^1}(\eta^1)) \\ \frac{\mathrm{d}\eta^1}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}^1_r}{\partial o^1} \end{array} \right.$$

where $C^{1,-1}_{o^1} = (C^1_{o^1})^{-1}$. Then the trajectory f starts at f^1 and satisfies $\dot{f} = \xi^2_f \circ \zeta^1_{o^1}(C^{1,-1}_{o^1} \rho^{1,*}_{o^1}(\eta^1))$.

We define the trajectory $\eta^2 : t \mapsto 0 \in T^*_{f_t}\mathcal{F}$. Let us now show that $(o^1, f^1, \eta^1, \eta^2)$ satisfies the geodesic equation for the SDGS framework. We set $M^2 = (\mathcal{F}, \{0\}, \zeta^2, \xi^2, c^2)$ the silent deformation module (see Section 4.2.8) induced by $\mathcal{F}$ and we build $M = (\mathcal{O}, H, \zeta, \xi, c)$ the combination of M^1 and M^2 (see Section 4.1.1.4). For $o = (o^1, f) \in \mathcal{O} = \mathcal{O}^1 \times \mathcal{F}$ and $h = (h^1, 0) \in H = H^1 \times \{0\}$,

$$\rho_o(h) \doteq \xi_o \circ \zeta_o(h) = (\xi^1_{o^1} \circ \zeta^1_{o^1}(h^1), \xi^2_f \circ \zeta^1_{o^1}(h^1)) = (\rho^1_{o^1}(h^1), \xi^2_f \circ \zeta^1_{o^1}(h^1)).$$

Then for $(o, \eta) = (o^1, f, \eta^1, \eta^2) \in T^*_o\mathcal{O}$,

$$(\eta | \rho_o(h))_{T^*_o\mathcal{O}} = (\eta^1 | \rho^1_{o^1}(h^1))_{T^*_{o^1}\mathcal{O}^1} + (\eta^2 | \xi^2_f \circ \zeta^1_{o^1}(h^1))_{T^*_f\mathcal{F}}$$

and we deduce:

$$\rho^*_o(\eta) = \left(\rho^{1,*}_{o^1}(\eta^1) + (\xi^2_f \circ \zeta^1_{o^1})^*(\eta^2), 0 \right)$$

Then $C^{-1}_o \rho^*_o(\eta) = \left(C^{1,-1} \rho^{1,*}_{o^1}(\eta^1) + C^{1,-1}(\xi^2_f \circ \zeta^1_{o^1})^*(\eta^2), 0 \right)$ and we deduce that the reduced Hamiltonian can be written

$$\mathcal{H}_r(o, \eta) = \frac{1}{2} c^1_{o^1} \left(C^{1,-1} \rho^{1,*}_{o^1}(\eta^1) + C^{1,-1}(\xi^2_f \circ \zeta^1_{o^1})^*(\eta^2) \right).$$

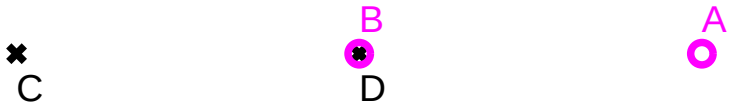


Figure 5.1: **Source shape:** magenta circles A and B. **Target Shape:** black crosses C and D. Goal: matching A with C and B with D thanks to a local translation centred at A.

This shows first that for the trajectory $(o, \eta) = (o^1, f^1, \eta^1, \eta^2)$, as $\eta^2 = 0$, geodesic controls for the SDGS framework are the same as these for the fixed dimension. We deduce that the geometrical descriptors follow the SDGS geodesic evolution:

$$\dot{o} = \rho_o C_o^{-1} \rho_o^*(\eta).$$

Besides, $\rho_o^*(\eta) = (\rho_{o^1}^{1,*}(\eta^1), 0)$ for each t and then, we can show that the momentum $\eta = (\eta^1, \eta^2)$ also follows the SDGS geodesic evolution:

$$\begin{aligned} -\tfrac{\partial \mathcal{H}_r}{\partial o^1}(o^1, f, \eta^1, \eta^2) &= -\tfrac{1}{2}\tfrac{\partial c^1}{\partial h^1}(o^1, C_o^{-1}\rho_o^*(\eta)) \cdot \tfrac{\partial C_o^{-1}\rho_o^*(\eta)}{\partial o^1}(o^1, f, \eta^1, \eta^2) \\ &= -\tfrac{1}{2}\tfrac{\partial c^1}{\partial h^1}(o^1, C^{1,-1}\rho_{o^1}^{1,*}(\eta^1)) \cdot \tfrac{\partial C^{1,-1}\rho_{o^1}^{1,*}(\eta^1)}{\partial o^1}(o^1, f, \eta^1, \eta^2) \\ &= -\tfrac{\partial \mathcal{H}_r^1}{\partial o^1}(o^1, \eta^1) \\ &= \dot{\eta}^1 \end{aligned}$$

and

$$\begin{aligned} -\tfrac{\partial \mathcal{H}_r}{\partial f}(o^1, f, \eta^1, \eta^2) &= -\tfrac{1}{2}\tfrac{\partial c^1}{\partial h^1}(o^1, \rho_o C_o^{-1}\rho_o^*(\eta)) \cdot \tfrac{\partial C_o^{-1}\rho_o^*(\eta)}{\partial f}(o^1, f, \eta^1, \eta^2) \\ &= 0 \\ &= \dot{\eta}^2 \end{aligned}$$

□

5.4.1.2 Limits of the SIGS framework: a simple example

A legitimate question is whether this sub-space of trajectory is big enough to capture the modular differences between the two shapes f^1 and f^2. We present here a simple example where the SIGS approximation is not satisfying. Let us consider the two shapes $f^1 = (A, B)$ and $f^2 = (C, D)$ presented in Figure 5.1: A needs to be matched onto C and B onto D. The data shape space $\mathcal{F}$ is then the space of two landmarks in $\mathbb{R}^2$. Let us assume that we want to match these two shapes thanks to a large deformation generated by a local translation initially centred at A. Then we consider the deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ that generates vector fields that are always a local translation at a fixed scale $\sigma = 1$: $\mathcal{O}^1 = \mathbb{R}^2$, $H^1 = \mathbb{R}^2$, $\zeta_{o^1}^1(h^1) = K_\sigma(o^1, \cdot)h^1$, $\xi_{o^1}^1(v) = v(o^1)$

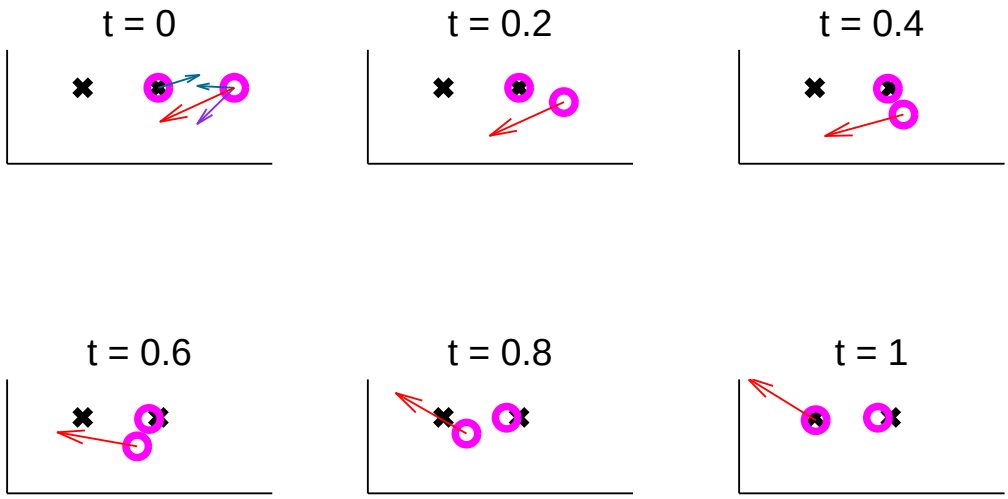


Figure 5.2: **SDGS framework**. Trajectory of source shape: magenta circles. Red arrows are the control at each time, marine blue arrows are initial momenta corresponding to the silent deformation module (attached to the source shape) and the purple arrow is the initial momentum corresponding to the deformation module generating a local translation (attached to the centre of the translation).

and $c^1_{o^1}(h^1) = |h^1|^2$. In this example we set the initial position of the centre of the translation at A. If we use the SDGS framework presented in Section 5.2.2, we consider $M = (\mathcal{O}, H, \zeta, \xi, c)$ the compound module of M^1 and the silent module induced by $\mathcal{F}$, and we want to find a value of initial momentum $\eta = (\eta^1, \eta^2) \in T^*_A\mathcal{O}^1 \times T^*_{A,B}\mathcal{F} = (\mathbb{R}^2)^3$ so that it minimizes

$$J(A, \eta, f^1, f^2) = \frac{1}{\lambda}\Big(|\varphi^v_{t=1}(A) - C|^2 + |\varphi^v_{t=1}(B) - D|^2\Big) + |h^1_{t=0}|^2$$

where $v = \zeta_o(h) = K_\sigma(o^1, \cdot)h^1$, $o \doteq (o^1, f) \doteq (o^1, f^A, f^B) \in \mathcal{O}^1 \times \mathcal{F} = (\mathbb{R}^2)^3$, $h = (h^1, 0)$ is defined by

$$h = C_o^{-1}\rho^*_o(\eta) = \eta^1 + K_\sigma(o, f^A)\eta^A + K_\sigma(o, f^B)\eta^B$$

with $\eta^2 = (\eta^A, \eta^B)$ and o starts at $o_{t=0} = (A, A, B)$ and satisfies for each t:

$$\begin{aligned}
\dot{o}_t &= \xi_{o_t} \circ \zeta_{o_t}(h_t) \\
&= \Big(\zeta_{o_t}(h_t)(o^1_t), \zeta_{o_t}(h_t)(f^A_t), \zeta_{o_t}(h_t)(f^B_t)\Big) \\
&= \Big(h^1_t, K_\sigma(o^1_t, f^A_t)h^1_t, K_\sigma(o^1_t, f^B_t)h^1_t\Big).
\end{aligned}$$

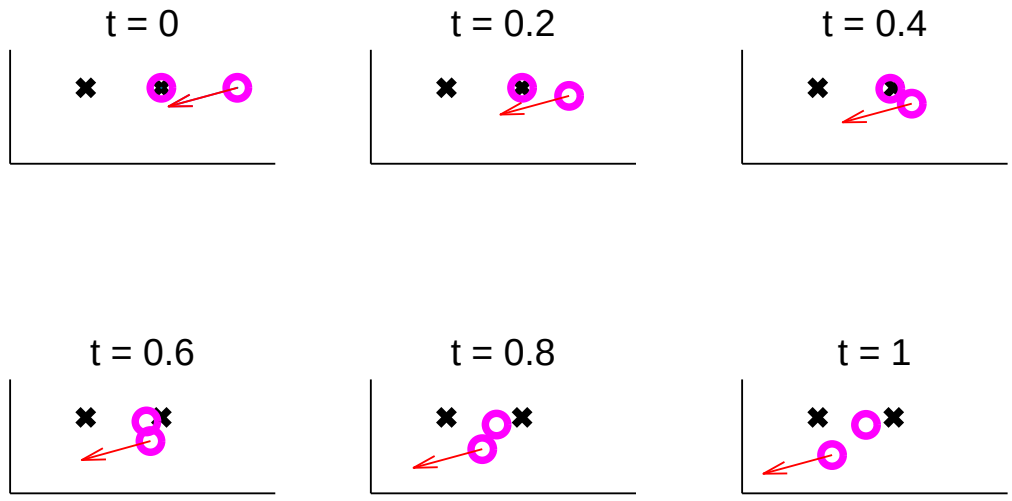


Figure 5.3: **SIGS framework**. Trajectory of source shape: magenta circles. Red arrows are the control at each time, it is equal to the momentum.

The optimal trajectory is parametrized by the initial value of the momentum and is then parametrized in dimension 6. We estimate this initial momentum thanks to a gradient descent, the resulting trajectory is presented in Figure 5.2.

Let us now consider the same matching problem with the SIGS framework: we restrict ourselves to optimal trajectories for the module M^1 whose geometrical descriptor starts at A. Such trajectories are parametrized by an initial momentum $\eta_0^1 \in T^*_{o^1}\mathcal{O}^1 = \mathbb{R}^2$. For a given initial momentum $\eta_0^1 \in T^*_{o^1}\mathcal{O}^1$, the corresponding trajectory of control satisfies: $h^1 = C^{-1}_{o^1}\rho^{1,*}_{o^1}(\eta^1) = \eta^1$. Besides the reduced Hamiltonian is $\mathcal{H}_r(o,\eta) = \frac{1}{2}c^1_{o^1}(C^{-1}_{o^1}\rho^{1,*}_{o^1}(\eta^1))$ so here $H_r(o,\eta) = |\eta^1|^2$. Therefore $\dot{h}^1 = \dot{\eta}^1 = -\frac{\partial \mathcal{H}_r}{\partial o^1}(o^1,\eta^1) = 0$ so for optimal trajectories the control is constant. Then studying the matching problem in the SIGS framework amounts here to minimize the following quantity with respect to η_0^1:

$$J(A,\eta_0^1,f^1,f^2) = \frac{1}{\lambda}\Big(|\varphi^v_{t=1}(A) - C|^2 + |\varphi^v_{t=1}(B) - D|^2\Big) + |\eta^1_{t=0}|^2$$

where $v = \zeta^1_{o^1}(\eta^1_0) = K_\sigma(o^1,\cdot)\eta^1_0$ and $\dot{o}^1 = \eta^1_0$. The optimal trajectory is here parametrized in dimension 2 and the initial momentum η_0^1 is estimated by a gradient descent, the resulting trajectory is presented in Figure 5.3. Here we can see that the needed deformation to transform f^1 into f^2 cannot be well approximated by this SIGS framework as it requests a non-constant trajectory of control. However we will show in Chapter 7 that in general for complex deformation modules, this approximation allows to build satisfying trajectories.

5.4.2 An optimal dimension ?

5.4.2.1 An intermediate framework

In the previous example was shown that in some cases the SIGS framework is not satisfying. However, the SDGS framework gives optimal trajectories parametrized in a dimension that can be uselessly high. We suggest here an intermediate framework where we build a silent deformation module $M^3 = (\mathcal{O}^3, H^3, \zeta^3, \xi^3, c^3)$ *independent* from the data shape space $\mathcal{F}$, we combine it with the user-defined deformation module M^1 and we apply the SIGS framework with the new deformation module $\tilde{M} \doteq C(M^1, M^3) \doteq (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$. In practice the space of silent geometrical descriptor $\mathcal{O}^3$ will be a space of landmarks (but it is of course not necessary). In this framework matching two shapes f^1 and f^2 thanks to the deformation module M^1 amounts to fixing a number of silent landmarks N_s, building the corresponding silent deformation module $M^3 = (\mathcal{O}^3, H^3, \zeta^3, \xi^3, c^3)$ (with $\mathcal{O}^3 = \mathbb{R}^{dN_s}$ where d is the dimension of the ambient space), then considering the compound deformation module $\tilde{M} \doteq C(M^1, M^3)$ and minimizing the following functional :

$$J(o_0^1, o_0^3, \eta_0^1, \eta_0^3, f^1, f^2) = \frac{1}{\lambda}\mu\Big(\varphi_{t=1}^v \cdot f^1, f^2\Big) + c_{o_0^1}^1(h_{t=0}^1)$$

where $v = \tilde{\zeta}_{\tilde{o}}(\tilde{C}_{\tilde{o}}^{-1}\tilde{\rho}_{\tilde{o}}^*(\tilde{\eta}))$ with $\tilde{o} = (o^1, o^3)$ and $\tilde{\eta} = (\eta^1, \eta^3)$ start at $\tilde{o}_{t=0} = (o_0^1, o_0^3) \in \tilde{\mathcal{O}}$, $\tilde{\eta}_{t=0} = (\eta_0^1, \eta_0^3) \in T^*_{\tilde{o}_{t=0}}\tilde{\mathcal{O}}$ and satisfy with $\mathcal{H}_r(\tilde{o}, \tilde{\eta}) = \frac{1}{2}\tilde{c}_{\tilde{o}_{t=0}}(\tilde{C}_{\tilde{o}}^{-1}\tilde{\rho}_{\tilde{o}}^*(\tilde{\eta}))$

$$\begin{cases} \frac{\mathrm{d}\tilde{o}}{\mathrm{d}t} &= \frac{\partial \mathcal{H}_r}{\partial \tilde{\eta}} = \tilde{\rho}_{\tilde{o}}(\tilde{C}_{\tilde{o}}^{-1}\tilde{\rho}_{\tilde{o}}^*(\tilde{\eta})) \\ \frac{\mathrm{d}\tilde{\eta}}{\mathrm{d}t} &= -\frac{\partial \mathcal{H}_r}{\partial \tilde{o}}. \end{cases}$$

The initial position of geometrical descriptors of this new silent deformation is optimized, and can then be interpreted as *significant* areas for the matching problem.

By choosing a particular number of silent landmarks N_s (or in general a particular space of geometrical descriptors for the silent deformation module), we can choose the dimension of the trajectories that we restrict ourselves to. However in practice there is no easy way to guess the optimal dimension a priori, but it could be estimated through a Bayesian framework following the idea of [AK15] where a notion of *active* and *inactive* control point in the sparse LDDMM model is presented. This Bayesian approach is not studied here.

5.4.2.2 Back to the example

We come back to the example of the previous section 5.4.1: the data shape space $\mathcal{F}$ is $\mathbb{R}^2 \times \mathbb{R}^2$ and we want to match $f^1 = (A, B)$ onto $f^2 = (C, D)$ (see Figure 5.1) thanks to a modular large deformation generated by the deformation module M^1 that creates vector fields equal to a local translation at a fixed scale σ. Besides we impose that the initial position for the center of the translation is A.
We consider the shape space of one landmark $\mathcal{O}^3 = \mathbb{R}^2$, we build the silent deformation module M^3 induced by this shape space and then $\tilde{M}$ the compound deformation module of M^1 and M^3. We can show that if we force the initial geometrical descriptor $o =$

$(o^1, o^3) \in \mathcal{O}^1 \times \mathcal{O}^3$ to be equal to f^1, then the SIGS framework with $\tilde{M}$ is equivalent to the SDGS framework with $M = C(M^1, M^2)$ (with M^2 the silent deformation module induced by $\mathcal{F}$). This will show that, even though in the SDGS framework optimal trajectories are parametrized in dimension 6, in fact they lie in a subspace of dimension at most 4. In order to highlight this, let us first consider equations of optimal trajectories in the SDGS framework for deformation module $M = C(M^1, M^2)$. The geodesic control associated to geometrical descriptor $o = (o^1, f^A, f^B) \in \mathcal{O}^1 \times \mathcal{F}$ and a momentum $\eta = (\eta^1, \eta^A, \eta^B) \in T^*_{o^1}\mathcal{O}^1 \times T^*_f\mathcal{F}$ (with $f = (f^A, f^B)$) is $h = \eta^1 + K_\sigma(o^1, f^A)\eta^A + K_\sigma(o^1, f^B)\eta^B$. The reduced Hamiltonian equations (5.2.2), becomes here: $\dot{o}^1 = h$, $\dot{f}^A = K(o^1, f^A)h$, $\dot{f}^B = K(o^1, f^B)h$ and

$$\begin{aligned} \dot{\eta}^1 &= -(h^T\eta^A)\partial_1 K_\sigma(o^1, f^A) - (h^T\eta^B)\partial_1 K_\sigma(o^1, f^B) \\ \dot{\eta}^A &= -(h^T\eta^A)\partial_2 K_\sigma(o^1, f^A) \\ \dot{\eta}^B &= -(h^T\eta^B)\partial_2 K_\sigma(o^1, f^B). \end{aligned}$$

Besides, if we set $o^1_{t=0} = f^A_{t=0} = A$, then for each t, $o^1_t = f^A_t$. Therefore the geodesic control is $h = \eta^1 + \eta^A + K_\sigma(o^1, f^B)\eta^B$ and previous equations become $\dot{o}^1 = h$, $\dot{f}^A = h$, $\dot{f}^B = K(o^1, f^B)h$ and

$$\begin{aligned} \dot{\eta}^1 &= -(h^T\eta^B)\partial_1 K_\sigma(o^1, f^B) \\ \dot{\eta}^A &= 0 \\ \dot{\eta}^B &= -(h^T\eta^B)\partial_2 K_\sigma(o^1, f^B). \end{aligned}$$

Then let us introduce $\tilde{\eta} = \eta^1 + \eta^A$, the system $(o^1, \tilde{\eta}, f^B, \eta^B)$ satisfies, with $h = \tilde{\eta} + K_\sigma(o^1, f^A)\eta^A$, $\dot{o}^1 = h$, $\dot{f}^B = K(o^1, f^B)h$ and

$$\begin{aligned} \dot{\tilde{\eta}} &= -(h^T\eta^B)\partial_1 K_\sigma(o^1, f^B) \\ \dot{\eta}^B &= -(h^T\eta^B)\partial_2 K_\sigma(o^1, f^B). \end{aligned}$$

This is the equation, for the deformation module $\tilde{M} = C(M^1, M^3)$, of the optimal trajectory with initial geometrical descriptor equal to (A, B) and initial momentum $(\tilde{\eta}_{t=0}, \eta^B_{t=0})$. Then with the SIGS framework for the deformation module $\tilde{M}$, we can build the same trajectories as with the SDGS framework: the dimension of SDGS geodesic with geometrical descriptor starting at (A, A, B) is at most 4. Here the intermediate framework is perfectly adapted.

5.4.2.3 Dimension of geodesics in a simple example

A consequence of the sub-Riemannian framework is that there may exists several geodesics starting at the same geometrical descriptors $o_{t=0}$ and with the same initial speed $\dot{o}_{t=0}$. Let us study these trajectories in the case of the deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ obtained through the combination of $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ generating a local translation at scale σ in dimension 2 (then $\mathcal{O}^1 = \mathbb{R}^2$ and $H^1 = \mathbb{R}^2$), and the silent module $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ with shape space $\mathcal{O}^2 = \mathbb{R}^2$ (space of one landmark). Then $\mathcal{O} = \mathcal{O}^1 \times \mathcal{O}^2 = \mathbb{R}^2 \times \mathbb{R}^2$. For a value $(o^1, o^2, \eta^1, \eta^2) \in T^*\mathcal{O}$, the geodesic control is $h^{o,\eta} = \eta^1 + K_\sigma(o^1, o^2)\eta^2$, then the reduced Hamiltonian is $H_r(o, \eta) = \frac{1}{2}|\eta^1 + K_\sigma(o^1, o^2)\eta^2|^2$. We deduce the following geodesic equations:

$$\left\{\begin{array}{lcl} \frac{\mathrm{d}o^1}{\mathrm{d}t} &=& \eta^1 + K_\sigma(o^1,o^2)\eta^2 \\ \frac{\mathrm{d}o^2}{\mathrm{d}t} &=& K_\sigma(o^1,o^2)(\eta^1 + K_\sigma(o^1,o^2)\eta^2) \\ \frac{\mathrm{d}\eta^1}{\mathrm{d}t} &=& -(\eta^1 + K_\sigma(o^1,o^2)\eta^2)^T\eta^2(-2/\sigma^2)K_\sigma(o^1,o^2)(o^1-o^2) \\ \frac{\mathrm{d}\eta^2}{\mathrm{d}t} &=& -\frac{\mathrm{d}\eta^1}{\mathrm{d}t}. \end{array}\right. \tag{5.14}$$

As the geodesic control is $h = \eta^1 + K_\sigma(o^1,o^2)\eta^2$, it can be easily seen that its speed is $\dot{h} = (-2/\sigma^2)(1-K_\sigma(o^1,o^2))K(o^1,o^2)h \wedge ((o^1-o^2)\wedge\eta^2)$ where $\wedge$ is the cross product in $\mathbb{R}^3$ and $\mathbb{R}^2$ is seen as a subspace of $\mathbb{R}^3$. The system of equations (5.14) can be re-written in function of the geometrical variable $u \doteq o^1 - o^2$, $h = \eta^1 + K_\sigma(o^1,o^2)\eta^2$ and η^2 :

$$\left\{\begin{array}{lcl} \frac{\mathrm{d}u}{\mathrm{d}t} &=& (1-K_\sigma(u))h \\ \frac{\mathrm{d}h}{\mathrm{d}t} &=& (-2/\sigma^2)(1-K_\sigma(u))K(u)h\wedge(u\wedge\eta^2) \\ \frac{\mathrm{d}\eta^2}{\mathrm{d}t} &=& h^T\eta^2(-2/\sigma^2)K_\sigma(u)u \end{array}\right.$$

where $K(u) \doteq \exp(-\frac{|u|^2}{\sigma^2})$. Let us set initial values u_0, h_0 and study the geodesic trajectories that we can build thanks to an initial η_0^2. Let us suppose that η_0^2 and $\tilde{\eta}_0^2$ generate trajectories of control h and $\tilde{h}$ that are equal. Then $\dot{h}_{t=0} = \dot{\tilde{h}}_{t=0}$ and we can deduce $u_0\wedge\eta_0^2 = u_0\wedge\tilde{\eta}_0^2$ (since they are orthogonal to the 2-D plane). The expression of the second derivative of the geodesic control is

$$\begin{array}{rcl} \mathrm{d}^2h/\mathrm{d}t^2 &=& (-2/\sigma^2)K(u)(1-K(u))\Big((-2/\sigma^2)(1-2K(u))(u^Th)h\wedge(u\wedge\eta^2) \\ && +(-2/\sigma^2)K(u)(1-K(u))(h\wedge(u\wedge\eta^2))\wedge(u\wedge\eta^2) \\ && +(1-K(u))h\wedge(h\wedge\eta^2)\Big) \end{array}$$

From this equation, as $u_0\wedge\eta_0^2 = u_0\wedge\tilde{\eta}_0^2$, we obtain that $\mathrm{d}^2h/\mathrm{d}t^2 = \mathrm{d}^2\tilde{h}/\mathrm{d}t^2$ at time $t=0$ then $h_0\wedge(h_0\wedge\eta_0^2) = \tilde{h}_0\wedge(\tilde{h}_0\wedge\tilde{\eta}_0^2)$ and then $h_0\wedge\eta_0^2 = \tilde{h}_0\wedge\tilde{\eta}_0^2$. If h_0 and u_0 are not collinear we can deduce that $\eta_0^2 = \tilde{\eta}_0^2$. Therefore the only way to generate same trajectories of controls with different momentum η^2 is to have at each time $u\wedge h = 0$. Besides $\frac{\mathrm{d}}{\mathrm{d}t}(u\wedge h) = u\wedge\dot{h}$ since $\dot{u}$ is collinear to h, and as $h^T\dot{h} = 0$ ($|h|^2$ is constant) we deduce, if h collinear to u, that $\dot{h}$ needs to be null. Then from the expression of $\dot{h}$ one gets that $u\wedge\eta^2$ needs to be null, *ie* η^2 needs to be collinear to u. Then $\eta_0^2 \neq \tilde{\eta}_0^2$ generate different trajectories of controls except if they are collinear to u and h too. We can easily show that if (u,h,η^2) satisfies this condition (all collinear), then the trajectory of geodesic control only depend on the initial value of control h, not on η^2, and is a constant trajectory. This simple example shows several things. First, different initial momenta can give the same geodesic trajectory: here if η^1 and η^2 are collinear to u, the geodesic trajectory only depends on the value of $\eta^1 + K(u)\eta^2$. Second, the same initial geodesic control h (and then same initial speed $\dot{o}_{t=0}$) can correspond to different geodesic trajectory. For instance if η_0^1 is not collinear to u, we showed that the initial momentum $(\eta_0^1 + K(u)u, -u)$ will generate a geodesic trajectory which is different from the one generated by the initial momentum $(\eta_0^1, 0)$. However these to initial momenta will generate the same initial geodesic control (equal to η_0^1). Last, the example shows that an *optimal dimension* might not be globally relevant.

Chapter 6

Algorithm

Contents

6.1 Introduction

We presented in Section 5.2 how our model enables to compare two shapes by computing a *modular large deformation* warping the first as close as possible to the second one. Then in Section 5.3 we presented how we could study a population of shapes by computing an atlas. In this chapter we study the computational aspects of our framework. We will concentrate on the atlas computation since matching one shape onto another is equivalent to studying the atlas problem while fixing the template equal to the first shape and having only one shape in the population equal to the second shape.

We suppose that we want to compute an atlas of P shapes $f_k^{targets}$ belonging to a shape space $\mathcal{F}$ (with infinitesimal action ξ^F) thanks to a deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$. We recall that we need to consider the silent deformation module $M^0 = (\mathcal{O}^0, H^0, \zeta^0, \xi^0, c^0)$ induced by $\mathcal{F}$: $\mathcal{O}^0 = \mathcal{F}$, $H^0 = \{0\}$, $\zeta^0 = 0$, $\xi^0 = \xi^F$ and $c^0 = 0$. Then we define M the combination of the silent deformation module $M^0 = (\mathcal{F}, H^0, \zeta^0, \xi^0, c^0)$ induced by $\mathcal{F}$ and the deformation module $\tilde{M}$. In particular $\mathcal{O} = \mathcal{F} \times \tilde{\mathcal{O}}$. As explained in previous section, the theoretical computation of the atlas amounts to minimizing the following quantity over $(o_{temp}, (\eta_{k,0})_k) \in \prod_{\mathcal{O}} T^*\mathcal{O}$:

$$E(o_{temp}, (\eta_{k,0})_k) = \sum_{k=1}^{P} c_{o_{temp}}(h^{o_{temp},\eta_{k,0}}) + \frac{1}{\sigma^2}\mu\Big(f_k(t=1), f_k^{target}\Big) \tag{6.1}$$

where for each k, we denote $o_k = (f_k, \tilde{o}_k)$, $(o_k, \eta_k) : [0,1] \mapsto T^*\mathcal{O}$ starts at $(o_k(t=0), \eta_k(t=0)) = (o_{temp}, \eta_{k,0}) \in T^*\mathcal{O}$ and satisfies:

$$\left\{ \begin{array}{rcl} h_k & = & h^{o_k,\eta_k} \doteq C_{o_k}^{-1} \rho_{o_k}^\star \eta_k \\ \frac{\mathrm{d}o_k}{\mathrm{d}t} & = & \xi_{o_k} \circ \zeta_{o_k}(h_k) \\ \frac{\mathrm{d}\eta_k}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}}{\partial o}(o_k, \eta_k, h_k) \end{array} \right. \tag{6.2}$$

with $\mathcal{H}(o, \eta, h) = (\eta | \xi_o \circ \zeta_o(h)) - \frac{1}{2} c_o(h)$ the Hamiltonian. In the following we will show how these equations can be easily computed in practice.

This chapter is organized as follow: in a first section 6.2 we present the global structure of the algorithm we built. In a second section 6.3 we introduce some mathematical results which are needed in the implementation. In the last section 6.4 we present explicit calculus in a simple example.

6.2 Structure of the algorithm

6.2.1 Global presentation of the gradient descent

We present here the global pseudo-code for the estimation of an atlas: we are given P shapes $f_1, \cdots, f_P$ of a shape space $\mathcal{F}$ and a deformation module $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ thanks to which we want to build this atlas. Shapes f_k are implemented thanks to column matrices of points which will be denoted $\underline{f_k}$, and potentially a matrix of connectivity. In general in this chapter we will denote $\underline{X}$ the implementation in the algorithm of the mathematical notion X. As explained in Section 4.1.1.4, a simple way to build complex deformation modules is to see them as the compound deformation module of simple ones.

Therefore in practice, we consider N base-deformation module $M^i = (\mathcal{O}^i, H^i, \zeta^i, \xi^i, c^i)$ so that the deformation module $\tilde{M}$ their combination. For each i, geometrical descriptors o^i of $\mathcal{O}^i$ are implemented as column matrices which will be denoted $\underline{o^i}$. As recalled previously, in order to compute the atlas of shapes f_k thanks to $\tilde{M}$ we need to consider its combination $M \doteq (\mathcal{O}, H, \zeta, \xi, c)$ with the silent deformation module induced by $\mathcal{F}$. Then each geometrical descriptor $o = (f, o^1, \cdots, o^N) \in \mathcal{O} = \mathcal{F} \times \mathcal{O}^1 \times \cdots \times \mathcal{O}^N$ is implemented as a list $\underline{o}$ of the matrices $\underline{f}, \underline{o^1}, \cdots, \underline{o^N}$. In order to compute the atlas we need to estimate such an initial geometrical descriptor $\underline{o_{temp}}$ as well as P initial momenta (one per subject) $\eta_{k,0} = (\eta^f_{k,0}, \eta^1_{k,0}, \cdots, \eta^N_{k,0}) \in T_f\mathcal{F} \times T_{o^1}\mathcal{O}^1 \times \cdots \times T_{o^N}\mathcal{O}^N$. Similarly to geometrical descriptors, each of these initial momenta is implemented as a list $\underline{\eta_{k,0}}$ of column matrices, and then the total initial momentum $\underline{\eta_0}$ is implemented as a list of a lists of matrices. The estimation of the initial geometrical descriptor $\underline{o_{temp}}$ and the initial momentum $\underline{\eta_0}$ so that they minimize Equation (6.1) is performed thanks to a gradient descent. A global view of the implementation of this gradient descent is presented in Algorithm 1. We present separately the computation of the energy in the Algorithm 2 where the integration of Equation 6.2 is performed thanks to an Euler integration scheme. We will present how the cost and the derivatives of the Hamiltonian are implemented in next Section 6.3.2.1. We also present separately an overview of the computation of the gradient in the Algorithm 3, the reason why we perform the given differential equation is presented in Section 6.3.2.2. Its numerical resolution is also performed via an Euler integration scheme.
We implemented our algorithm in the software Deformetrica [DPC+14], so that the data attachment term $\mu(f_k(t=1), f_k^{target})$, and its derivatives, can be automatically computed for a large collection of shapes: point clouds, curve and surface meshes in 2D and 3D (corresponding to currents or varifolds and then without point of correspondence). Therefore we do not detail the computation of μ and its derivatives.

6.2.2 Object-oriented implementation

We present now how this method is implemented in practice. We use an object-oriented language as the proposed modular deformation framework is very well adapted for this type of implementation. The structure is presented in Figure 6.1. All kind of modules (such as these presented in Section 4.2) inherit from a single abstract module class, named ABSTRACTMODULE, which contains abstract methods defining the deformation module. Attributes of this class are matrices of GEOMETRICALDESCRIPTOR, MOMENTUM and CONTROL. This class is also defined thanks to three methods which use values of these attributes. The first method is COMPUTEVECTORFIELDS, it implements the field generator ζ. Its input is a matrix of points and its output is a new matrix (of the same size) with the application of the vector field generated by the deformation module thanks to the values of GEOMETRICALDESCRIPTOR and CONTROL to this list of points. The second method is APPLYMODULE, it takes in input a second deformation module and returns the application of the its application to the geometrical descriptors, this implements the infinitesimal action ξ. The last method is COMPUTEB, it computes the matrix $B(o)$ such that for each $(o, h) \in O \times H$,

Algorithm 1 Gradient Descent

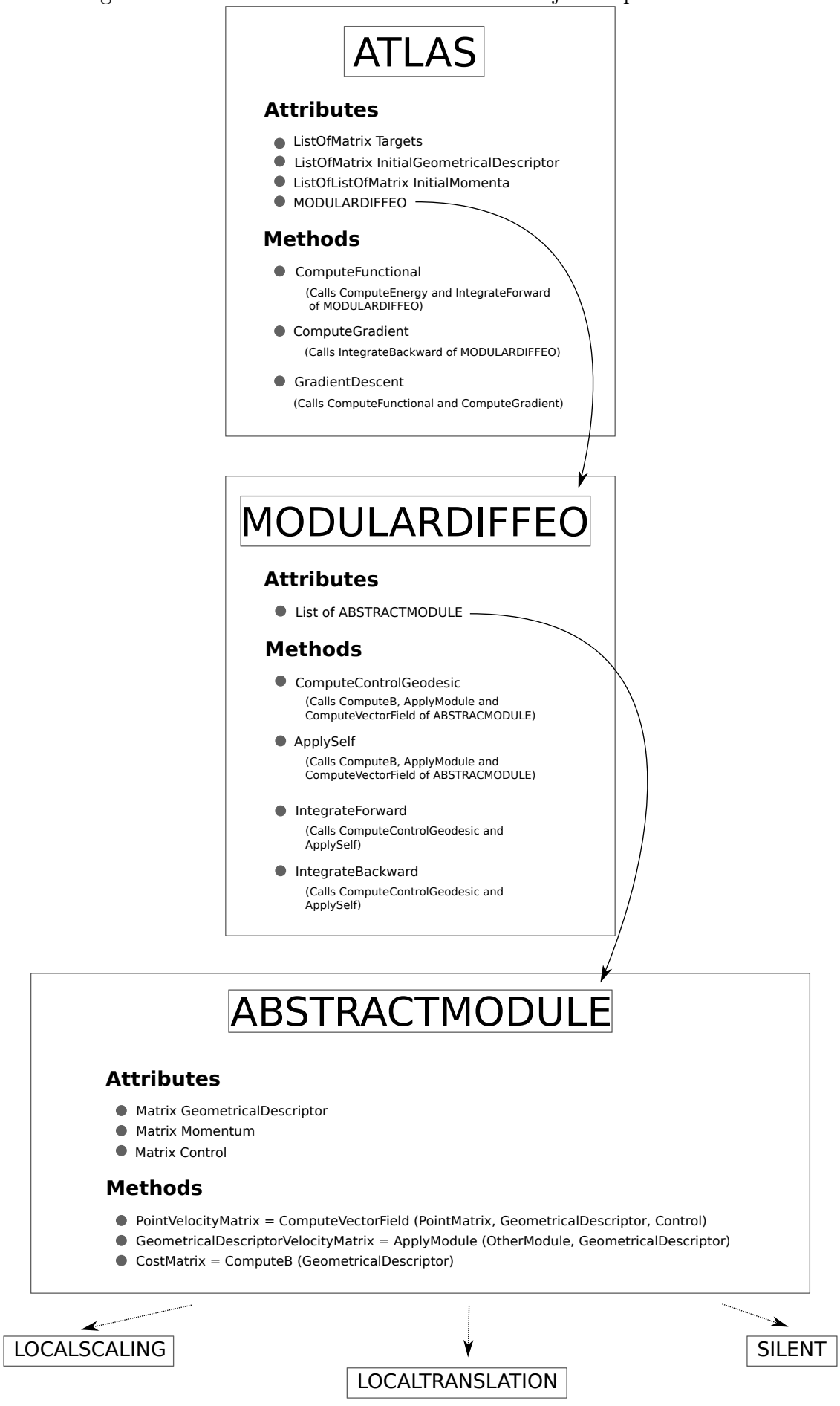
Input: :

- P targets f_k^{target}, $k = 1 \cdots P$
- M a deformation module
- Initialisation of $\underline{o_{temp}}$ (list of $N+1$ matrices)
- Initialisation of $\underline{\eta_0} = [\underline{\eta_{1,0}}, \cdots, \underline{\eta_{P,0}}]$ (list of P lists of $N+1$ matrices)
- ϵ (criterion to stop the gradient descent)

$\underline{E} \doteq E(\underline{o_{temp}}, \underline{\eta_0})$ (see Algorithm 2)
Gradient Descent :
$iter \doteq 0$
while $iter < iter\_max$ & $stop == FALSE$ **do**
  $\underline{Grad} = Grad(\underline{o_{temp}}, \underline{\eta_0})$ (see Algorithm 3)
  $Found \doteq FALSE$
  $iter\_line\_search \doteq 0$
  while $Found == FALSE$ **do**
    $\underline{\tilde{o}_{temp}} \doteq \underline{o_{temp}} - Step \times \underline{Grad_{o_{temp}}}$
    for $k = 1 \cdots P$ **do**
      $\underline{\tilde{\eta}_{k,0}} \doteq \underline{\eta_{k,0}} - Step \times \underline{Grad_{\eta_{k,0}}}$
    end for
    $\underline{\tilde{E}} = E(\underline{\tilde{o}_{temp}}, \underline{\eta_0})$ (see Algorithm 2)
    $iter\_line\_search = iter\_line\_search + 1$
    $\underline{Found} \doteq \underline{\tilde{E}} < \underline{E}$ OR $iter\_linesearch = iter\_linesearch\_max$
  end while
  if $iter\_line\_search == iter\_line\_search\_max$ **then**
    $stop \doteq TRUE$
  else
    $iter = iter + 1$
    $\underline{o_{temp}} = \underline{\tilde{o}_{temp}}$
    $\underline{\eta_0} = \underline{\tilde{\eta}_0}$
    $stop \doteq$ "$(\underline{E} - \underline{\tilde{E}}) > \epsilon \underline{E}$"
    $\underline{E} = \underline{\tilde{E}}$
  end if
end while

Output: $(\underline{o_{temp}}, \underline{\eta_0})$

Figure 6.1: Schematic view of the oriented-object implementation.



Algorithm 2 Compute_Energy $E(o_{temp}, (\eta_{k,0})_k)$ (Equation (6.1))

$E = 0$
for $k = 1 \cdots P$ **do**
  Integrate Equation (6.2)
  Save trajectories $(o_k(t), h_k(t), \eta_k(t))$
  $E = E + c_{o_k(t=0)}(h_k(t=0)) + \frac{1}{\sigma^2}\mu(f_k(t=1), f_k^{target})$ where
  $o_k(t) = (f_k(t), o_k^1(t), \cdots, o_k^N(t))$
end for
Output: E

Algorithm 3 Compute_Gradient $Grad(o_{temp}, (\eta_{k,0})_k)$

$Grad_{o_{temp}} = 0$
for $k = 1 \cdots P$ **do**
  Initialise $\nu_{t=1} = (\nu_{o,t=1}, \nu_{\eta,t=1}) = \frac{1}{\sigma^2}\nabla_{o_k(t=1),\eta_k(t=1)}\mu(f_k(t=1), f_k^{target})$
  Integrate backward $\dot{\nu} = \mathrm{d}(\nabla_{(o_k,\eta_k)}\mathcal{H})_{(o_k,\eta_k)} \cdot (-\nu_\eta, \nu_o)$
  $Grad_{o_{temp}} = Grad_{o_{temp}} + \nu_{o,t=0}$
  $Grad_{\eta_{k,0}} = \nu_{\eta,t=0}$
end for
Output: $Grad = (Grad_{o_{temp}}, Grad_{\eta_{k,0}}, k = 1 \cdots P)$

$c_o(h) = \underline{h}^T B(o)\underline{h}$ where $\underline{h}$ is the column vector representing control h. These three methods can also compute the derivations of ζ, ξ and B.
Then a meta class named MODULARDIFFEO has as attributes a list of base-modules. Its methods implement the combination rules whose implementation will be detailed in Section 6.3. First the method COMPUTECONTROLGEODESIC computes geodesic controls from values of geometrical descriptor and momenta thanks to methods of ABSTRACTMODULE. A second method APPLYSELF enables to compute speeds of geometrical descriptors and momenta from their values. The geodesic trajectories of these variables are computed thanks to an integration scheme in a method INTEGRATEFORWARD. The last method of this class is INTEGRATEBACKWARD. It enables, from final values of geometrical descriptors, momenta and their adjoints, to compute the directional derivative which will be presented in Proposition 30 and then the backward integration necessary in the computation of the gradient (Algorithm 3).
The class MODULARDIFFEO is itself the attribute of the class ATLAS which performs the gradient descent presented in Algorithm 1 thanks to a method GRADIENTDESCENT. The computation of the Energy (Algorithm 2) and the gradient (Algorithm 3) are made thanks to calls to its attribute MODULARDIFFEO. A schematic view of the computation of an atlas thanks to the ATLAS class is presented in Figure 6.2.

In this architecture one can see that the definition of new modules is as simple as "plug-and-play". A collection of base deformation modules has been implemented and one only needs to set their parameters (such as the scale for local translations) to use them in the computation of the atlas. In order to implement a new deformation module,

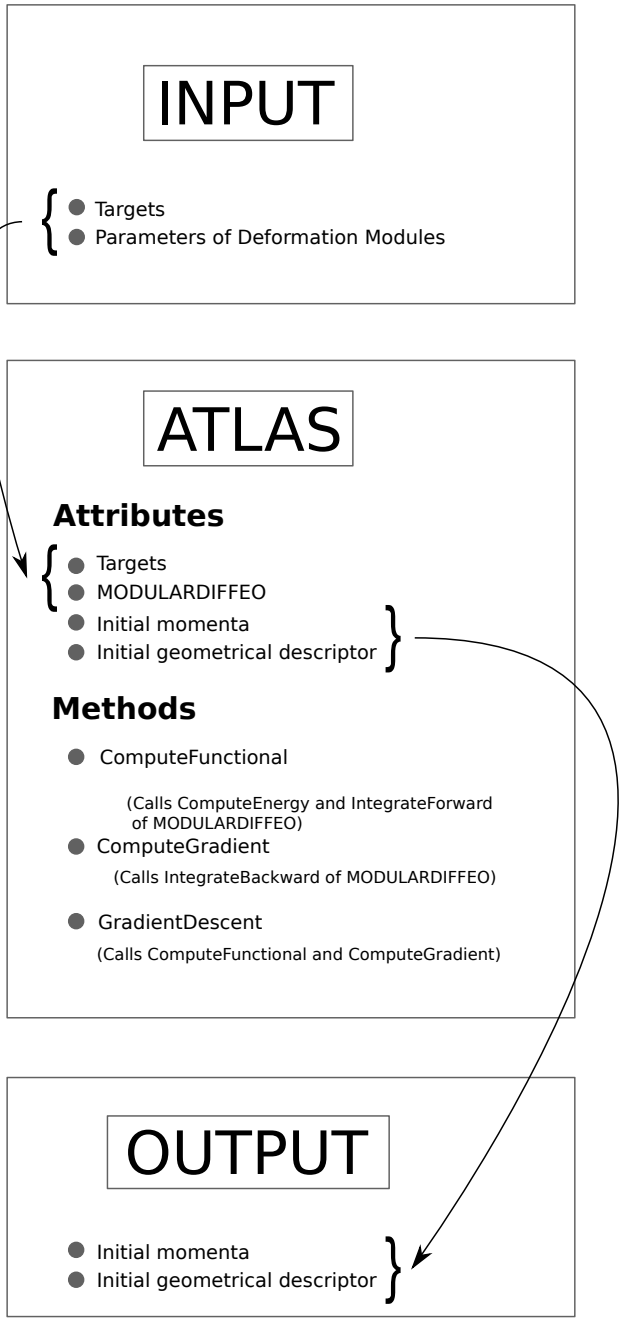


Figure 6.2: Schematic view of the computation of an atlas.

the only requirement is to define functions COMPUTEVECTORFIELDS, APPLYMODULE and COMPUTEB, then it is directly incorporated in the global algorithm and can be used.

6.3 Mathematical aspects of the algorithm

In this section we will present in detail how we implemented the calculus required to compute the gradient descent minimizing (6.1). For each k we set $E_k(o_{temp}, \eta_{k,0}) = c_{o_{temp}}(h^{o_{temp},\eta_{k_0}}) + \frac{1}{\sigma^2}\mu\Big(f_k(t=1), f_k^{target}\Big)$ so that $E(o_{temp}, (\eta_{k,0})_k) = \sum_{k=1}^{P} E_k(o_{temp}, \eta_{k,0})$. Then

$$\nabla_{o_{temp}} E(o_{temp}, (\eta_{k,0})_k) = \sum_{k=1}^{P} \nabla_{o_{temp}} E_k(o_{temp}, \eta_{k,0})$$

and for each k,

$$\nabla_{\eta_{k,0}} E(o_{temp}, (\eta_{k',0})_{k'}) = \nabla_{\eta_{k,0}} E_k(o_{temp}, \eta_{k,0}).$$

In this section we fix k and we detail how to compute ∇E_k, the computation of ∇E is then easily obtained by summations. We denote $REG_k(o_{temp}, \eta_{k,0}) = c_{o_{temp}}(h^{o_{temp},\eta_{k_0}})$ and $DATA_k(o_{temp}, \eta_{k,0}) = \frac{1}{\sigma^2}\mu\Big(f_k(t=1), f_k^{target}\Big)$.

We first give in Section 6.3.1 a theoretical result which is needed and then in Section 6.3.2 we present how energy E_k and its gradient can be computed.

6.3.1 Theoretical gradient computation

The first term of E_k is $REG_k(o_{temp}, \eta_{k,0}) = c_{o_{temp}}(h^{o_{temp},\eta_{k,0}}) = c_{o_{temp}}(C^{-1}_{o_{temp}}\rho^*_{o_{temp}}\eta_{k,0})$, it explicitly only depends on $(o_{temp,k,0})$ so there is no theoretical problem to the computation of the gradient (see Section 6.3.2 for the detail of its implementation). The second term is $DATA_k(o_{temp}, \eta_{k,0}) = \frac{1}{\sigma^2}\mu\Big(f_k(t=1), f_k^{target}\Big)$, it depends on values at time $t=1$ of $o_k = (f_k, o_k^1, \cdots, o_k^N)$, so in order to calculate its gradient we use the following result (see [Arg14]):

Proposition 28. *Let $n \in \mathbb{N}$, let U be an open subset of $\mathbb{R}^n$, $w : U \longrightarrow \mathbb{R}^n$ be a complete C^j vector field on U ($j \geq 1$), G be the function of class C^1 defined on U by $G(q_0) = g(q_{t=1})$ where g is a function of class C^1 and $q : [0,1] \longrightarrow \mathbb{R}^n$ is the solution of $\dot{q}_t = w(q_t)$ for almost every $t \in [0,1]$ and $q_{t=0} = q_0$. Then $\nabla G(q_0) = Z(1)$ where $Z : [0,1] \longrightarrow \mathbb{R}^n$ is the solution of $\dot{Z}(t) = dw^T_{q_{(1-t)}} Z_t$ for almost every $t \in [0,1]$ and $Z(0) = \nabla g(q_{t=1})$.*

In our case we have $U = T^*\mathcal{O} = T^*F \times T^*\mathcal{O}^1 \times \cdots T^*\mathcal{O}^N$, $q = (o_k, \eta_k) : t \in [0,1] \longrightarrow (o_k(t), \eta_k(t)) \in T^*\mathcal{O}$, $g(o_k(t=1), \eta_k(t=1)) = \mu\Big(f_k(t=1), f_k^{target}\Big)$ with $o_k = (f_k, o_k^1, \cdots, o_k^N)$, and $w(o_k, \eta_k) = (\partial_\eta \mathcal{H}, -\partial_o \mathcal{H})(o_k, \eta_k, h^{o_k,\eta_k})$ with $\mathcal{H}$ and h^{o_k,η_k} defined in Equation (6.2) (See Proposition 29 for the practical computation of geodesic control h^{o_k,η_k}). Then the gradient of the quantity $\mu(f_k(t=1), f_k^{target})$ with respect to initial values $(o_k(t=0), \eta_k(t=0))$

can be computed by first integrating Equation 6.2. Then an adjoint variable $\nu \in T_{(o_k(t=1),\eta_k(t=1))}T^*O$ is initialized with $\nu(t=1) = \nabla_{(o_k(t=1),\eta_k(t=1))}\mu(f_k(t=1), f_k^{target})$ such that only the adjoint variable of $f_{t=1}^k$ is non zero. Finally by integrating $\dot{\nu}_t = dw^T_{(o_k(1-t),\eta_k(1-t))}\nu_t$, where $w(o_k, \eta_k) = (\nabla_{\eta_k} H_r, -\nabla_{o_k} H_r)(o_k, \eta_k)$, one obtains $\nabla_{(o_k(t=0),\eta_k(t=0))}\mu(f_k(t=1), f_k^{target}) = \nu(t=0)$.

6.3.2 Computation in practice

We will explain in this section how previous equations are implemented. We will first detail the computation of the energy E_k, *ie* the implementation of geodesic controls and of the integration of the forward Equation (6.2). These calculus will also enable to obtain directly the derivation of the regularity term. Then we will present how the backward integration of Section 6.3.1 is computed, allowing to compute the gradient of the data term. We recall that in practice the deformation module $M = (\mathcal{O}, H, \zeta, \xi, c)$ that we consider is a combination of N deformation modules $M^i = (\mathcal{O}^i, H^i, \zeta^i, \xi^i, c^i)$ and a silent deformation module $M^0 = (\mathcal{O}^0, H^0, \zeta^0, \xi^0, c^0)$. Let us set such deformation modules. As detailed previously, they are implemented thanks to their methods COMPUTEVECTORFIELD, APPLYMODULE and COMPUTEB. In the following equations, we will denote $\underline{\zeta}$ the function of a deformation module which takes in input a matrix of geometrical descriptor, a column matrix of controls, and returns a vector field. It corresponds to the implemented function COMPUTEVECTORFIELD. We also implement its derivatives: we denote $\underline{\zeta}(l, \underline{o}, \underline{h})$ the column vector corresponding to the derivative of $\zeta_o(h)$ with respect to the $l-$th component of o, with $\underline{o}$ a column matrix of geometrical descriptor and $\underline{h}$ a column matrix of control. We will denote $\underline{\xi}$ the function of a deformation module which takes in input a matrix of geometrical descriptor and a vector field (potentially generated by another deformation module) and returns the matrix of the application of the vector field to the geometrical descriptor. We will also denote $\underline{\xi}(l, \underline{o}, \underline{v})$ the column vector corresponding to the derivative of $\xi_o(v)$ with respect to the $l-$th component of o, with $\underline{o}$ a column matrix of geometrical descriptor and $\underline{v}$ a vector field. This function corresponds to the implemented function APPLYMODULE. Note that these two functions $\underline{\zeta}$ and $\underline{\xi}$ are not exactly these that are implemented: they use the notion of vector field while in practice only notions of matrices and deformation modules are used. However they will be used in the following for simplicity. Last, we will denote $\underline{B(o)}$ the matrix depending on the geometrical descriptor o such that for all control h, $c_o(h) = \underline{h}^T \underline{B(o)} \underline{h}$, with $\underline{h}$ the column matrix of control. We will also denote $\underline{B(l, o)}$ the matrix corresponding to the derivative of $\underline{B(o)}$ with respect to the $l-$th component of o so that for all controls h, $\frac{\partial c_o(h)}{\partial o(l)} = \underline{h}^T \underline{B(l, o)} \underline{h}$. This function corresponds to COMPUTEB. For each deformation module M^i we will denote $\underline{X^i}$ its implementation of $\underline{X}$. Geometrical descriptors, momenta and controls will be implemented by column vectors.

6.3.2.1 Forward integration

We will exhibit the expression of the geodesic control $h^{o,\eta} = C_o^{-1}\rho_o^*\eta$ and $\frac{\partial \mathcal{H}}{\partial o}$ in terms of functions $\underline{\zeta}$, $\underline{\xi}$, $\underline{B}$ and their derivatives. Indeed deformation modules are defined by these functions and their derivatives, and then will be implemented thanks to them (see Section 6.2.2).

Proposition 29. *We denote $\underline{C^i_j(o^i)}$ the j-th column of the matrix $\underline{B^i(o^i)}^{-1}$. Then the j-th component of the control of i-th module for geodesics with geometrical descriptors $o = (o^i)_i \in \prod_i \mathcal{O}^i$ and momenta $\eta = (\eta^i)_i \in \prod_i T^*_{o^i}\mathcal{O}^i$ is*

$$\underline{h^i}(j) = \sum_l \left(\underline{\xi^l}(\underline{o^l}, \underline{\zeta^i}(\underline{o^i}, \underline{C^i_j(o^i)}))\right)^T \underline{\eta^l}.$$

Proof. For each $i, l \in \{1, \cdots, N\}$ and $o^i \in \mathcal{O}^i$, $o^l \in \mathcal{O}^l$ let us define a matrix $\underline{A^{i,l}(o^i, o^l)}$ of size $\dim(\mathcal{O}^i) \times dim(H^l)$ such that $\underline{\xi^i}(\underline{o^i}, \underline{\zeta^l}(\underline{o^l}, \underline{h^l})) = \underline{A^{i,l}(o^i, o^l)}\underline{h^l}$. Then we can rewrite the Hamiltonian, see Theorem 6, as follow :

$$\mathcal{H}(o, \eta, h) = \sum_i \sum_l (\underline{\eta^l})^T \underline{A^{l,i}(o^l, o^i)}\underline{h^i} - \frac{1}{2}\sum_i (\underline{h^i})^T \underline{B^i(o^i)}\underline{h^i}$$

We can then deduce that along geodesic trajectories, as $\nabla_h H = 0$, for each i

$$\underline{h^i} = \sum_l \underline{B^i(o^i)}^{-1} \underline{A^{l,i}(o^l, o^i)}^T \underline{\eta^l} = \sum_l \left(\underline{A^{l,i}(o^l, o^i)C^i(o^i)}\right)^T \underline{\eta^l}$$

Then the j-th component of $\underline{h^i}$ is

$$\underline{h^i}(j) = \sum_l \left(\underline{A^{l,i}(o^l, o^i)}\underline{C^i_j(o^i)}\right)^T \underline{\eta^l} = \sum_l \left(\underline{\xi^l}(\underline{o^l}, \underline{\zeta^i}(\underline{o^i}, \underline{C^i_j(o^i)}))\right)^T \underline{\eta^l}.$$

□

Remark 28. *Note that in this equality we see that a momentum η^l influences h^i through the inner product of a vector field generated by the $i-th$ deformation module applied to geometrical descriptor o^l. We had already pointed it out in Remark 23.*

Thanks to this Proposition, we see that if we know for each base-deformation module M^i, its field generator ζ^i, its infinitesimal action ξ^i and its cost c^i (implemented thanks to matrices B^i), we can compute geodesic controls. Once this geodesic control is computed, it is easy to compute speeds of geometrical descriptor $o = (o^i)_{1\leq i\leq N}$ and momentum $\eta = (\eta^i)_{1\leq i\leq N}$. Let us set $(o, \eta) = (o^i, \eta^i) \in \mathcal{O} = \prod_i \mathcal{O}^i$, let $h = (h^i) \in H = \prod_i H^i$ be the corresponding geodesic control and let $i \in [|1, n|], \frac{\mathrm{d}o^i}{\mathrm{d}t} = \xi^i_{o^i}(\zeta_o(h))$ and then

$$\underline{\dot{o^i}} = \sum_{l=1}^{N} \underline{\xi^i}(\underline{o^i}, \underline{\zeta^l}(\underline{o^l}, \underline{h^l}))\,.$$

The velocity of the momentum is given by derivating the Hamiltonian: $\mathcal{H}(o,\eta,h) = (\eta|\xi_o \circ \zeta_o(h)) - \frac{1}{2}c_o(h)$ then

$$\frac{\partial \mathcal{H}}{\partial o}(o,\eta,h)\cdot\delta o = \left(\eta|(\frac{\partial \xi_o}{\partial o}(o)\cdot\delta o)\circ\zeta_o(h)\right) + \left(\eta|\xi_o\circ(\frac{\partial \zeta_o}{\partial o}(o)\cdot\delta o)(h)\right) - \frac{1}{2}\frac{\partial c_o(h)}{\partial o}(o)\delta o\,.$$

Then let D be the dimension of $\mathcal{O}^i$, let $j \in [|1,D|]$ and let $X(j)$ refer to the $j-$th component of X, then

$$\begin{aligned}
-\frac{\mathrm{d}\eta^i(j)}{\mathrm{d}t} &= \frac{\partial \mathcal{H}}{\partial o^i(j)}(o,\eta,h)\\
&= \left(\eta|(\frac{\partial \xi_o}{\partial o^i(j)}(o))\circ\zeta_o(h)\right) + \left(\eta|\xi_o\circ(\frac{\partial \zeta_o}{\partial o^i(j)}(o))(h)\right) - \tfrac{1}{2}\frac{\partial c_o(h)}{\partial o^i(j)}(o)\\
&= \textstyle\sum_l \left(\eta^l|(\frac{\partial \xi^l_{o^l}}{\partial o^i(j)}(o))\circ\zeta_o(h)\right) + \sum_l \left(\eta|\xi_o\circ(\frac{\partial \zeta^l_{o^l}}{\partial o^i(j)}(o))(h^l)\right) - \tfrac{1}{2}\sum_l \frac{\partial c^l_{o^l}(h)}{\partial o^i(j)}(o)\\
&= \left(\eta^i|(\frac{\partial \xi^i_{o^i}}{\partial o^i(j)}(o^i))\circ\zeta_o(h)\right) + \left(\eta|\xi_o\circ(\frac{\partial \zeta^i_{o^i}}{\partial o^i(j)}(o^i))(h^i)\right) - \tfrac{1}{2}\frac{\partial c^i_{o^i}(h^i)}{\partial o^i(j)}(o^i)\\
&= \textstyle\sum_l \left(\eta^i|(\frac{\partial \xi^i_{o^i}}{\partial o^i(j)}(o^i))\circ\zeta^l_{o^l}(h^l)\right) + \sum_l \left(\eta^l|\xi^l_{o^l}\circ(\frac{\partial \zeta^i_{o^i}}{\partial o^i(j)}(o^i))(h^i)\right) - \tfrac{1}{2}\frac{\partial c^i_{o^i}(h^i)}{\partial o^i(j)}(o^i)
\end{aligned}$$

Finally we showed that speeds of geometrical descriptor and momentum can easily be obtained by defining functions ζ, ξ, c and their derivatives for each base deformation module, and then implementing the following combination rules: for lists of matrices $\underline{o} = [\underline{o^0},\cdots,\underline{o^N}]$ and $\underline{\eta} = [\underline{\eta^0},\cdots,\underline{\eta^N}]$ the geodesic speeds (encoded as lists of matrices) are

$$\left\{\begin{array}{rcl}
\underline{\dot{o^i}} &=& \sum_{l=1}^N \underline{\xi^i}(\underline{o^i},\underline{\zeta^l}(\underline{o^l},\underline{h^l}))\\
\underline{\dot{\eta^i}(j)} &=& -\sum_l(\underline{\eta^i})^T\underline{\xi^i}(j,\underline{o^i},\zeta^l(\underline{o^l},\underline{h^l})) - \sum_l(\underline{\eta^l})^T\underline{\xi^l}(\underline{o^l},\underline{\zeta^i}(j,\underline{o^i},\underline{h^i}))\\
&& +\frac{1}{2}(\underline{h^i})^T\underline{B^i(j,o^i)}\underline{h^i}
\end{array}\right. \tag{6.3}$$

These expressions enable also to compute the derivation of the regularity term $c_o(h)$. Indeed, thanks to Equation (5.5), $c_o(h^{o,\eta}) = 2\mathcal{H}(o,\eta,h^{o,\eta})$ with $h^{o,\eta}$ the geodesic control associated to o and η. Then

$$\begin{aligned}
\frac{\partial c_o(h^{o,\eta})}{\partial(o,\eta)} &= 2\frac{\partial \mathcal{H}(o,\eta,h^{o,\eta})}{\partial(o,\eta)}\\
&= 2\frac{\partial \mathcal{H}}{\partial(o,\eta)}(o,\eta,h^{o,\eta}) + 2\frac{\partial \mathcal{H}}{\partial h}(o,\eta,h^{o,\eta})\cdot\frac{\partial h^{o,\eta}}{\partial(o,\eta)}(o,\eta,h^{o,\eta})\\
&= 2\frac{\partial \mathcal{H}}{\partial(o,\eta)}(o,\eta,h^{o,\eta})
\end{aligned}$$

since by definition $\frac{\partial \mathcal{H}}{\partial h}(o,\eta,h^{o,\eta}) = 0$. Then as $(\dot{o},\dot{\eta}) = (\nabla_\eta\mathcal{H}(o,\eta,h^{o,\eta}), -\nabla_o\mathcal{H}(o,\eta,h^{o,\eta}))$, one gets $\frac{\partial c_o(h^{o,\eta})}{\partial(o,\eta)} = (-2\dot{\eta}, 2\dot{o})$ which is given by previous Equation 6.3. We deduce that matrices implementing the derivatives of the first term of E_k with respects to variables $\underline{\eta^i}$ and $\underline{o^i(j)}$ are:

$$\left\{\begin{array}{rcl}
\underline{\nabla_{o^i(j)}REG_k} &=& 2\sum_l(\underline{\eta^i})^T\underline{\xi^i}(j,\underline{o^i},\zeta^l(\underline{o^l},\underline{h^l})) + 2\sum_l(\underline{\eta^l})^T\underline{\xi^l}(\underline{o^l},\underline{\zeta^i}(j,\underline{o^i},\underline{h^i}))\\
&& -(\underline{h^i})^T B^i(j,o^i)\underline{h^i}\\
\underline{\nabla_{\eta^i}REG_k} &=& 2\sum_{l=1}^N \underline{\xi^i}(\underline{o^i},\underline{\zeta^l}(\underline{o^l},\underline{h^l}))
\end{array}\right. \tag{6.4}$$

6.3.2.2 Adjoint equation

Thanks to proposition 28 we can now compute the derivative of functional E^k. However, we need to compute second derivatives of the Hamiltonian $\mathcal{H}$, which in practice can only be done through a long computation time. In order to overcome this drawback, S. Arguillère showed in [Arg14] that this second derivative can be obtained thanks to a directional derivative which is easy to approximate. We recall this result in the following proposition.

Proposition 30. *Let* $f : (o,\eta) \in T^*\mathcal{O} \mapsto (\nabla_\eta \mathcal{H}, -\nabla_o \mathcal{H})(o,\eta,h^{o,\eta})$, *and let* $((o,\eta),(\nu_o,\nu_\eta)) \in T^*T^*\mathcal{O}$, *then* $\mathrm{d}f^T_{(o,\eta)} \cdot (\nu_o,\nu_\eta) = \mathrm{d}(\nabla_{(o,\eta)}\mathcal{H})_{(o,\eta,h^{o,\eta})} \cdot (-\nu_\eta,\nu_o)$.

Proof. We have :

$$\begin{aligned}
\mathrm{d}f^T_{(o,\eta)} \cdot (\nu_o,\nu_\eta) &= \nabla_{(o,\eta)}(\langle f(o,\eta),(\nu_o,\nu_\eta)\rangle)(o,\eta) \\
&= \nabla_{(o,\eta)}(\langle \nabla_\eta \mathcal{H},\nu_o\rangle - \langle \nabla_o \mathcal{H},\nu_\eta\rangle) \\
&= \begin{pmatrix} \nabla_o(\nabla_\eta \mathcal{H}^T \nu_o) & - & \nabla_o(\nabla_o \mathcal{H}^T \nu_\eta) \\ \nabla_\eta(\nabla_\eta \mathcal{H}^T \nu_o) & - & \nabla_\eta(\nabla_o \mathcal{H}^T \nu_\eta) \end{pmatrix} \\
&= \begin{pmatrix} \nabla_\eta(\nabla_o \mathcal{H})\nu_o & - & \nabla_o(\nabla_o \mathcal{H})\nu_\eta \\ \nabla_\eta(\nabla_\eta \mathcal{H})\nu_o & - & \nabla_o(\nabla_\eta \mathcal{H})\nu_\eta \end{pmatrix} \\
&= \nabla_\eta(\nabla_{(o,\eta)}\mathcal{H})\nu_o + \nabla_o(\nabla_{(o,\eta)}\mathcal{H})(-\nu_\eta)
\end{aligned}$$

So finally $\mathrm{d}f^T_{(o,\eta)} \cdot (\nu_o,\nu_\eta) = \mathrm{d}(\nabla_{(o,\eta)}\mathcal{H})_{(o,\eta,h^{o,\eta})} \cdot (-\nu_\eta,\nu_o)$. □

The computation of the gradient of the Hamiltonian was explained in Section 6.3.2.1 (see Equation 6.4) and then the temporal derivative of the adjoint variable can be obtained thanks to two estimations of this gradient at (o,η) and $(o,\eta) + \delta s(-\nu_\eta,\nu_o)$ with δs small enough.

6.4 Explicit calculus in an example

We will detail here the case where one wants to compute an atlas of two shapes f_1^{target} and f_2^{target}, each one composed of 3 landmarks, thanks to a deformation module obtained by combination of a deformation module $M^1 = (\mathcal{O}^1, H^1, \zeta^1, \xi^1, c^1)$ generating a sum of two unconstrained translations at scale σ_1 (see Section 4.2.2) and a deformation module $M^2 = (\mathcal{O}^2, H^2, \zeta^2, \xi^2, c^2)$ generating a local scaling at scale σ_2 (see Section 4.2.3). We will restrain ourselves to the 2 dimensional case. We recall the definitions of deformation modules M^1 and M^2:

- M^1:
  - $\mathcal{O}^1 = (\mathbb{R}^2)^2$, $H^1 = (\mathbb{R}^2)^2$

- $\zeta^1(o^1, h^1) = K_{\sigma_1}(z^1, \cdot)\alpha^1 + K_{\sigma_1}(z^2, \cdot)\alpha^2$ where $o^1 = (z^1, z^2)$ and $h^1 = (\alpha^1, \alpha^2$
- $\xi^2(o^1, v) \in (\mathbb{R}^2)^2(v(z^1), v(z^2))$ where $o^1 = (z^1, z^2)$
- $c^1(o^1, h^1) = |\zeta^1_{(z^1,Z^2))}((\alpha^1, \alpha^2))|^2_{V_{\sigma_1}} = |\alpha^1|^2 + |\alpha^2|^2 + 2K_{\sigma_1}(z^1, z^2)\langle\alpha^1, \alpha^2\rangle$ where $o^1 = (z^1, z^2)$ and $h^1 = (\alpha^1, \alpha^2)$

- M^2 :
  - $\mathcal{O}^2 = \mathbb{R}^2$, $H^2 = \mathbb{R}$
  - $\zeta^2(z, \alpha) = \sum_{j=1}^3 K_{\sigma_2}(z + d_j, \cdot)d_j\alpha$ where $d_1 = 0.3\sigma_2(0, -1)$, $d_2 = 0.3\sigma_2(-\frac{\sqrt{3}}{2}, \frac{1}{2})$, $d_3 = 0.3\sigma_2(\frac{\sqrt{3}}{2}, \frac{1}{2})$
  - $\xi^2(z, v) = v(z)$
  - $c^2(z, \alpha) = |\zeta^2_z(\alpha)|^2_{V_{\sigma_2}} = \sum_{1\leq j,j'\leq 3} K_{\sigma_2}(z + d_j, z + d_{j'})\alpha^2\langle d_j, d_{j'}\rangle$

Remark 29. *We emphasize that we will detail here all calculus as an example, for comprehension purpose, but that in practice one only implements functions ζ, ξ and c (and their derivatives) of each deformation module, as well as combination rules in the class MODULARDIFFEOS (which does not depend on the modules that are used and so only has to be implemented once).*

6.4.1 Procedure

We first have to set the two parameters σ_1 and σ_2. We then use 3 base module: M^1 (generating sums of two local translations at scale σ_1), M^2 (generating local scalings at scale σ_2) and $M^0 = (\mathcal{O}^0, H^0, \zeta^0, \xi^0, c^0)$ the silent deformation module induced by $(\mathbb{R}^2)^3$ (shape space to which f_1 and f_2 belong). Then $\mathcal{O}^0 = (\mathbb{R}^2)^3$. In the computation of the atlas we will estimate one shared initial geometrical descriptor o_{temp} and two initial momenta. The geometrical descriptor is made of one template $f_{temp} \in (\mathbb{R}^2)^3$ (made of three landmarks similarly to f_1^{target} and f_2^{target}), initial values for the centres of translations $o^1_{temp} \doteq (z^1_{temp}, z^2_{temp}) \in (\mathbb{R}^2)^2$, an initial value for the center of the scaling $o^2_{temp} \in \mathbb{R}^2$. Each of the two initial moment is a dual variable of o_{temp} and then belongs to $(\mathbb{R}^2)^3 \times (\mathbb{R}^2)^2 \times \mathbb{R}^2$. As we proceed by gradient descent, we need to initiate these three variables. The mean shape f_{temp} can be initialized at the Euclidean mean of f_1^{target} and f_2^{target}, the center of the scaling at the center of $_{temp}$ and the two centres of translations regularly positioned around it. Momenta are in general initialized at zero. Then we perform a gradient descent in order to minimize functional 6.1. In our case the attachment term is given by the euclidean distance between landmarks: $\sum_{k=1}^2 \mu\Big(f^k(t = 1), f^k_{target}\Big) = \sum_{k=1}^2 \sum_{j=1}^3 |x^k_j(t = 1) - x^k_{j,target}|^2$ where for each k, $x^k = (x^k_1, x^k_2, x^3_3)$ and $x^k_{target} = (x^k_{1,target}, x^k_{2,target}, x^k_{3,target})$. The regularity term, the forward and backward integrations will be detailed in next sections.

6.4.2 Energy

We will first explain how one can compute the regularity term $\sum_{k=1}^{2} c_o(h_k)$ from values of geometrical descriptor o and controls h_k, $k \in \{1,2\}$, (which in practice will be a geodesic controls computed thanks to Proposition 29, see next Section 6.4.3 for its computation). The space of controls is $H \doteq H^0 \times H^1 \times H^2 = \{0\} \times (\mathbb{R}^2)^2 \times \mathbb{R}$. Let us set for $k \in \{1,2\}$, $h_k = (0, \alpha_k^1, \alpha_k^2, \beta_k)$ with $\alpha_k^1, \alpha_k^2 \in \mathbb{R}^2$ and $\beta_k \in \mathbb{R}$. We also set $o = (o^0, o^1, o^2) \in \mathcal{O}^0 \times \mathcal{O}^1 \times \mathcal{O}^2$, and $o^1 = (z^1, z^2) \in \mathcal{O}^1 = (\mathbb{R}^2)^2$. Then the energy will be given by

$$\begin{array}{rcl} \sum_{k=1}^{2} c_o(h_k) & = & \sum_{k=1}^{2} c^0_{o^0}(0) + c^1_{(z^1,z^2)}((\alpha_k^1, \alpha_k^2)) + c^2_{o^2}(\beta_k) \\ & = & \sum_{k=1}^{2} |\alpha_k^1|^2 + |\alpha_k^2|^2 + 2K_{\sigma_1}(z^1, z^2)\langle \alpha_k^1, \alpha_k^2 \rangle \\ & & + \sum_{1 \leq j,j' \leq 3} K_{\sigma_2}(z + d_j, z + d_{j'}) \alpha_k^2 \langle d_j, d_{j'} \rangle \end{array}$$

In practice in order to compute this energy we define for each deformation module M^i a function B^i which associate to each geometrical descriptor $o^i \in \mathcal{O}^i$ a matrix $B^i(o^i)$ so that with h^i the column vector representation of a control, the cost equals $h^{iT} B^i(o^i) h^i$. Let us explicit these matrices in our example. For deformation module M^0, as the space of controls is the null space, $B^0(o^0)$ is the empty matrix for all $o^0 \in \mathcal{O}^0$. For deformation module M^1, let $o^1 = (z^1, z^2) \in \mathcal{O}^1 = (\mathbb{R}^2)^2$, the control is of dimension 4 then $B^1(o^1)$ is of size 4×4:

$$\begin{array}{rcl} B^1(o^1) & = & \begin{pmatrix} 1 & 0 & K_{\sigma_1}(z^1, z^2) & 0 \\ 0 & 1 & 0 & K_{\sigma_1}(z^1, z^2) \\ K_{\sigma_1}(z^1, z^2) & 0 & 1 & 0 \\ 0 & K_{\sigma_1}(z^1, z^2) & 0 & 1 \end{pmatrix} \\ & = & \begin{pmatrix} 1 & K_{\sigma_1}(z^1, z^2) \\ K_{\sigma_1}(z^1, z^2) & 1 \end{pmatrix} \otimes I_2 \end{array} \tag{6.5}$$

where I_2 is the identity matrix in dimension 2 and $\otimes$ is the Kronecker product of matrices.

For deformation module M^2, let $o^2 \in \mathcal{O}^2 = \mathbb{R}^2$, the control is of dimension 1 then $B^2(o^2)$ is of size 1×1:

$$\begin{array}{rcl} B^2(o^2) & = & \sum_{1 \leq j,j' \leq 3} K_{\sigma_2}(o^2 + d_j, o^2 + d_{j'}) \langle d_j, d_{j'} \rangle \\ & = & 0.3\sigma^{\frac{1}{2}} \sum_{1 \leq j,j' \leq 3} K_{\sigma_2}(o^2 + d_j, o^2 + d_{j'}) \\ & = & 0.3\sigma^{\frac{1}{2}} \Big(3 + 6K_{\sigma_2}(o^2 + d_1, o^2 + d_2) \Big) \end{array} \tag{6.6}$$

with d_j given at the beginning of this Section 6.4, last line is obtained thanks to the fact that values of $K_{\sigma_2}(o^2 + d_j, o^2 + d_{j'})$ are all equal for $j \neq j'$ and equal 1 for $j = j'$.

6.4.3 Geodesic controls

We will here detail how one can compute geodesic control from values of geometrical descriptor $o = (o^0, o^1, o^2) \in \mathcal{O}^1 \times \mathcal{O}^2 \times \mathcal{O}^3$ and momentum $\eta = (\eta^0, \eta^1, \eta^2) \in T^*_{o^0}\mathcal{O}^0 \times$

$T^*_{o^1}\mathcal{O}^1 \times T^*_{o^2}\mathcal{O}^2$ thanks to Proposition 29. We set $o^0 \doteq (a,b,c) \in \mathcal{O}^0 = (\mathbb{R}^2)^3$ and $o^1 \doteq (z^1,z^2) \in \mathcal{O}^1 = (\mathbb{R}^2)^2$. We need to compute inverses of matrices B^i (see 6.5 and 6.6): for $o^1 = (z^1,z^2) \in \mathcal{O}^1 = (\mathbb{R}^2)^2$

$$B^1(o^1)^{-1} = \frac{1}{1-K_{\sigma_1}(z^1,z^2)^2} \begin{pmatrix} 1 & -K_{\sigma_1}(z^1,z^2) \\ -K_{\sigma_1}(z^1,z^2) & 1 \end{pmatrix} \otimes I_2$$

and for $o^2 \in \mathcal{O}^2 = \mathbb{R}^2$, $B^2(o^2)^{-1} = \frac{1}{0.3\sigma_2^{\frac{1}{2}}\left(3+6K_{\sigma_2}(o^2+d_1,o^2+d_2)\right)}$.

We set $o^0 = (y_1,y_2,y_3) \in \mathcal{O}^0 = (\mathbb{R}^2)^3$, $\eta^0 = (\eta^0_1,\eta^0_2,\eta^0_3) \in T^*_{o^0}\mathcal{O}^0 = (\mathbb{R}^2)^3$, $o^1 = (z_1,z_2) \in \mathcal{O}^1 = (\mathbb{R}^2)^2$ and $\eta^1 = (\eta^1_1,\eta^1_2) \in T^*_{o^1}\mathcal{O}^1 = (\mathbb{R}^2)^2$. We want to compute the geodesic control $h \in H = H^0 \times H^1 \times H^2$ corresponding to o and η. We can write $h = (0,\alpha,\beta) \in H^0 \times H^1 \times H^2 = \{0\} \times (\mathbb{R}^2)^2 \times \mathbb{R}$. Control α is of dimension 4, first two components corresponding to the vector of the first translation (centred at z_1) and last two components corresponding to the vector of the second translation (centred at z_2). We use the expression given in Proposition 29. Let $u \in [|1,4|]$, we denote C^1_u the $u-$th column of $B^1(o^1)^{-1}$, it has 4 lines and we set γ_1 the 2D vector made of the first two components, and γ_2 the 2D vector made of the last two components. Then the $u-th$ component of α is

$$\begin{aligned}
\alpha(u) &= \sum_{i=0}^2 \left(\xi^i_{o^i}(\zeta^1_{o^1}(C^1_u(o^1))\right)^T \eta^i \\
&= \sum_{i=0}^2 \left(\xi^i_{o^i}(\sum_{r=1}^2 K_{\sigma_1}(z_r,\cdot)\gamma_r)\right)\eta^i \\
&= \sum_{j=1}^3 \left(\sum_{r=1}^2 K_{\sigma_1}(z_r,y_j)\gamma_r\right)^T \eta^0_j + \sum_{j=1}^2 \left(\sum_{r=1}^2 K_{\sigma_1}(z_r,z_j)\gamma_r\right)^T \eta^1_j \\
&\quad + \left(\sum_{r=1}^2 K_{\sigma_1}(z_r,o^2)\gamma_r\right)^T \eta^2
\end{aligned}$$

We denote $\gamma = B^2(o^2)^{-1} \in \mathbb{R}$, the control of the second deformation module is the scalar

$$\begin{aligned}
\beta &= \sum_{i=0}^2 \left(\xi^i_{o^i}(\zeta^2_{o^2}(\gamma)\right)^T \eta^i \\
&= \sum_{i=0}^2 \left(\xi^i_{o^i}(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,\cdot)\gamma d_r)\right)\eta^i \\
&= \sum_{j=1}^3 \left(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,y_j)\gamma d_r\right)^T \eta^0_j + \sum_{j=1}^2 \left(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,z_j)\gamma d_r\right)^T \eta^1_j \\
&\quad + \left(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,o^2)\gamma d_r\right)^T \eta^2 \\
&= \sum_{j=1}^3 \left(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,y_j)\gamma d_r\right)^T \eta^0_j + \sum_{j=1}^2 \left(\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,z_j)\gamma d_r\right)^T \eta^1_j
\end{aligned}$$

since $\sum_{r=1}^3 K_{\sigma_2}(o^2+d_r,o^2)d_r = \sum_{r=1}^3 \exp -\frac{|d_r|^2}{\sigma_2^2} d_r = 0$ because vectors d_r all have the same norm and their sum equals 0. One can note here that momentum of deformation module M^2 does not influences its geodesic control. This could have been figured out from Remarks 23 and 28 since vector fields generated by its geometrical descriptor has

a null action on its geometrical descriptor.

6.4.4 Integration of trajectories

The Hamiltonian is:

$$\begin{aligned}\mathcal{H}(o,\eta,h) &= (\xi_o(\zeta_o(h))|\eta) - \tfrac{1}{2}c_o(h)\\ &= \textstyle\sum_{0\leq i,j\leq 2}(\xi^i_{o^i}(\zeta_{o^j}(h^j))|\eta^i) - \frac{1}{2}\sum_{i=0}^2 c^i_{o^i}(h^i)\\ &= \textstyle\sum_{r=1}^2\sum_{s=1}^3 K_{\sigma_1}(z_r,y_s)\eta_s^{0T}\alpha_r^1 + \sum_{1\leq r,s\leq 2} K_{\sigma_1}(z_r,z_s)\eta_s^{1T}\alpha_r^1\\ &\quad + \textstyle\sum_{r=1}^2 K_{\sigma_1}(z_r,o^2)\eta^{2T} + \sum_{r=1}^3\sum_{s=1}^3 K_{\sigma_2}(o^2+d_r,y_s)\beta\eta_s^{0T}d_r\\ &\quad + \textstyle\sum_{r=1}^3\sum_{s=1}^2 K_{\sigma_2}(o^2+d_r,z_s)\beta\eta_s^{1T}d_r\end{aligned}$$

where $o=(o^0,o^1,o^2)\in\mathcal{O}=\mathcal{O}^0\times\mathcal{O}^1\times\mathcal{O}^2$, $o^0=(y_1,y_2,y_3)\in\mathcal{O}^0=(\mathbb{R}^2)^3$, $o^1=(z_1,z_2)\in\mathcal{O}^1=(\mathbb{R}^2)^2$, $o^2\in\mathcal{O}^2=\mathbb{R}^2$, $h=(0,\alpha,\beta)\in H=H^0\times H^1\times H^3$, $\alpha=(\alpha_1,\alpha_2)\in H^1=(\mathbb{R}^2)^2$, $\beta\in H^2=\mathbb{R}$. We recall that $c^0=0$, $\zeta^0=0$, $\xi^0=0$ and $\xi_2\circ\zeta_2=0$.

From this expression of the Hamiltonian and the geodesic controls (see previous Section 6.4.3) it is easy to integrate forward and backward trajectories which enable to compute the energy and its gradient. However we saw in this example how fastidious it would be to compute all these equations each time we want to use a new combination of base-deformation modules. This is why our oriented-object implementation is particularly adapted. We only have to implement combination rules ones, and for each deformation module we only have to define its methods (defining functions ζ, ξ and c) ones. They depend on different parameters that need to be specified at each time, such as the scale σ or the number of local translations for instance. Then it is directly possible to compute an atlas with any combination of these deformation modules.

Chapter 7

Numerical experiments

Contents

In this chapter we study the use of our modular framework in four different situations. For each case we need to determine the deformation module to employ. As explained in previous chapters, we will build it as a combination of simple deformation modules. Therefore for each situation we need to determine a list of simple deformation modules, which are adapted to the shapes under study. For each deformation module of this list we need to set successively three types of parameters:

- •Type of deformations (ex: sum of local scalings)
- •Parameters of the corresponding deformation module (ex: number of scalings and their scale)

} Structure parameters

- •Initialization of geometrical descriptors (ex: centre of scalings) and the template (silent geometrical descriptor)

} Optimisation parameters

The first two types of parameters correspond to parameters of structure and define the deformation module, while the last type of parameters is a parameter of optimisation. The strategy to set these parameters in practice depends on the degree of constraints we want to incorporate in the deformation model. We explain briefly here how one can choose them when there is no prior.

- Type of deformations: by default when there is no prior on the deformation pattern, we use a combination of local translations, scalings and rotations at different scales.
- Parameters of the corresponding module:
  - Scale: by default we use a combination of deformation modules generating deformations at three or four different scales, the largest one is roughly equal to the size of the data, and the other scales are obtained by dividing successively this one by 2 or 3.
  - Number of centre: by default we initialize the centres on a regular grid with a step size depending on the scale (typically between once and twice the scale, depending on the sparcity we want). The number of centres results from this grid. One pitfall to avoid here is to constrain the deformations too much (for example by allowing too few centres of transformations) because the required deformation patterns might not be attainable with these constraints. Then a strategy would be to start with a step size for the grid for initialisations equal to the scale, and then to increase this size if seems necessary (for interpretability for instance).
- Initialization of geometrical descriptors: by default centres are initialized on a regular grid, and vectors have all their component equal to 1.

At each of these steps, it is possible to choose another initialization depending on a particular prior. It is also possible to build a list a deformation module via this strategy, and then to combine it with a particular deformation modules corresponding to a prior.

In the following examples, we will use several strategies to set these parameters: in Sections 7.1 and 7.3 we will use strong priors and therefore use the deformation modules which naturally correspond to these priors. In Section 7.2 we use "mixed-strategies": we incorporate constraints in the deformation model, but we initialize the parameters of the corresponding deformation modules (in particular centres of deformations) without prior. Last, in Section 7.4 we have no prior and then we use the strategy described above to set the parameters of the deformation module. In Section 7.4.2 we study the robustness of our results with respect to these initial choices.

7.1 Studying a population with different priors

We present here how simple priors can be used to study a population of shapes. We use simple examples of populations (curves with a hump at different locations) and we study the variability by forcing deformations to be of very simple types (displacement of the hump or folding and unfolding patterns). In the case where different patterns of deformations could explain the variability but one possesses an additional knowledge specifying which pattern should be used, we show that this knowledge can be easily incorporated in the deformation model.

7.1.1 Two equally plausible priors

In this section, we study the set of shapes presented in Figure 7.1: each shape has a hump, three of them have the hump rather on their left part, and three other have it on their right part, all at variable locations. Intuitively, there are two possible descriptions of the variability of this collection of shapes. One possibility is to consider that shapes derive from a “template” shape with one central hump by random translations of the hump in either direction. Another possibility is to consider that shapes derive from “template” shape by unfolding the hump in one place and fold a hump at another place. These two models of shape variability would explain the observed samples equally well, and the problem is undecidable without assuming priors on the solution. Determining the template reflecting the structure in the data set, and allowing to study its variability is a well-known problem in computer vision and shape analysis, as described in [KT97, MV07] for instance. If no point-correspondence is assumed, the current implementation of most statistical shape analysis techniques will give one or the other solution in an unpredictable way depending mostly on implementation choices, initialisation, regularisation techniques, etc..

We show here that our approach based on modular deformations allows the user to decide beforehand which solution he wants to favor. The addition a such a prior on the sought solution is possible by the design of relevant modules.

7.1.1.1 Displacement of the hump

To obtain a description of shape variability based on horizontal translations of the hump, we use a user-defined deformation module, which generates a vector field that is always

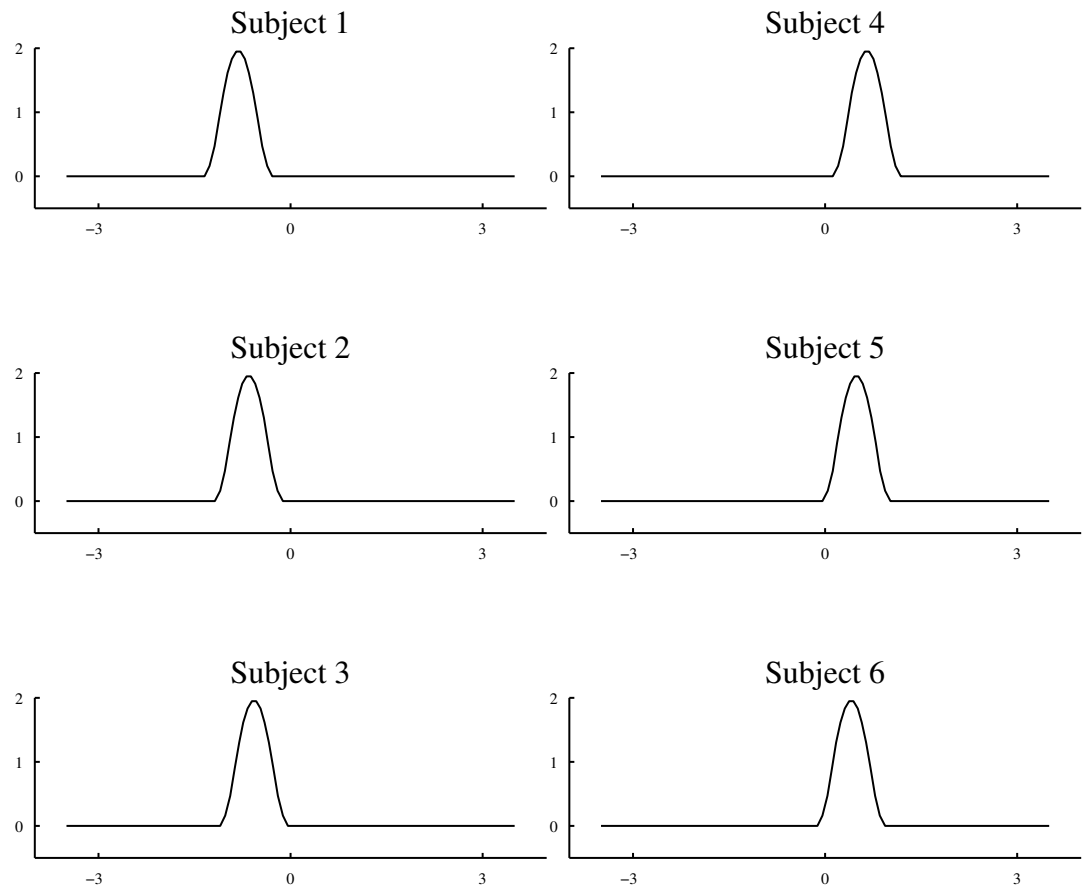


Figure 7.1: **Building an atlas using a prior**. Target shapes

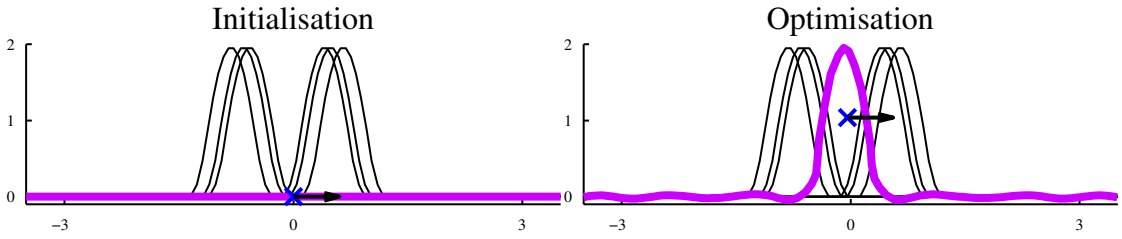


Figure 7.2: **Building an atlas using a prior.** Prior: horizontal displacement. **Template** at time $t = 0$ (purple curve) and other **geometrical descriptor** (blue cross). **Targets** in black. The black vector is the **fixed parameter** that defines the direction of the translation. Left: Before optimisation. Right: After optimisation.

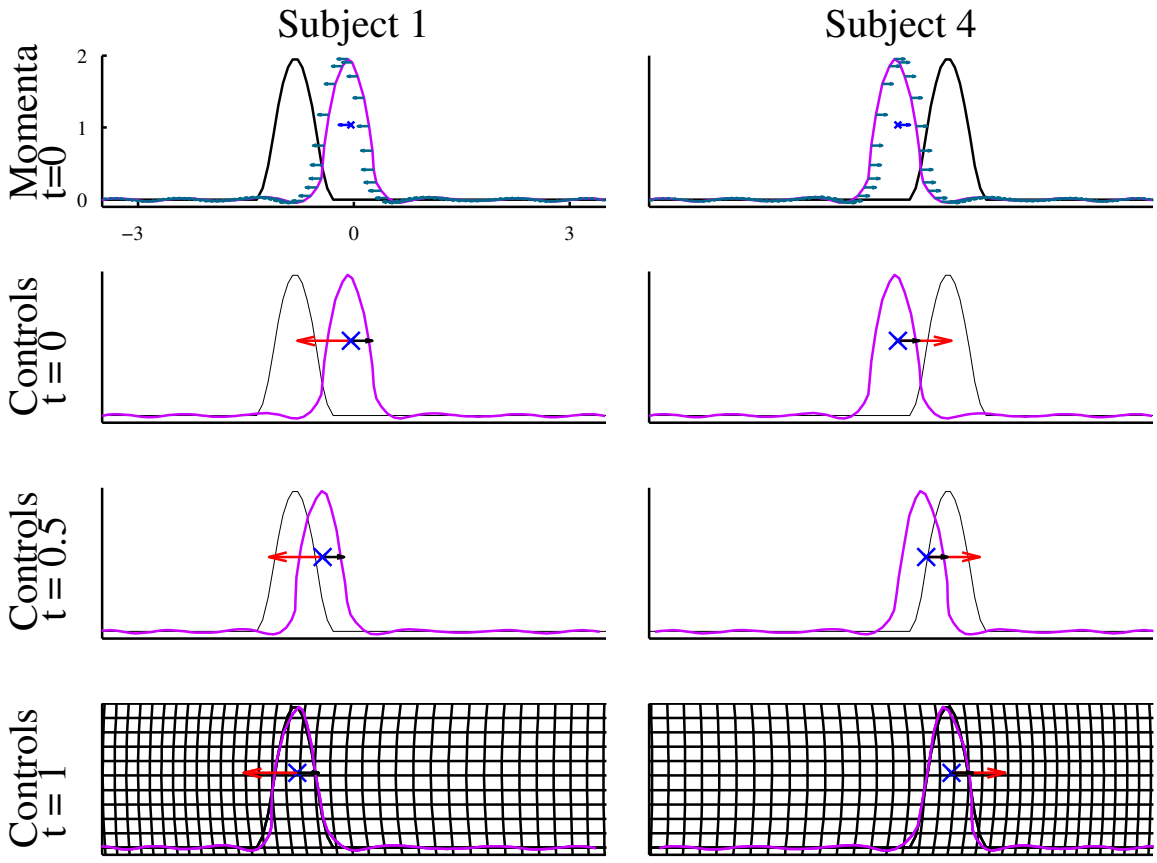


Figure 7.3: **Building an atlas using a prior.** Prior: horizontal displacement. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptor** at time t (blue cross). First line: **momenta** at $t = 0$ (in marine blue are momenta attached to the template's points and in blue the one attached to center of the translation with fixed direction). Three lower lines : the black vector is the fixed parameter that defines the direction of the translation, **controls** at t are represented by the length of the red arrow.

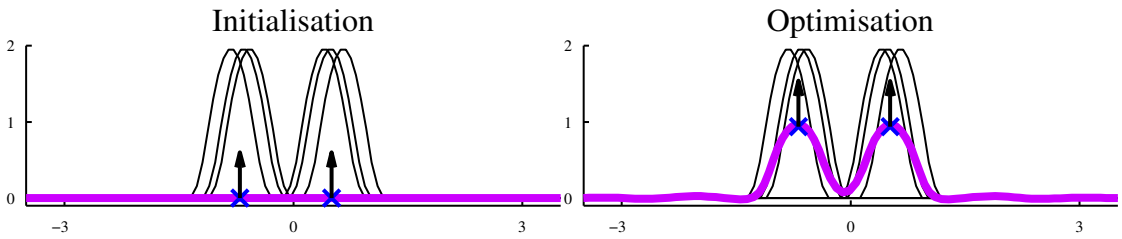


Figure 7.4: **Building an atlas using a prior.** Prior: vertical displacements. **Template** at time $t = 0$ (purple curve) and other **geometrical descriptor** (blue crosses). The black vector is the fixed parameter that defines the direction of the translation. **Targets** in black. Left: Before optimisation. Right: After optimisation.

a horizontal translation at a fixed large scale σ (see Section 4.2.2.3, here we use $\sigma = 3$ and a constant direction of translation). We initialise the template curve with a shape with no hump, include it into a silent module and combine it with the translation module. We minimize the functional (5.12), where μ is the varifold distance between the curves, which measures how well two curves are superimposed without the need to have point correspondence and consistent orientation [CT13]. The minimisation results is the optimal geometrical descriptors of the compound module, here the position of the vertices of the template curve and the base point of the translation, and one initial momentum per target shapes characterising the deformation of the template curve to the corresponding target shape. In Figure 7.2 (left part) we present the initialisation of the template and the geometrical descriptor, momenta η_0^k are initialized at zero. The fixed horizontal vector $u = (1, 0)$ is plotted in black, it is not optimised during the gradient descent and does not evolve during the integration of the trajectory of vector field. In Figure 7.2 (right part) we display the optimized shared parameters of optimal trajectories: the optimized template and position for the geometrical descriptor at time $t = 0$. The diffeomorphic deformation from the template to one target is parametrized by values at time $t = 0$ of the template, the geometrical descriptor and the momentum (dual variable of these quantities). We present on Figure 7.3, for two subjects, this parametrization of trajectories (first row), and also the transport of the template to targets with the geometrical descriptor and the control (last three rows). As the control is scalar, it is plotted as the length of the vector of the translation: the unit horizontal vector (fixed parameter) is plotted in black and in red is plotted this vector multiplied by the scalar control.

7.1.1.2 Folding and unfolding patterns

By contrast, we may decide to describe the variability in this same data set by using folding/unfolding pattern. The corresponding prior in the deformation model is encoded by one deformation module generating vector fields that are always a sum of two vertical translations at a fixed small scale σ (see Section 4.2.2.3, here we use $\sigma = 0.4$ and constant directions of translations). In Figure 7.4 can be seen the template and other

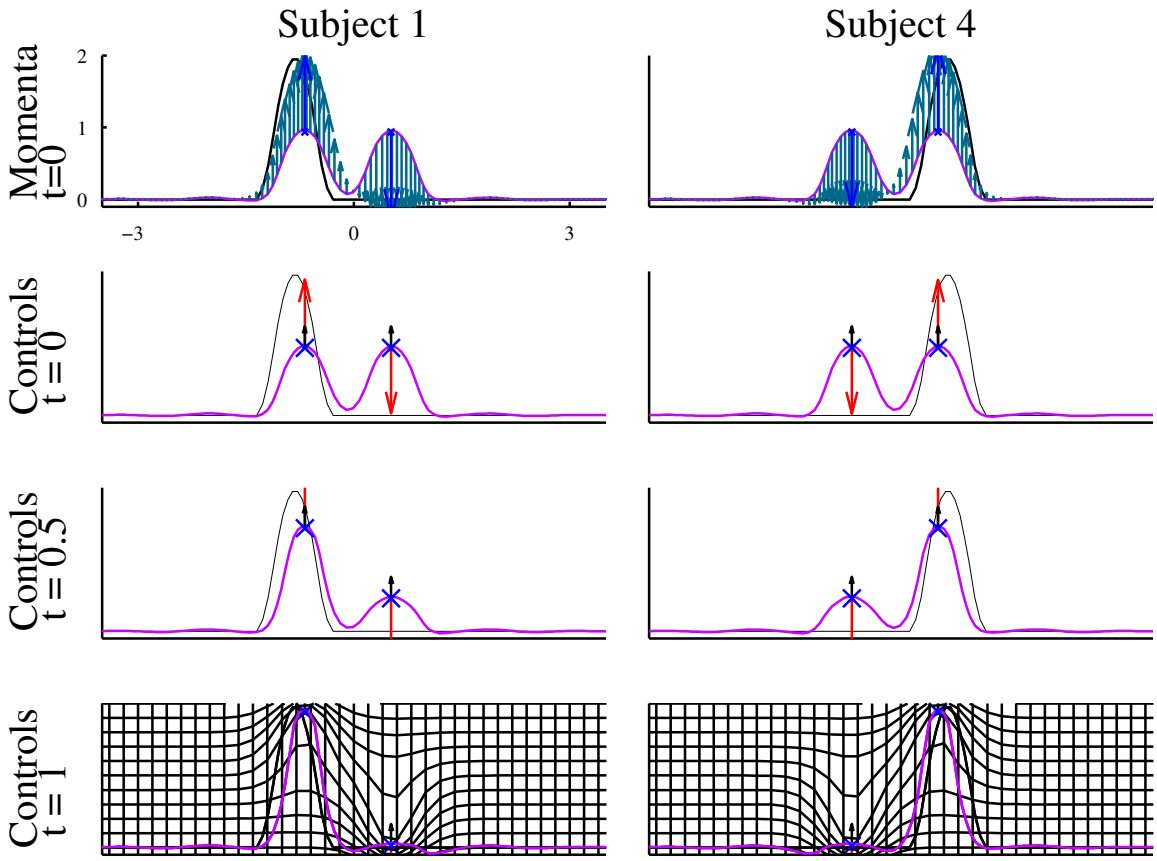


Figure 7.5: **Building an atlas using a prior.** Prior: vertical displacements. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptor** at time t (blue crosses). First line : **momenta** at $t = 0$ (in marine blue are momenta attached to the template's points and in blue the ones attached to centers of translations with fixed direction). Three lower lines : black vectors are the fixed parameter that defines the directions of translations, **controls** at t are represented by the lengths of red arrows.

geometrical descriptors before and after optimisation, as well as fixed vertical vectors . Parametrisation of minimizing trajectories, and trajectories of the template, geometrical descriptors and controls are plotted in Figure 7.5 for two subjects. As previously, controls can be seen in the lengths of vertical vectors, black vectors being fixed vertical unit vectors.

7.1.1.3 Fitting with data

In these two experiments the data term $\sum_{k=1}^{P} \frac{1}{\sigma^2}\mu\left(\phi^{o^k,\eta^k} \cdot f_{temp}, f^k_{target}\right)$ (see Equation (5.12)) decreases significantly during the optimisation process (divided respectively by 120 and 55). From these strong decreasing, we can deduce that both priors allow to study the variability amongst the population, even though the first one seems to explain a larger part of the variance.

These two experiments show that one may obtain different templates and deformations from the same data set by using different prior on shape variability. Our method allows the user to easily encode such prior in an intuitive and controlled manner by the design of suitable modules.

7.1.1.4 Recovering dimension of the population

Even though we have 6 subjects, our population is of dimension 1. Let us study whether this dimension can be recovered thanks to initial momenta. This would mean that these two priors allow to describe the variability of the population thanks to a parameter of dimension one which is not mandatory, even if the dimension of the population is one. For the first prior, however it is straightforward: the control is of dimension 1 and the cost is equal to the square of its norm, then the absolute value of the control is constant along optimal trajectories (because the cost is). Then, as it is a continuous trajectory, the control is constant. This shows that in this particular case, the initial value of the control determines the geodesic trajectory. We deduce that the space of geodesic trajectory (for a fix value of initial geometrical descriptor) is of dimension one and we will show that this property can be recovered in the initial momenta (which is of high dimension even though we know that the dimension of geodesic trajectory is one). For the second prior the control is of dimension 2 and then can be non constant even if its norm is. Therefore the space of geodesic trajectories is not of dimension one in this case. However, we will show now that the 6 initial momenta of our population lie in a subspace of dimension one.

In order to verify this, for each experiment, we compute a Principal Component Analysis on initial momenta, and we obtain the principal direction (one for each prior). In Figure 7.6 and 7.6 can be seen the result of shooting with these initial momenta. Then we compare the norm of each initial momentum with the norm of its orthogonal projection on the orthogonal of the principal direction (here we use the euclidean scalar product to perform these projections as we do not have any natural inner product on the cotangent space). For each subject, the norm of its projected initial momentum is always less than 2.10^{-8} times the norm of the momentum in the case of the first prior, and always less than 4.10^{-4} times the norm of the momentum in the case of the second

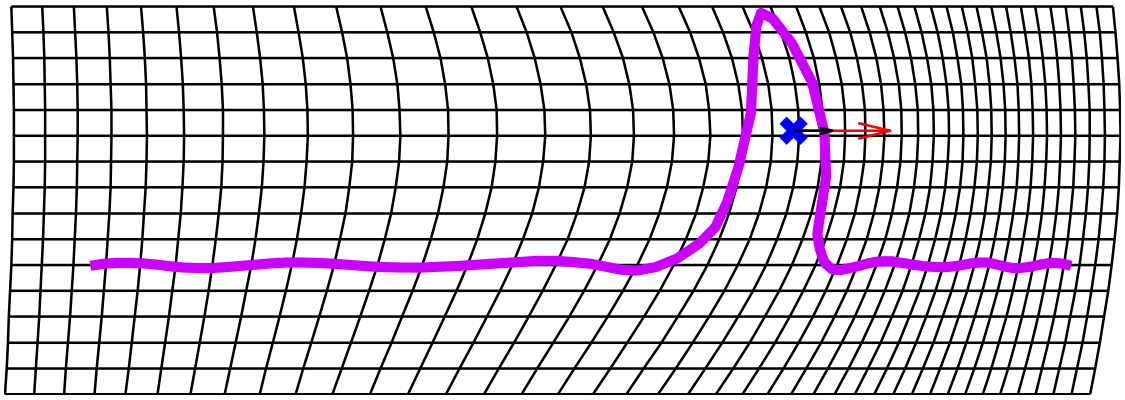


Figure 7.6: **Shooting in the principal direction.** Prior: horizontal displacements. **Template** at time $t = 1$ (in purple). Other **geometrical descriptor** at time $t = 1$ (blue cross). Black vector is the fixed parameter that defines the direction of translation, **control** is represented by the length of the red arrow.

prior. This shows that in both cases the space of initial momenta is in fact of dimension 1.

7.1.2 Another population: only one adapted prior

We present now the results obtained for another data set presented in Figure (7.8): here humps are at various locations. In Figure 7.9 is presented the initialisation and optimisation of the template and geometrical descriptor if we incorporate the first kind of prior in the deformation model, *ie* we use the deformation module generating always a horizontal translation. The template has one big centred hump and the data term is divided by 46 during the gradient descent. In Figure 7.10 are presented the same results but if we use the other kind of prior, *ie* the deformation module generating always a sum of two vertical translations. The optimised template seems less appropriate than the previous one. This is confirmed by the evolution of the data terms during the two gradient descents : it is only divided by 3. Here by comparing the decrease of the data term we understand that the first prior is more adapted to this data set, and then that the variability among the population is better described by horizontal displacements.

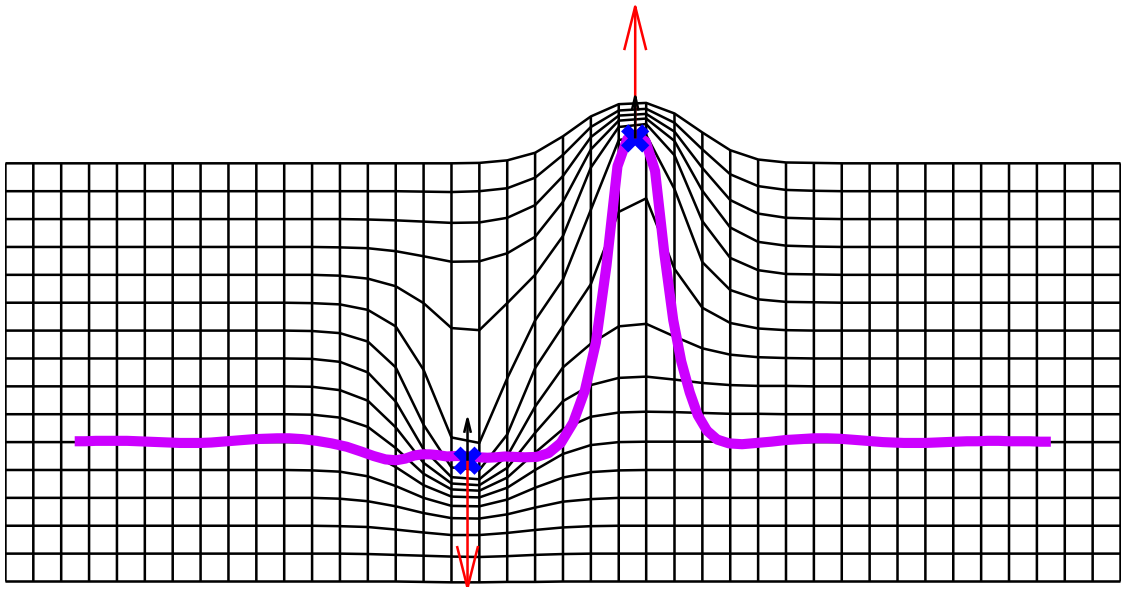


Figure 7.7: **Shooting in the principal direction.** Prior: vertical displacements. **Template** at time $t = 1$ (in purple). Other **geometrical descriptor** at time $t = 1$ (blue crosses). Black vectors are the fixed parameter that defines the directions of translations, **control** is represented by the lengths of the red arrows.

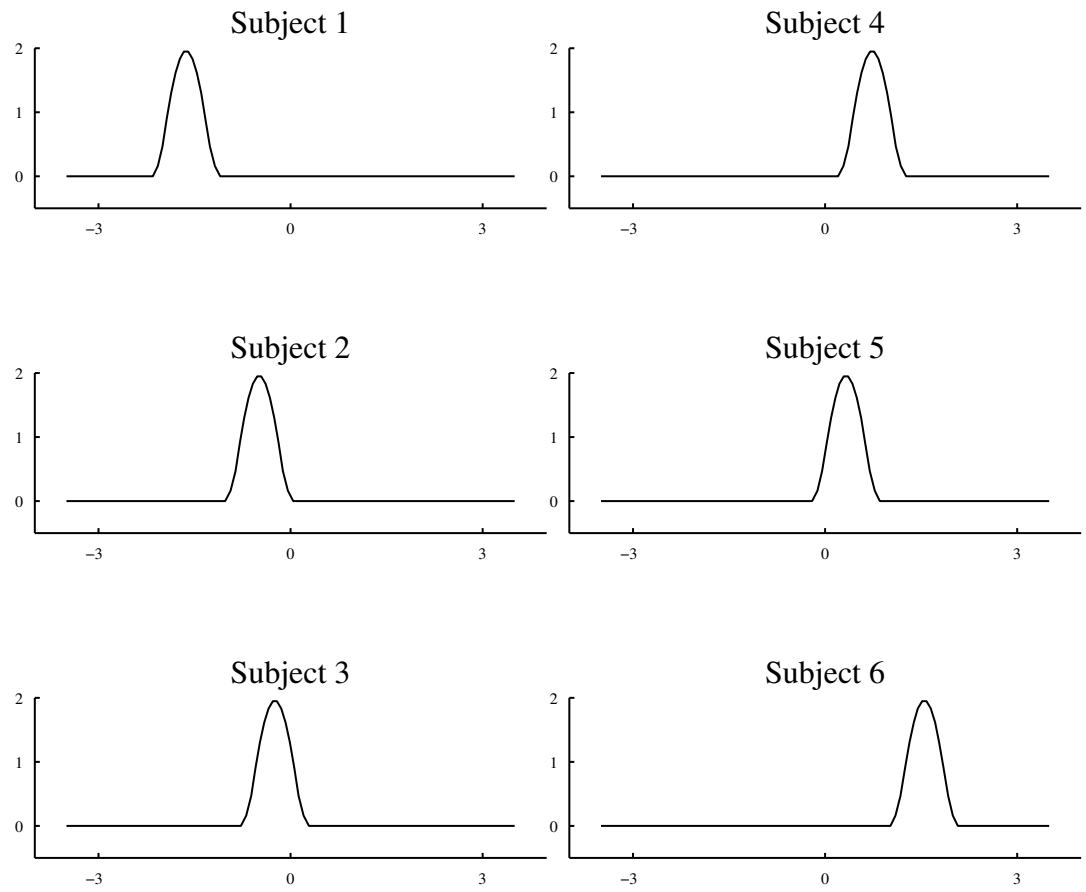


Figure 7.8: **Building an atlas using a prior.** A different data set.

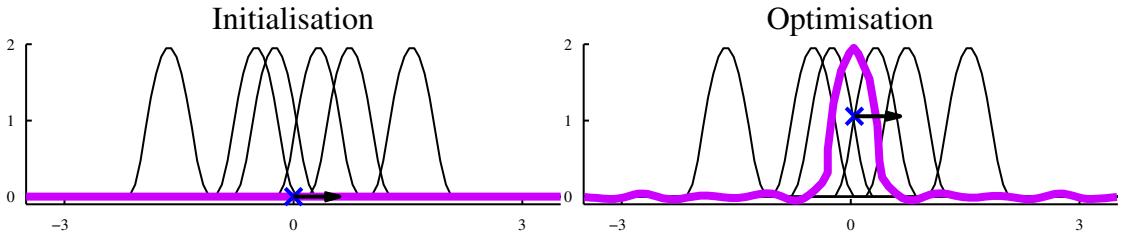


Figure 7.9: **Building an atlas using a prior.** Prior: horizontal displacement. **Template** at time $t = 0$ (purple curve) and other **geometrical descriptor** (blue cross). Black vector is the fixed parameter that defines the direction of translation. **Targets** in black. Left: Before optimisation. Right: After optimisation.

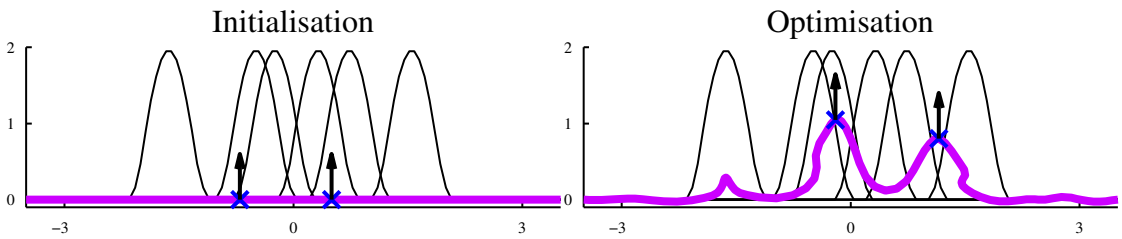


Figure 7.10: **Building an atlas using a prior.** Prior: vertical displacement. **Template** at time $t = 0$ (purple curve) and other **geometrical descriptors** (blue crosses). Black vectors are the fixed parameters that define the direction of translations. **Targets** in black. Left: Before optimisation. Right: After optimisation.

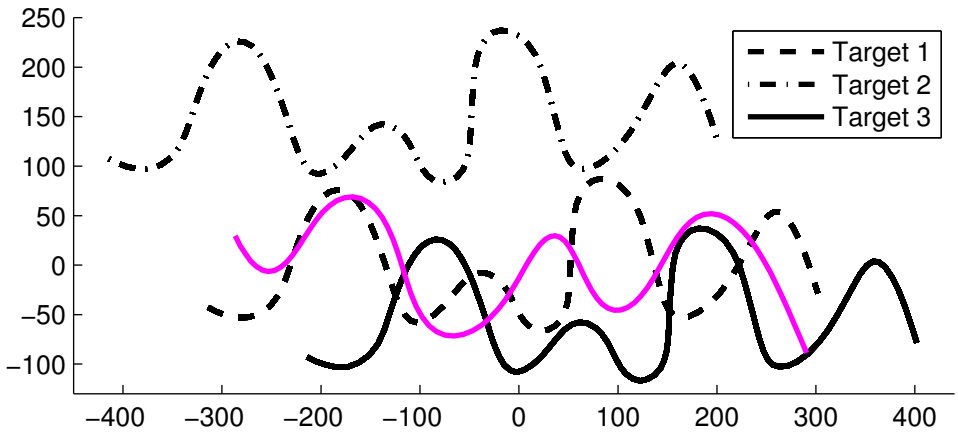


Figure 7.11: The source shape in purple and the three target shapes which will be considered in black.

7.2 A study of several priors with several imperfectly aligned target shapes

In this section we study the registration of two curves represented by currents (see Section 3.3.1), with different possibilities of rigid alignment of the target shape. In Figure 7.11 are plotted the three target shapes which will be considered, as well as the source shape. These three curves are simplified representations of one-dimensional section of hippocampi. Then, matching two hippocampi would be a generalisation of this 2D-matching in three dimension. In general before performing such a registration, one rigidly aligns the two shapes. However, this global rigid alignment may not align in a satisfying way all corresponding areas of the two hippocampi. For instance in Figure 7.12 one can see that even though global shapes are rigidly aligned, the alignment is not satisfying locally. This is the reason why we will study the registration of the source shape with these misaligned targets.
We will perform these registrations thanks to four different deformation modules in order to compare the stability of the deformation pattern they generate with variations of rigid alignments.

7.2.1 Presentation of the four deformation modules

7.2.1.1 Module M^1: sum of translations

The first deformation module M^1 that we consider generates a sum of local translations at scale $\sigma = 40$ (see Section 4.2.2.1). We fix this scale, and regularly initialize centres of translations (geometrical descriptors) around the Source shape. We obtain then a deformation module generating 60 local translations at scale 40.

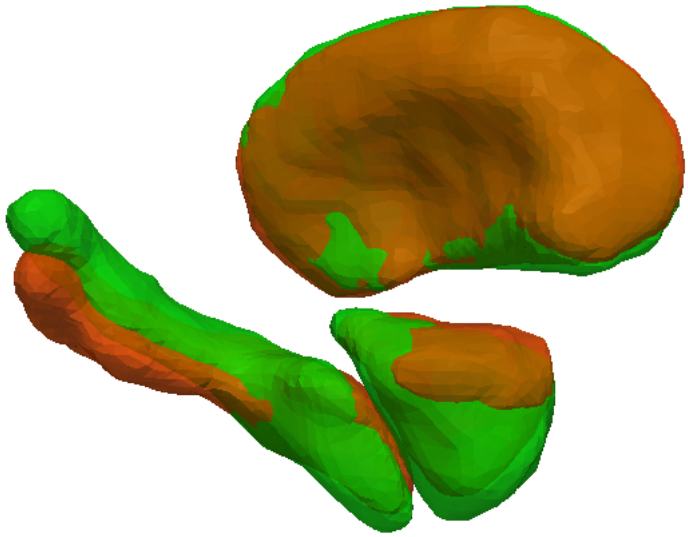


Figure 7.12: Hippocamp, amygdala and putamen for two subjects (green and orange) with a global rigid registration.

7.2.1.2 Module M^2: multi-scale sum of translations

The second deformation module which we consider is obtained through the combination (see Section 4.1.1.4) of the previous deformation module M^1 with the deformation module generating one translation at scale 600. This scale is large with respect to the size of curves so that the translation at scale 600 is a rigid one in the area of curves. The goal of this compound deformation module M^2 is to generate a multi-scale deformation, capturing both rigid and non linear deformations. The geometrical descriptor is now formed of the 60 centres of translations at scale 40 and the centre of the translation at scale 600.

7.2.1.3 Module M^3: studying deformation of humps

We build a third deformation module M^3 so that it will help studying the translations of humps and hollows, as well as their change of widths. To this end we build M^3 as the combination of the previous deformation module M^2 with two new deformation modules. The first one generates a sum of three anisotropic local translations with prior on directions (see Section 4.2.4) at scale 60 with coefficient of anisotropy equal to 1 (which means that each anisotropic translation is generated as a sum of three isotropic translations aligned in the direction of anisotropy). The direction of anisotropy is fixed to the vertical direction, the direction of the translation is fixed to the horizontal direction. Both are kept constant during the optimisation process as well as the integration of trajectories. The second deformation module that we combine generates a sum of three local anisotropic spreading (see Section 4.2.4) at scale 60 with coefficient of anisotropy also equal to 1. As for anisotropic translations, the direction of anisotropy is fixed to the vertical direction, the direction of spreading is fixed to the horizontal direction and

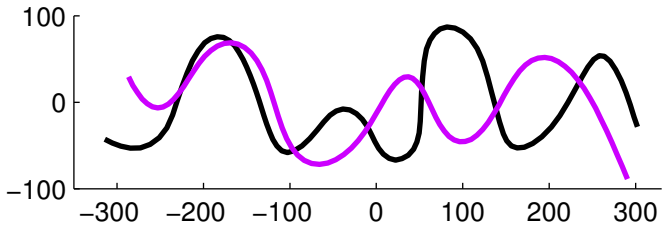


Figure 7.13: **Target 1.**. Source (in purple) and target (in black).

both are not transported by the flow.

7.2.1.4 Module M^4: dissociating horizontal and vertical movements

In the deformation module M^3, local horizontal displacements can be generated both by the isotropic translations at scale 40 and the anisotropic spreadings and translations at scale 60. In order to dissociate vertical and horizontal local displacements, we build a new deformation module M^4 as the combination of four deformation module: the first one generates one unconstrained translation at scale 600, the second generates a sum of four anisotropic spreadings at scale 60, the third generates a sum of four anisotropic translations at scale 60. Instead of combining these deformation modules with a deformation module generating unconstrained translations, we add translations with prior on the direction, constrained to generate vertical displacements. Then we have anisotropic horizontal displacements and isotropic vertical displacements, which seems to be a prior adapted to our data.

7.2.2 Results of these registrations

7.2.2.1 Target 1

We will first present the registrations of the source shape with the aligned target shape (target 1, see Figure 7.13) thanks to the four deformation modules. In these four cases we estimate an initial value of *active* geometrical descriptor and an initial momentum (dual variable of the active geometrical descriptors and the source shape). Momenta are always initialized at the null vector. All these variable are estimated thanks to a gradient descent (as presented in Chapter 6).

Module M^1. With the deformation module M^1, the initial geometrical descriptor to estimate is formed of 60 centres of the translations at scale 60. The momentum is formed of a dual variable of the source shape, and a dual variable of these centres of translations. In Figure 7.14 can be seen the initialisation of geometrical descriptors and their initial position after the gradient descent. They did not move much during the optimisation process. In Figure 7.15 we plot the initial momentum parametrizing the geodesic deformation in the first row. In other rows are plotted the evolution of the source shape during the integration of the flow, as well as the control (vectors of

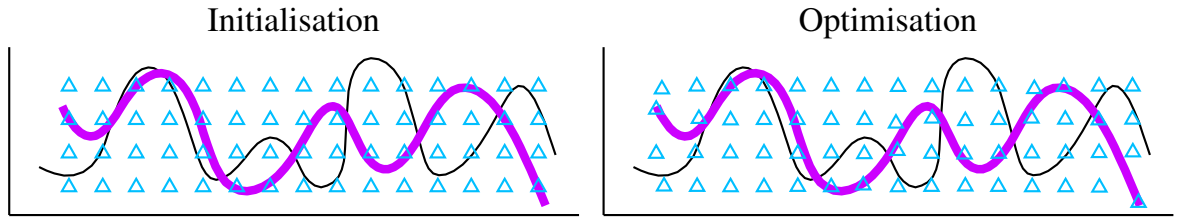


Figure 7.14: **Target 1, module M^1.** Source (in purple) and target (in black). △: geometrical descriptor (centres of translations). Left: before optimisation. Right: after optimisation.

translations). We can see that the matching is of good quality and that the deformation is intuitively satisfying.

Module M^2. With the deformation module M^2, the initial geometrical descriptor to estimate is formed of 60 centres of the translations at scale 60, and of the centre of the translation at scale 600. The momentum is formed of a dual variable of the source shape, and a dual variable of these centres of translations. In Figure 7.16 can be seen the initialisation of geometrical descriptors and their initial position after the gradient descent. In Figure 7.17 we plot the initial momentum in the first row. In other rows are plotted the evolution of the source shape during the integration of the flow, as well as the controls.
By comparing with the previous Section 7.2.2.1, we see that visually the same types of deformation is occurring, except for the creation of the small hump, and that the centres of translations at scale 40 have not moved during the optimisation process, similarly to Section 7.2.2.1. However the controls (vector of translations) are not the same. In particular with the deformation module M^2, the translation at scale 600 generates a large translation toward the right part of the curve and, in order to compensate, vector of translations at scale 40 on the left part generate small translations toward the left part. On the contrary, in Section 7.2.2.1, these translations have null vectors. But on the right hand of the curve, a translation to the right seems necessary and it is carried by small translations in Section 7.2.2.1, while it is carried by the large translation at scale 600 in the multi-scale framework and then vectors of small translations are in this part of smaller norm. Besides from the grid deformations of Figures 7.15 and 7.17, we can see that with M^2 the final deformation seems smoother (especially in the right-part of the grid).

Module M^3. With the deformation module M^3, the initial geometrical descriptor to estimate is formed of the centre of the translation at scale 600, 60 centres of the translations at scale 60 and the centres of the anisoptropic spreadings and translations. The momentum is formed of a dual variable of the source shape, and a dual variable of these centres. In Figure 7.18 can be seen the initialisation of geometrical descriptors and their initial position after the gradient descent. In Figure 7.19 we plot the initial momen-

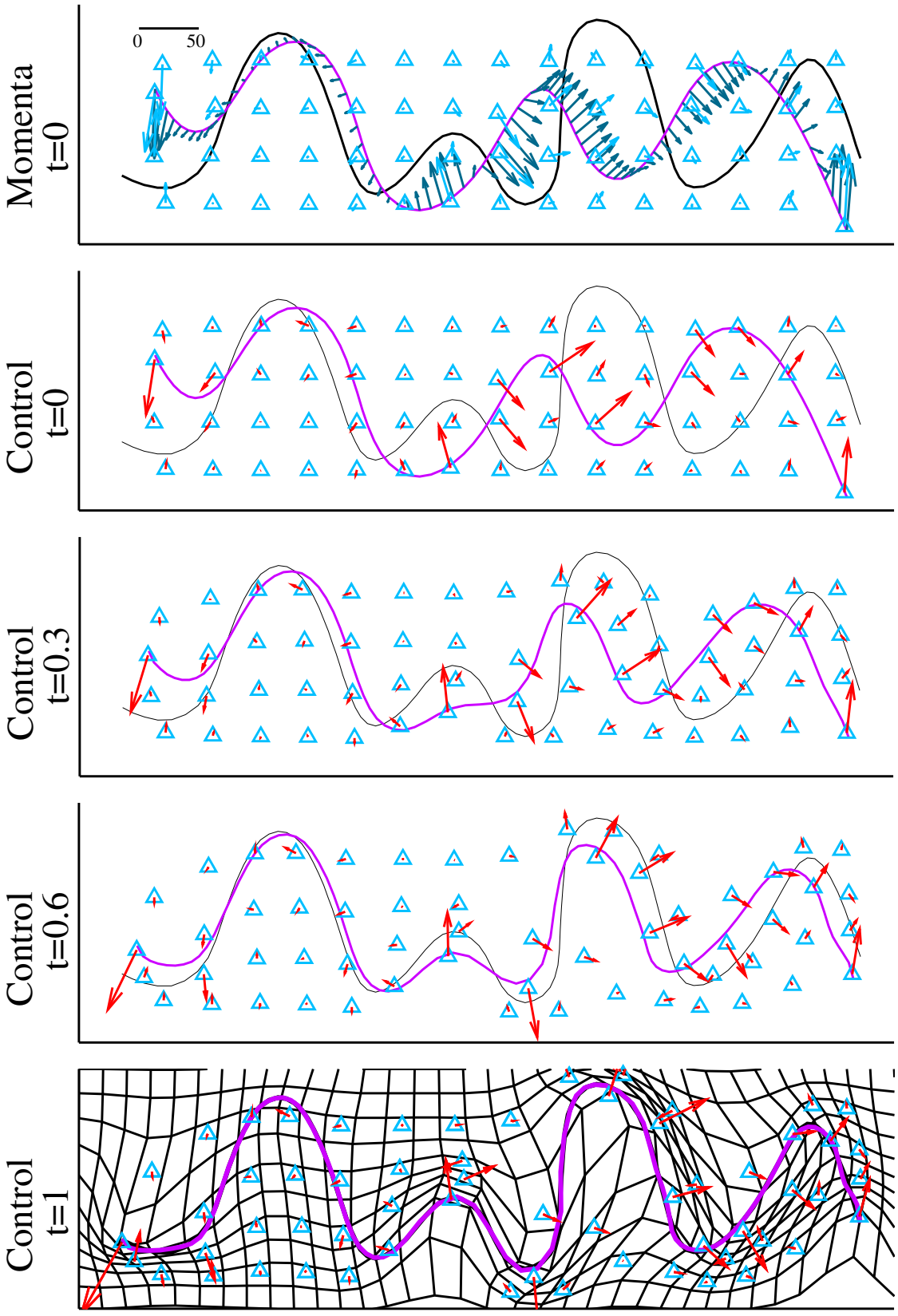


Figure 7.15: **Target 1, module M^1.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations). First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

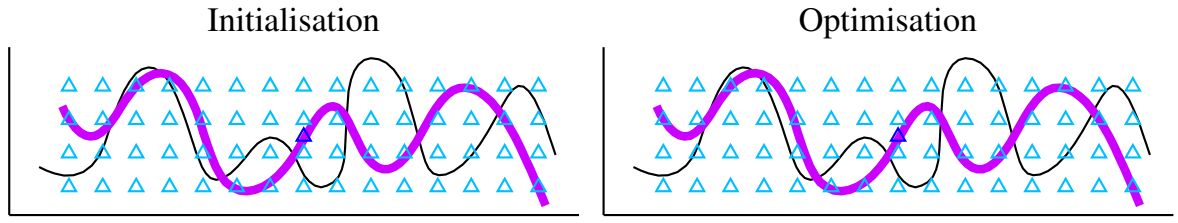


Figure 7.16: **Target 1, module M^2.** Source (in purple) and target (in black). △: geometrical descriptor (centres of translations), scales: 600 and 40. Left: before optimisation. Right: after optimisation.

tum in the first row and, in other rows, the evolution of the source shape during the integration of the flow, as well as the controls. For each anisotropic translation, the corresponding geometrical descriptor is a point associated with 2 directions (anisotropy and translation), and the corresponding control is a vector of dimension 2. The corresponding vector field is generated thanks to 3 points (one equal to the geometrical descriptor, one above and one below, since the direction of anisotropy is vertical and the anisotropic coefficient is one) represented by black circles and we plot the control vector attached to these three points for visualisation purpose. For each anisotropic spreading, the corresponding geometrical descriptor is a point associated with 2 directions (anisotropy and translation), and the corresponding control is a scalar. The corresponding vector field is generated thanks to 9 points (vertices of three triangles which are aligned in the vertical direction of anisotropy) represented by black crosses and 9 vectors attached to these points (see Section 4.2.4.2). Here as directions of anisotropy and spreading are orthogonal, vectors attached to the top of the triangles are always null, all these attached to the bottom-right vertices are equal and all these attached to the bottom-left vertices are the opposite of these attached to the bottom right. The control is represented by the length and direction of these 6 non-null vectors.

We can see that this deformation module allows also a good registration. However, only one anisotropic transform seems really useful: the translation with prior on the direction at the extreme right. Here non linear patterns are rather generated thanks to the local unconstrained translations.

Module M^4. With the deformation module M^4, the initial geometrical descriptor to estimate is formed of the centre of the translation at scale 600, 60 centres of the translations with prior on the direction at scale 60 and the centres of the anisoptropic spreadings and translations. The momentum is formed of a dual variable of the source shape, and a dual variable of these centres. In Figure 7.20 can be seen the initialisation of geometrical descriptors and their initial position after the gradient descent.

In Figure 7.21 we plot the initial momentum in the first row. In other rows are plotted the evolution of the source shape during the integration of the flow, as well as the controls. Controls of anisotropic spreadings and translations will be represented in the same way as in previous section. Similarly, for each isotropic vertical translation, the

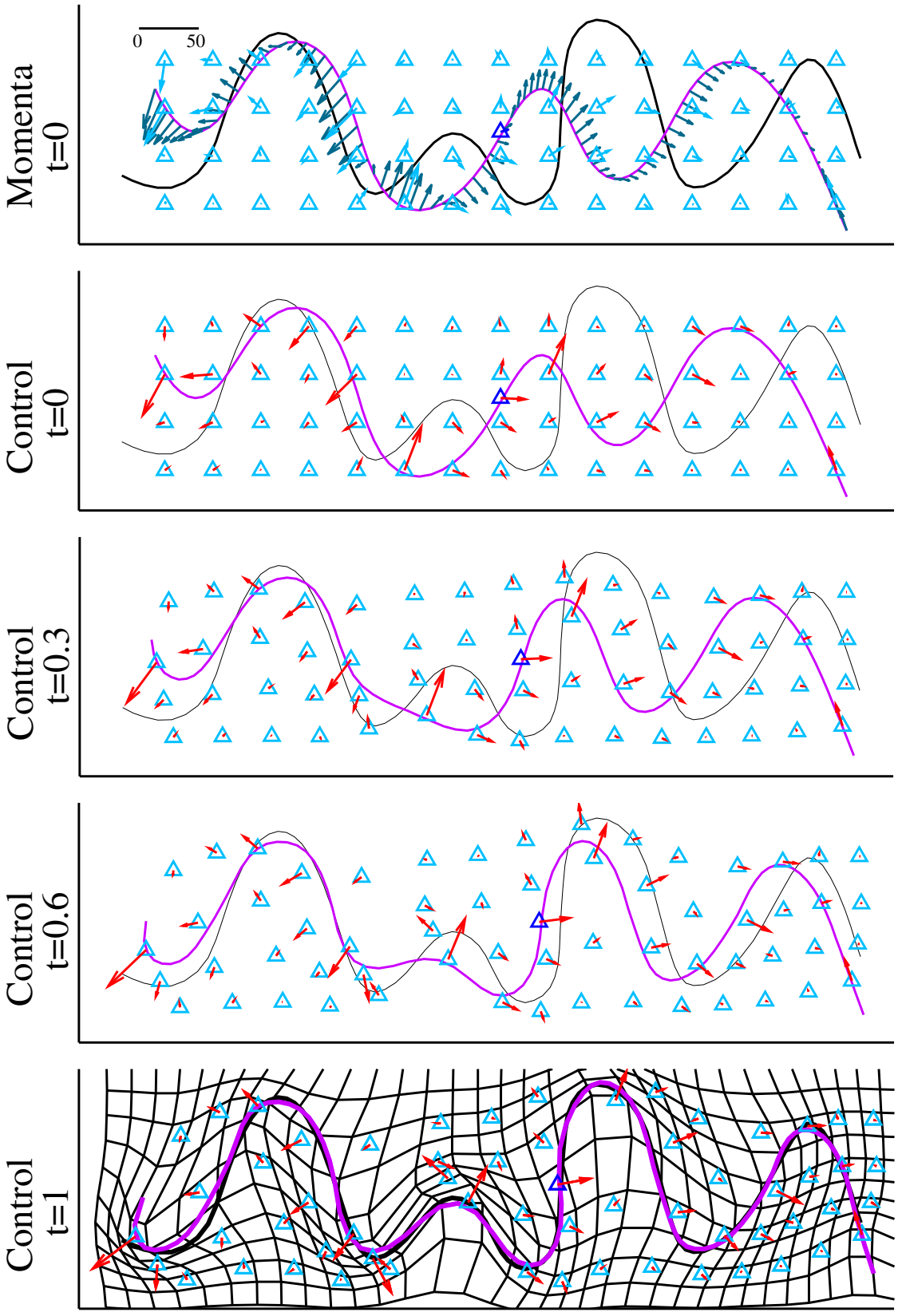


Figure 7.17: **Target 1, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

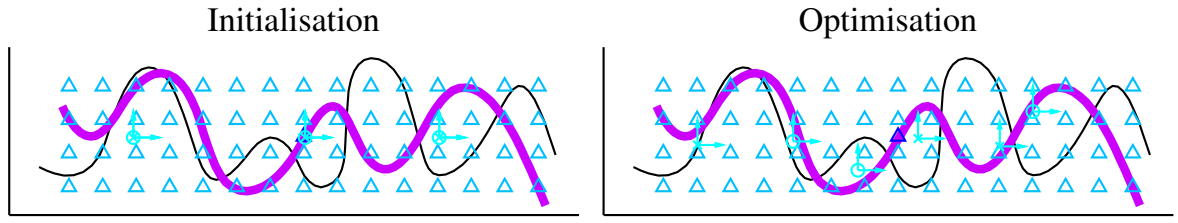


Figure 7.18: **Target 1, module M^3.** Source (in purple) and target (in black). Geometrical descriptor: $\triangle$ (centres of translations, scale 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). Left: before optimisation. Right: after optimisation.

corresponding control is a scalar and it will be represented by the length and direction of the vertical vector attached to the corresponding point (geometrical descriptor). We can see that this deformation module also allows good registration. Here deformations are more constrained and a combination of allowed displacements are used in order to generated complex movements. For instance with the three previous deformation modules, the small hump is created thanks to an oblique translation while with this more constrained one, the small hump is generated thanks to the combined action of a vertical translation and an anisotropic spreading.

7.2.2.2 Target 2

We will now present the registrations of the source shape with the second target (see Figure 7.22), using the same deformation modules as previously. Geometrical descriptors and momenta are initialized similarly to the previous Section. Results are presented in Figures 7.23, 7.24, 7.25 and 7.26. It is clear here that the deformation module M^1 generating a sum of translations (Figure 7.23) is not satisfying. The problems seems to come from the current measure: folds are created, so that their norm is null for currents. Besides we see that the deformation pattern is very complicated. This is due to the fact that sums of local translations can generate very complex local transforms and as the current measure is not adapted here, during the gradient descent converges to a solution minimizing the energy but in a way which is not satisfying. On the contrary, M^2 (see Figure 7.24) performs a very good matching. Here allowing a movement at a large scale enables translations at the smaller scale to generate satisfying local displacements. The third deformation module M^3 (see Figure 7.25) does not perform good registration. This is not intuitive because this deformation module is a combination of the one generating the multi-scale sum of translations with an other deformation module: it should be able to perform registration at least as good as M^3. This shows that local (but not global) minima can be a real issue. The last deformation module M^4 (see Figure 7.26) allows to perform a good registration: the addition of constraints (translations at scale 40 are constrained to be vertical) enables to obtained a more satisfying result.
We can see on Figures 7.24 and 7.26 that we obtain the same pattern of deformations

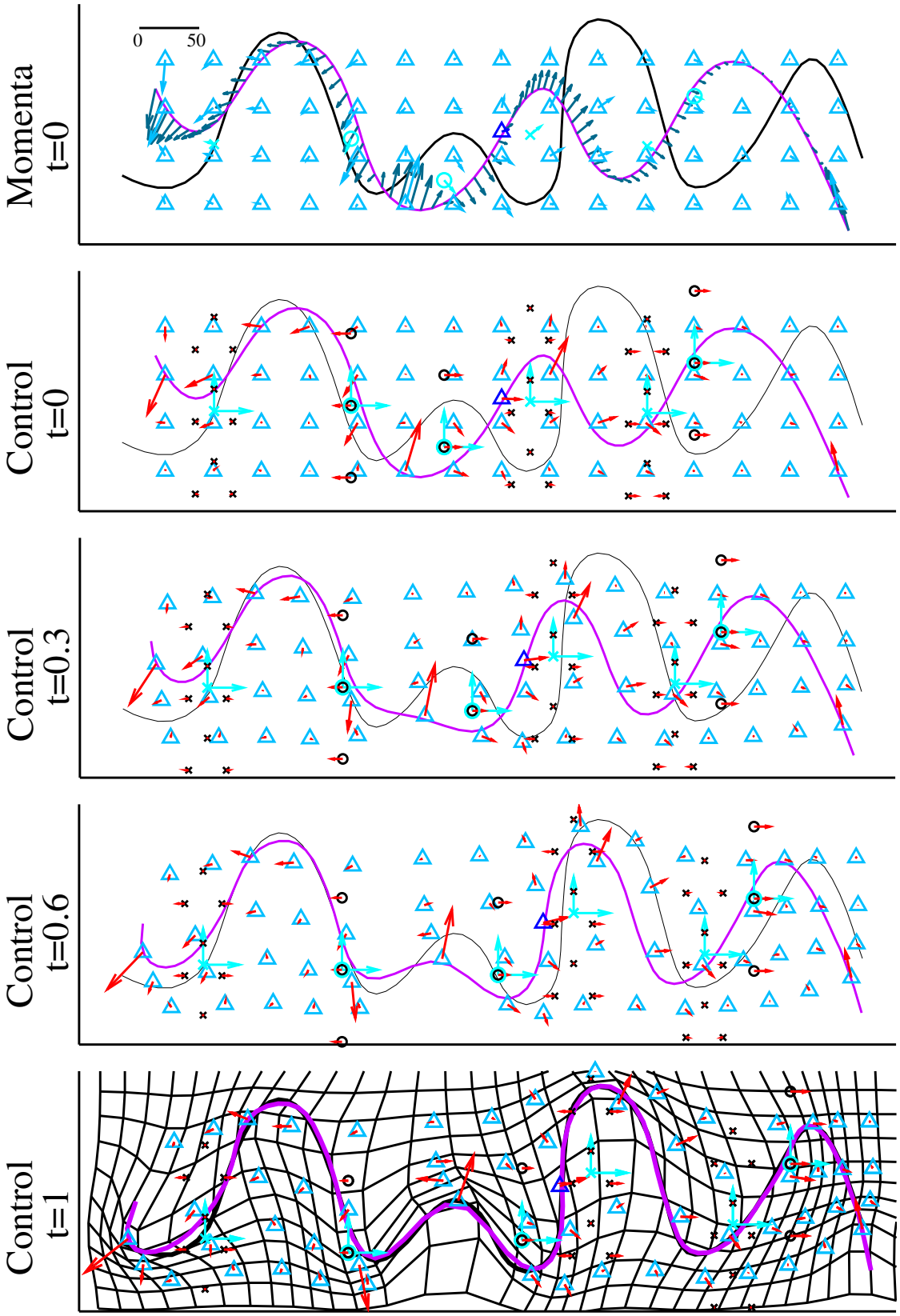


Figure 7.19: **Target 1, module M^3.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

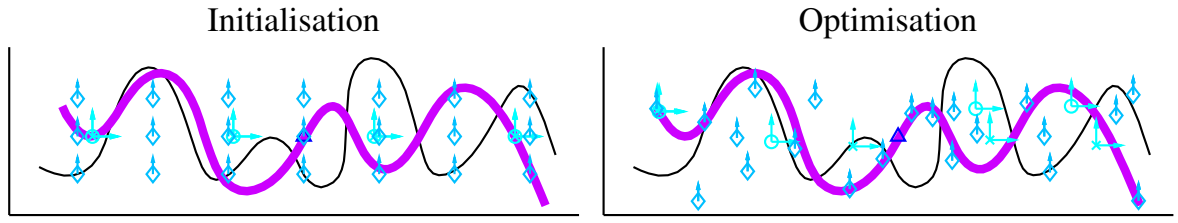


Figure 7.20: **Target 1, module M^4.** Source (in purple) and target (in black). Geometrical descriptor: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\Diamond$ (isotropic translations with prior on direction, scale 40). Left: before optimisation. Right: after optimisation.

as in the previous section when shapes were centred.

7.2.2.3 Target 3

We study now the registration of the source shape and the third target (see Figure 7.27). We show the results obtained with deformation modules M^2 and M^4 as, from the previous section, they are the only two susceptible to perform good registrations. Results are presented in Figures 7.28 and 7.29. We see that here they are not able to perform a good matching: they both fall in a local minimum.

7.2.2.4 Conclusion

These four deformation modules seem to allow good registration when shapes are well aligned. However, when shapes are displaced, optimised deformation patterns are no longer satisfying. This seems to be due to a too small scale for the current distance: the right part of the target curve is not taken into account because it is too far from the source shape with respect to other parts of the target shape. This is why in the next section we study a multi-scale data-attachment term.

7.2.3 A multi-scale data attachment term

In previous experiments we used the same kernel for currents for all experiments: it is equal to 50. However it seems clear that when the two shapes are far from each other, this scale does not allow to see long distance differences. Then some parts of the target curve might not be taken into account. On the contrary, if the scale of currents is to big, fine differences in hump patterns will not be taken into account. This suggests that we should use a multi-scale data attachment term. We perform here a common approximation of this idea: we change the scale of the currents at the middle of the optimisation process. This is not totally satisfying because we change the functional to minimize. However, this can be seen as an approximation of a data attachment term obtained by summing the distances obtained with two scales of kernels: in the first

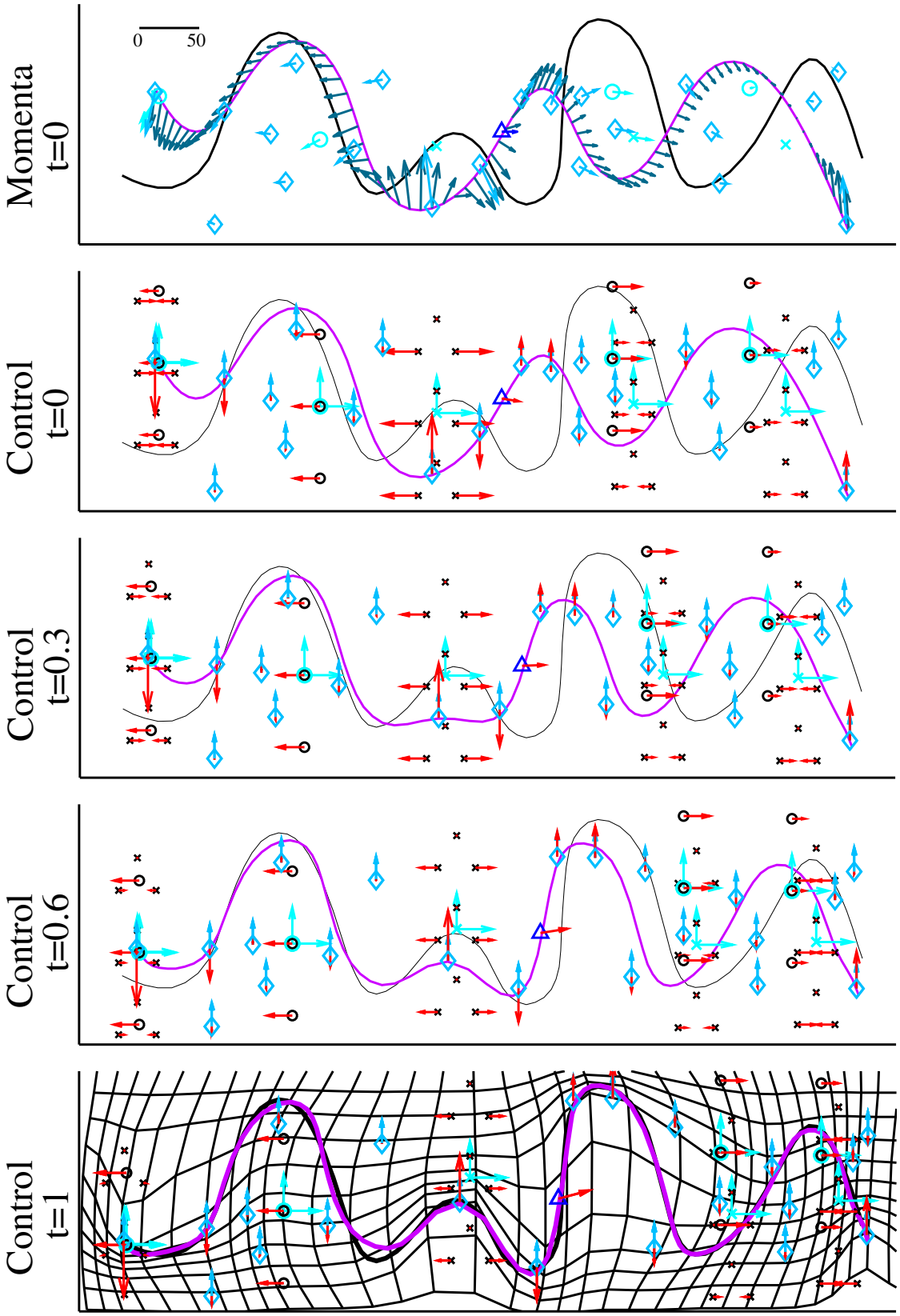


Figure 7.21: **Target 1, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations, scale 600), blue vectors attached to × (anisotropic spreading, scale 60), blue vectors attached to ∘ (anisotropic translations with prior on direction, scale 60), blue vectors attached to ♢ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black (× for spreading, ∘ for translations with prior on direction) and **Control** at time t in red.

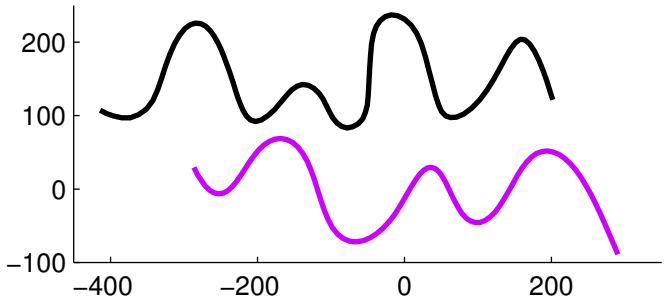


Figure 7.22: **Target 2.** Source (in purple) and target (in black).

phase, the two shapes are too far from each other and then only the distance for the large scale in non zero, while in the second phase shapes are close to each other and the distance for the large scale is close to zero whereas the distance for the small scale is now non zero. We emphasize here that even though we change the scale of currents, the scale of deformations do not change: it does not correspond to performing first rigid and then non linear registration. We still optimize simultaneously deformations at small and large scale. We present here the results for the three possible target shapes and the four deformation modules, we use two scales for currents: first we set it to 100 and after a fixed number of iterations (here 500) we change it to 30.

7.2.3.1 Target 1

We first present the result of this multi-scale data attachment term with the centred data since we need to ensure that we obtain the same result (*i.e.* same optimized geometrical descriptor and trajectory of controls) as when we directly register with a small scale for currents. For deformation module M^1, the result is presented in Figure 7.34 and one can see that it is similar to the one obtained Section 7.2.2.1. For the second deformation module M^2 (see Figure 7.35), results are also similar even though in this case of multi-scale data attachment term some controls are larger and enable to perform a registration slightly more efficient. For the third deformation module M^3 (see Figure 7.36), the result is less satisfying: geometrical descriptor of anisotropic spreadings and anisotropic translations are not optimized the same way. However, geometrical descriptors of isotropic translations, as well are their controls, are very similar to Section 7.2.2.1. This seems to be due to the fact that these anisotropic deformations do not have a large influence on the global deformation: most of the transform is generated by the isotropic translations. In this sense, it is not surprising that their is no stability on the optimisation of geometrical descriptors for anisotropic spreadings and translations. On the opposite for deformation module M^4 (see Figure 7.37) we recover the stability in optimisation: as translations at scale 40 are constrained to be vertical, horizontal displacements need to be generated thanks to anisotropic spreading and translations.

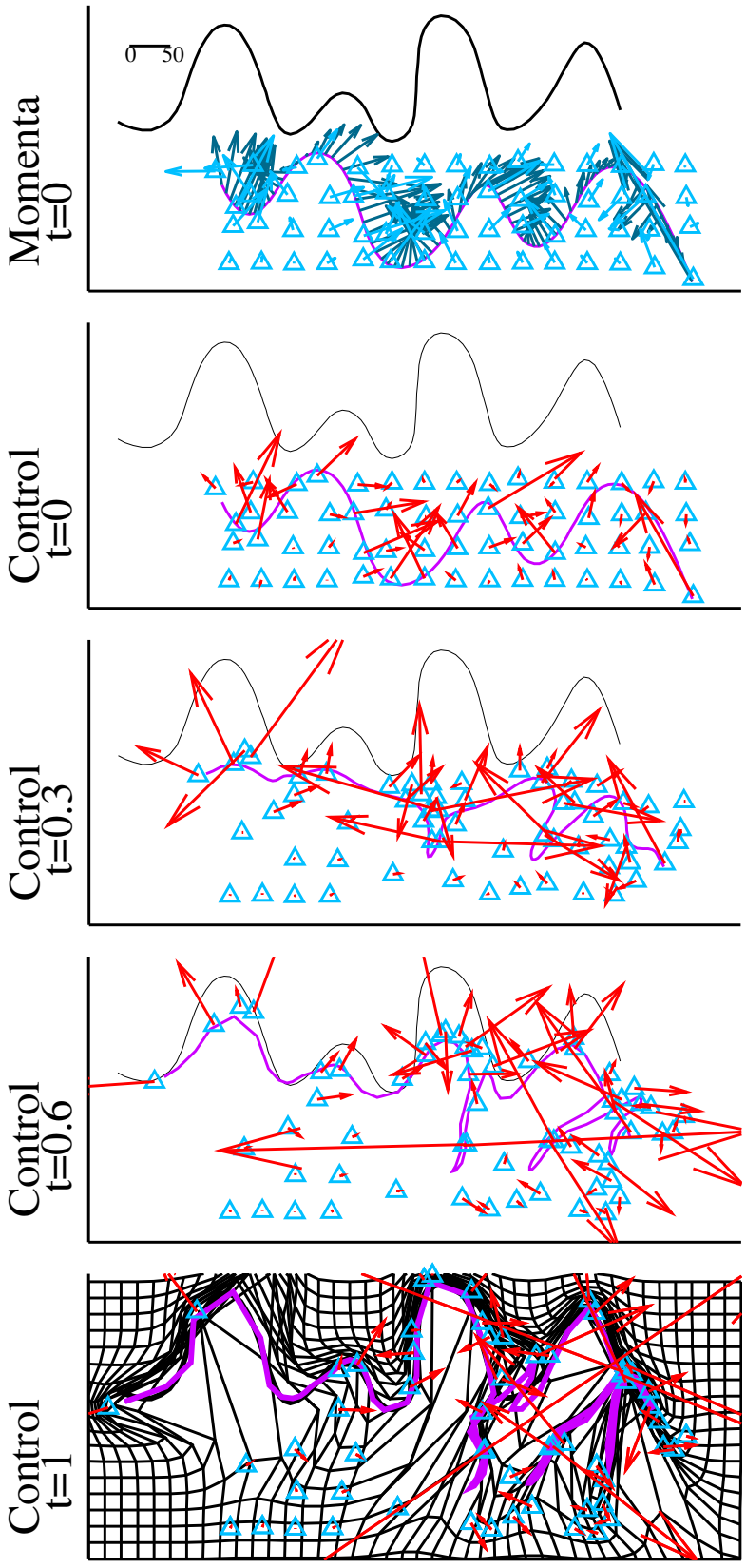


Figure 7.23: **Target 2, module M^1.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations). First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

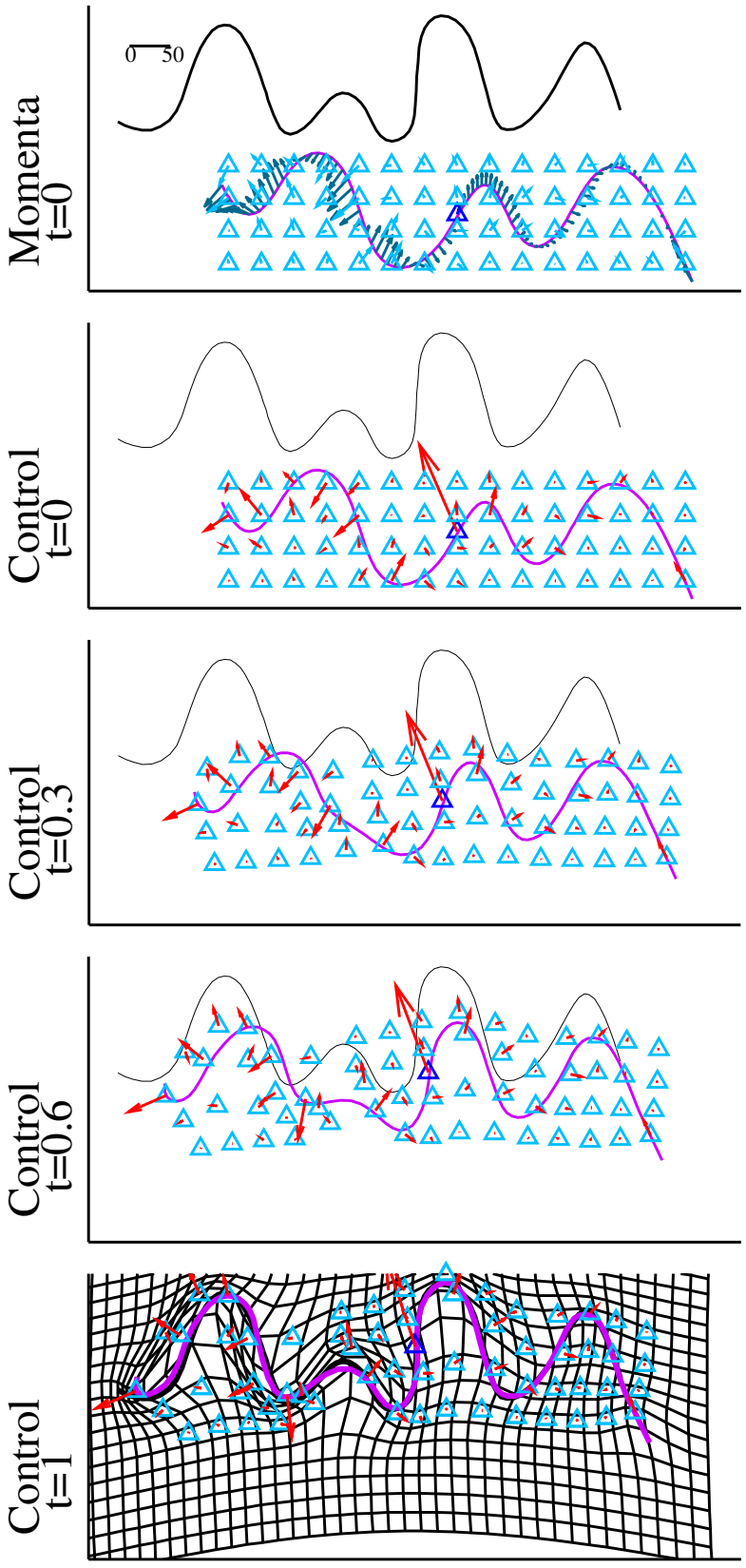


Figure 7.24: **Target 2, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

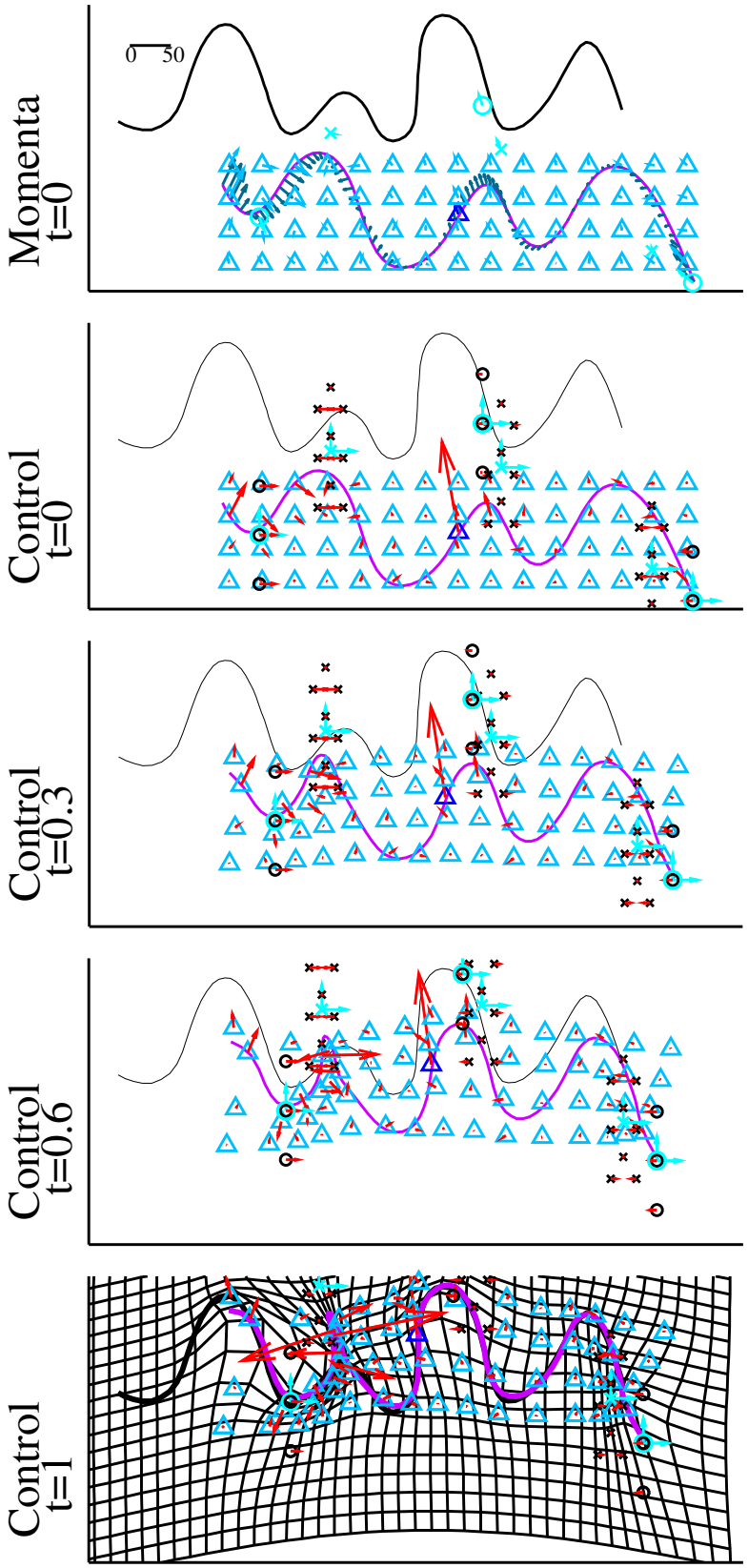


Figure 7.25: **Target 2, module M^3.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

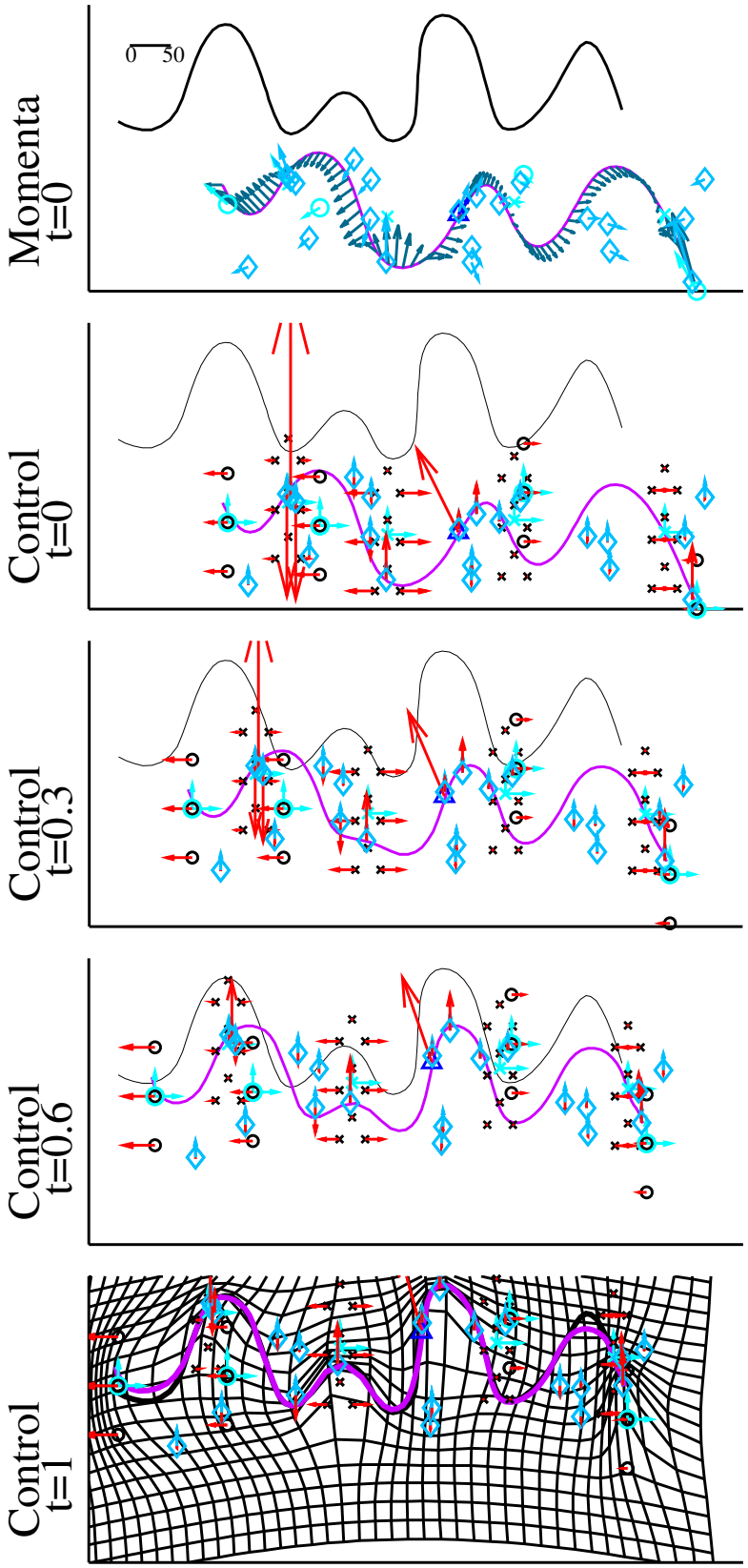


Figure 7.26: **Target 2, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\Diamond$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

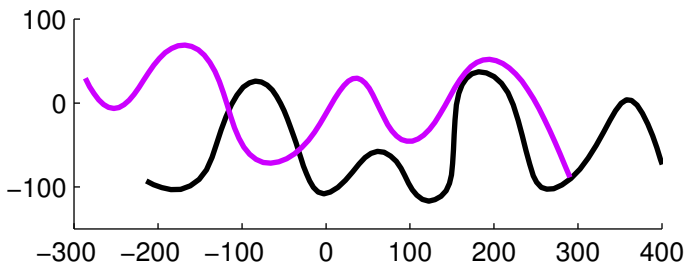


Figure 7.27: **Target 3.** Source (in purple) and target (in black).

7.2.3.2 Target 2

We present here the result we obtain with this multi-scale data-attachment term for shapes of Section 7.2.2.2. In Figure 7.34 we see that the first module M^1 still does not allow to perform a satisfying registration. On the contrary, in Figure 7.36, one can see that now the trajectory obtained thanks to M^3 is satisfactory: the multi-scale data attachment term enables to use the multi-scale deformations. The two other deformations modules M^2 (see Figure 7.35) and M^4 (see Figure 7.37) are still satisfying. Besides, optimisation of geometrical descriptors are similar but trajectories of controls are different for some translations at scale 40 (unconstrained for M^2 and constrained to be vertical for M^4) and the registration is of slightly better quality with these multi-scale data attachment term. However, same patterns of deformations are used to match the source shape into the target shape, and they are generated in similar ways with the two data attachment terms.

7.2.3.3 Target 3

As a last example, we present the result we obtain with this multi-scale data-attachment term for the target shape of Section 7.2.2.3. In Figure 7.38 we see that similarly to the previous example, the first module M^1 does not allow to perform a satisfying registration. For the three other deformation modules (see Figures 7.35, 7.40 and 7.41). This shows that a multi-scale data attachment term is more natural in case of multi-scale deformations.

7.2.4 Interpretation of optimized parameters

We will now study how parametrization of trajectories can be interpreted, by comparing the different results we obtained for the three different target shapes for deformation modules M^2, M^2 and M^3. The deformation module M^1 does not give satisfying registrations and then it will not be considered here. We will only consider results obtained with multi-scale data-attachment term because from the results we presented earlier, it is the only one allowing good registrations.

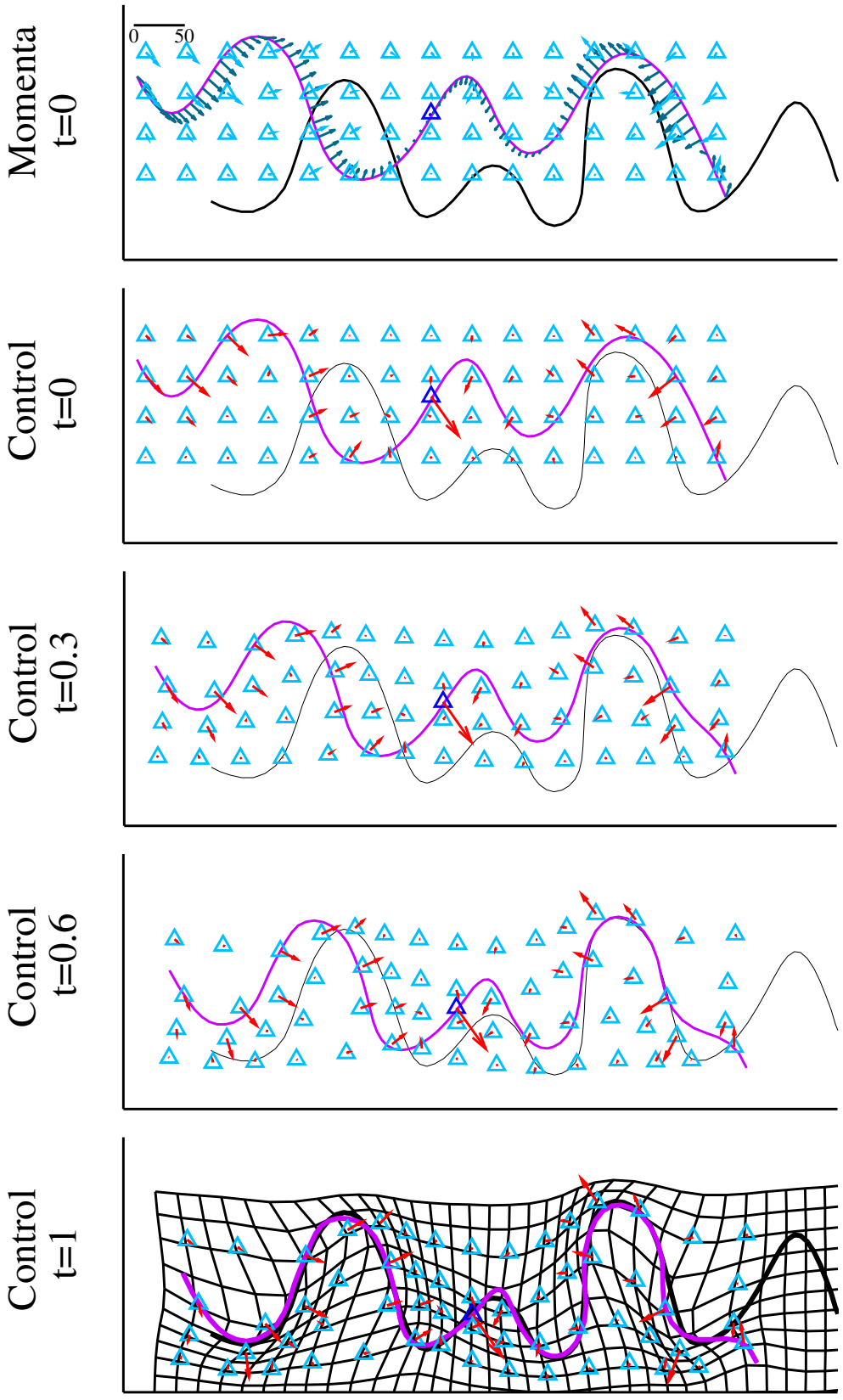


Figure 7.28: **Target 3, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

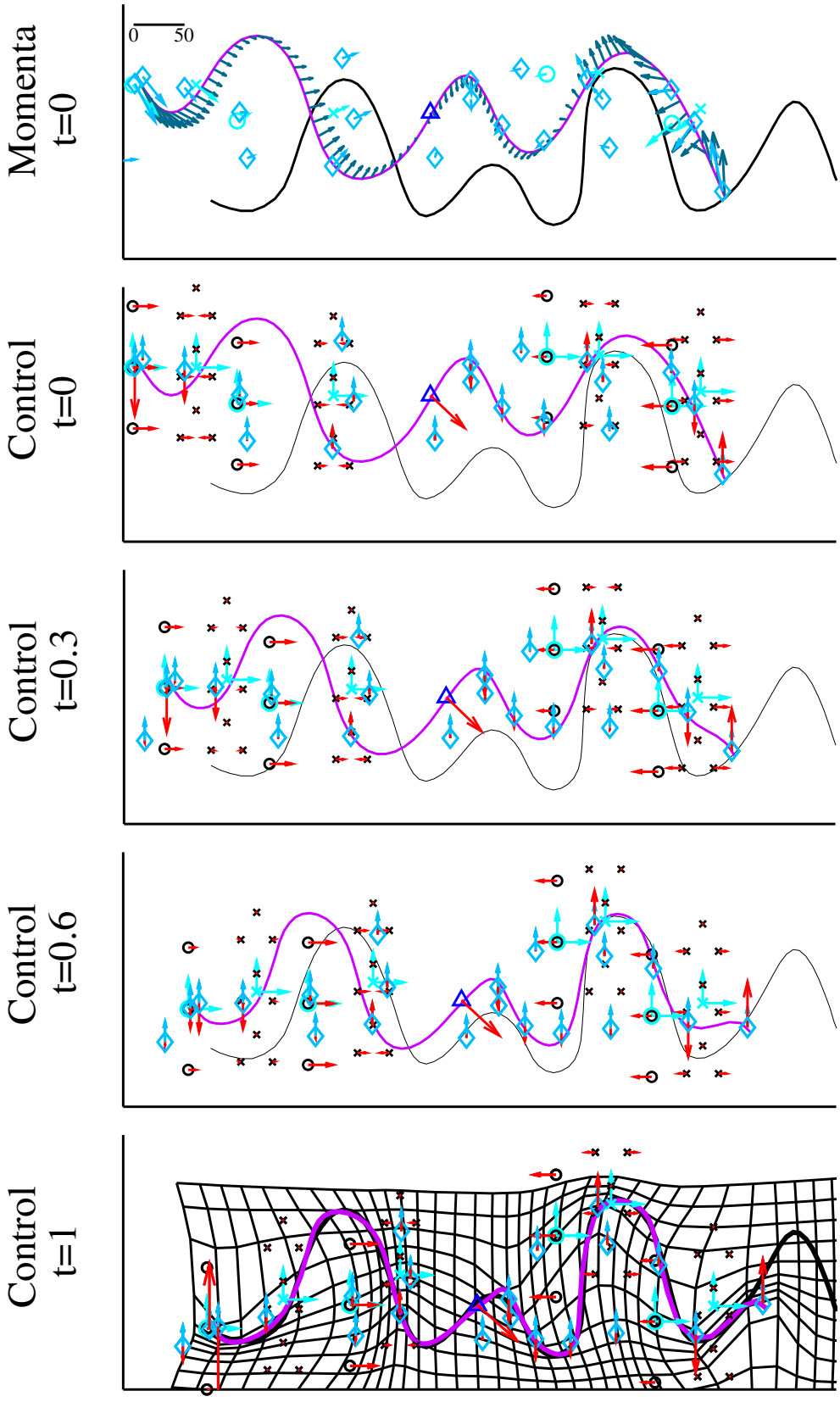


Figure 7.29: **Target 2, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\diamondsuit$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

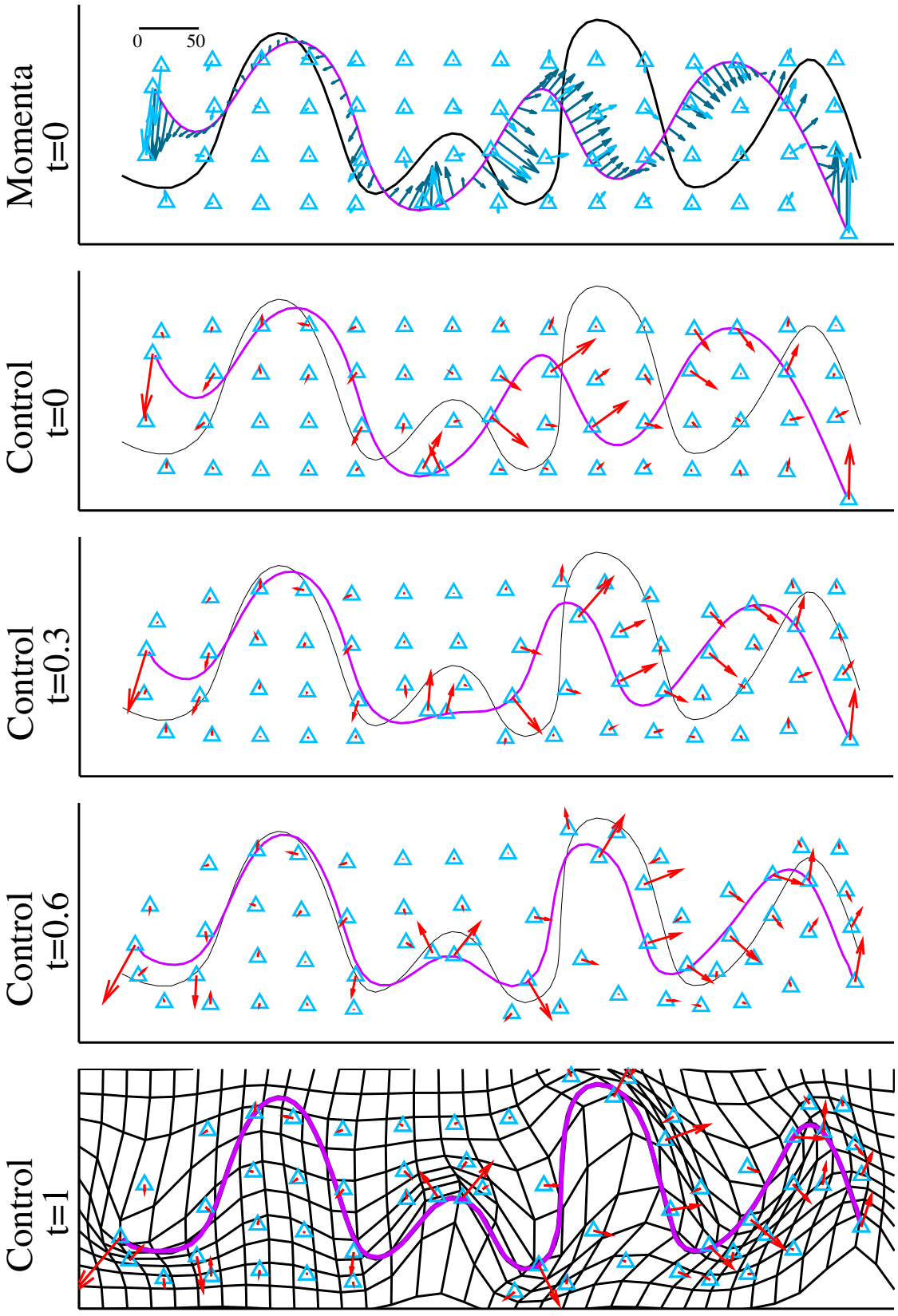


Figure 7.30: **Target 1, multi-scale data attachment term, module M^1.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations). First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

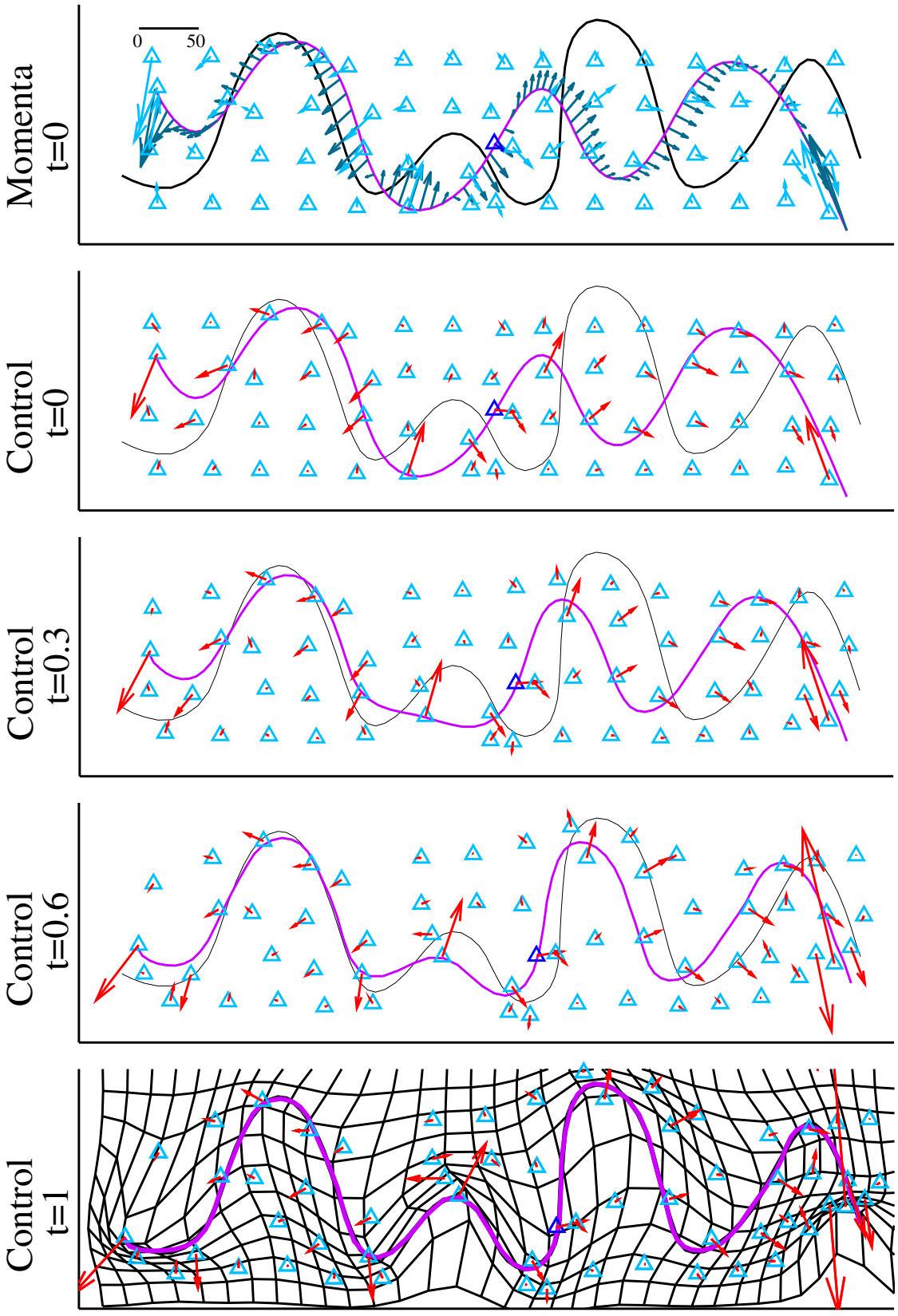


Figure 7.31: **Target 1, multi-scale data attachment term, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

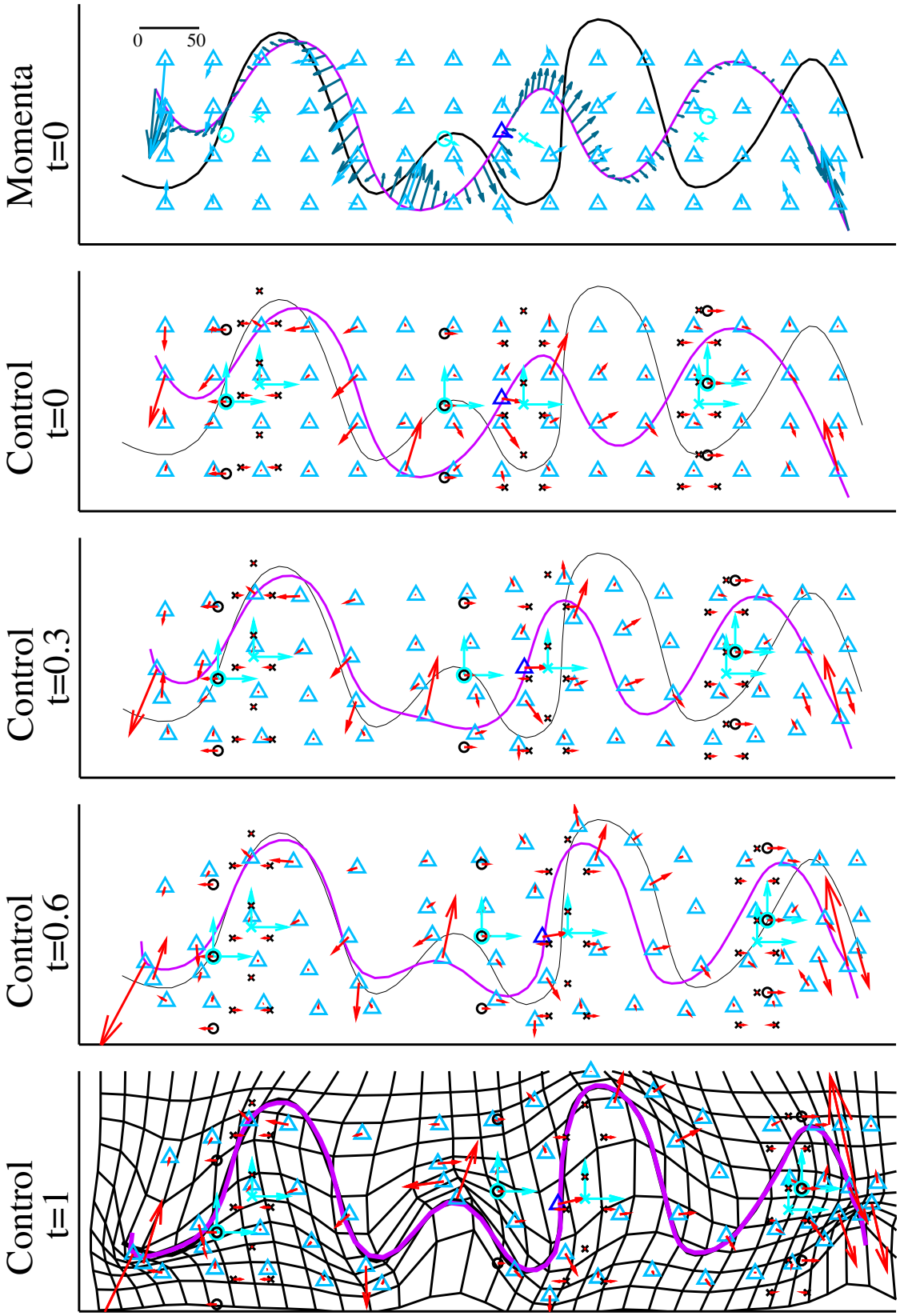


Figure 7.32: **Target 1, multi-scale data attachment term, module M^3.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

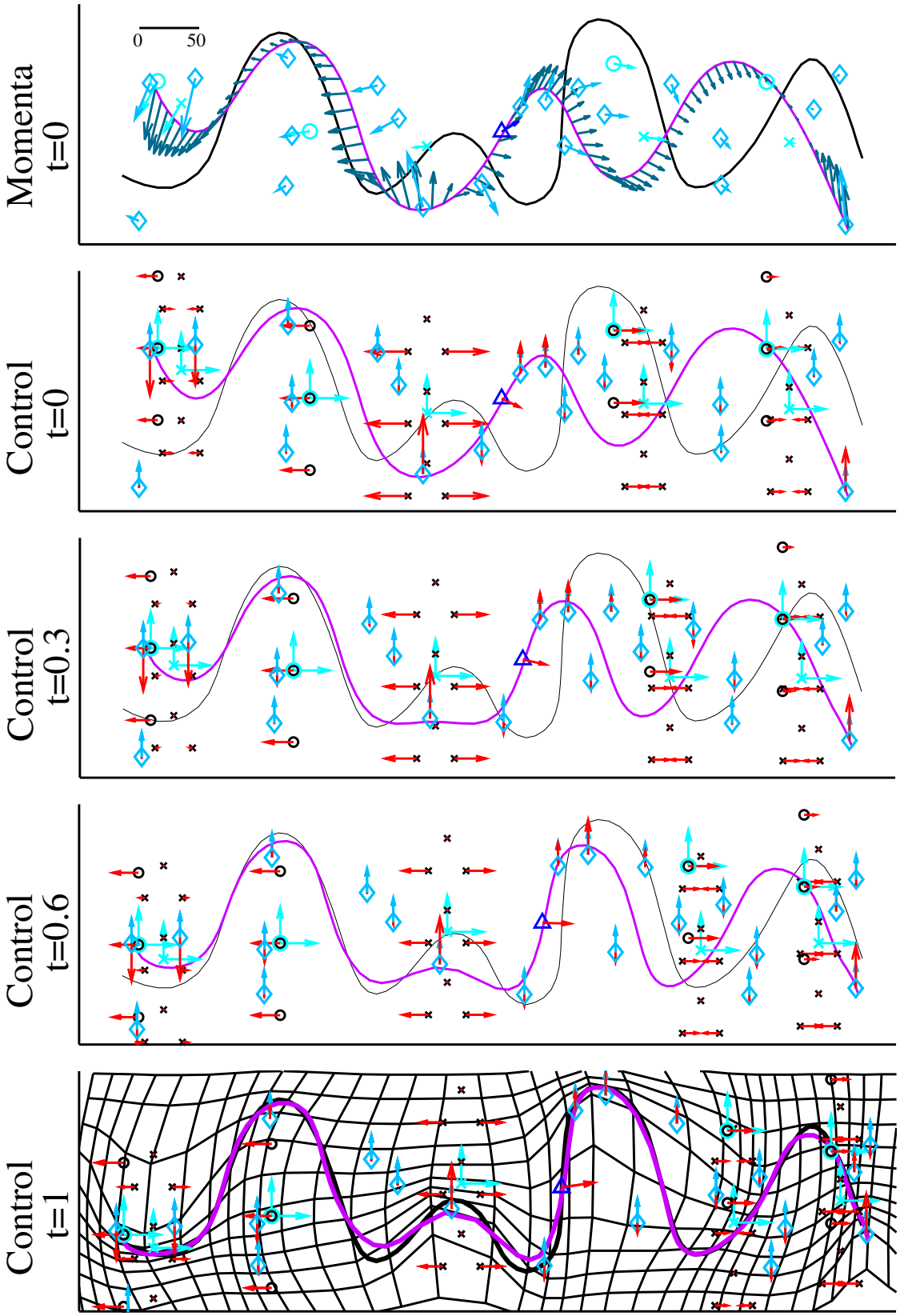


Figure 7.33: **Target 1, multi-scale data attachment term, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\Diamond$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

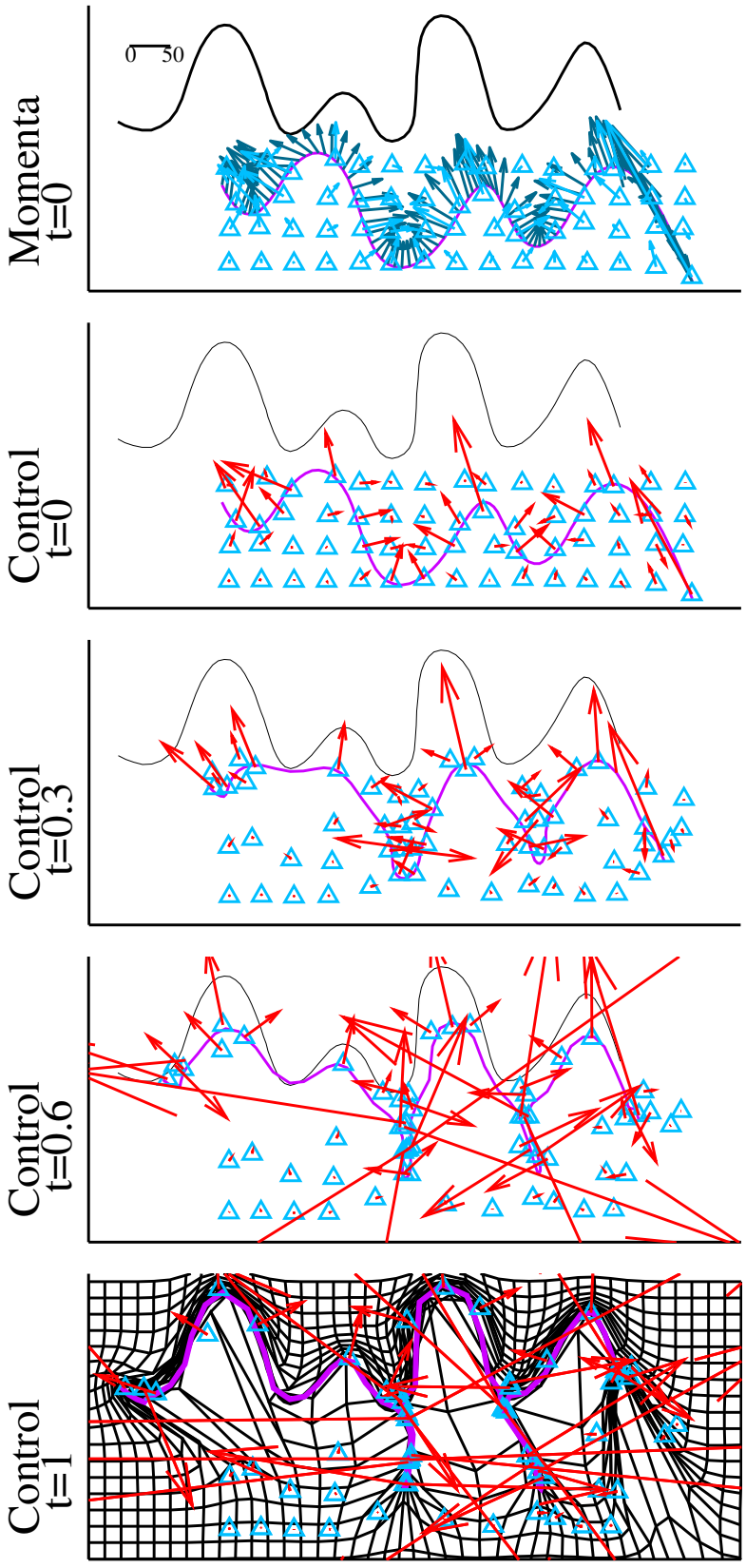


Figure 7.34: **Target 2, multi-scale data attachment term, module M^1.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations). First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

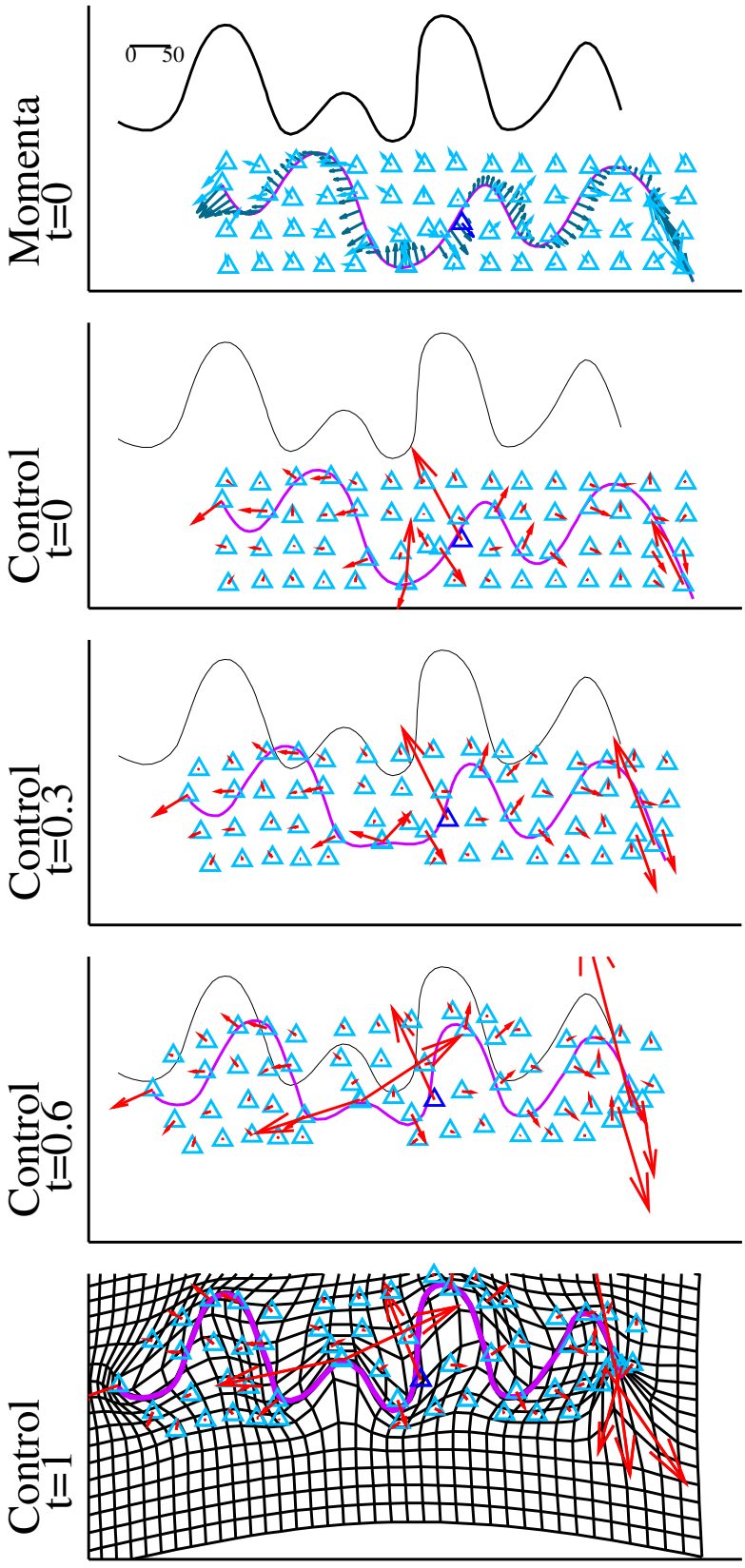


Figure 7.35: **Target 2, multi-scale data attachment term, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

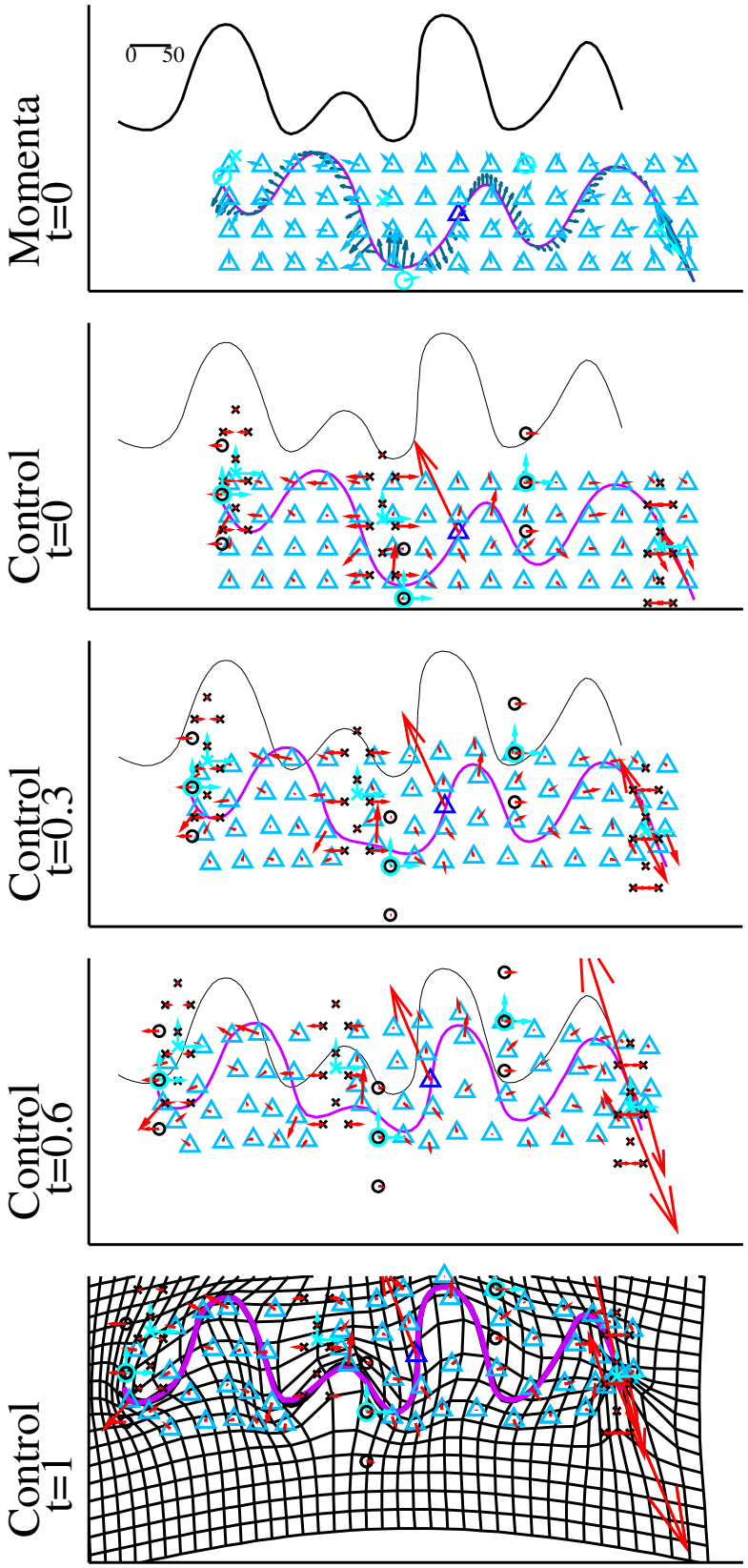


Figure 7.36: **Target 2, multi-scale data attachment term, module M^3.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

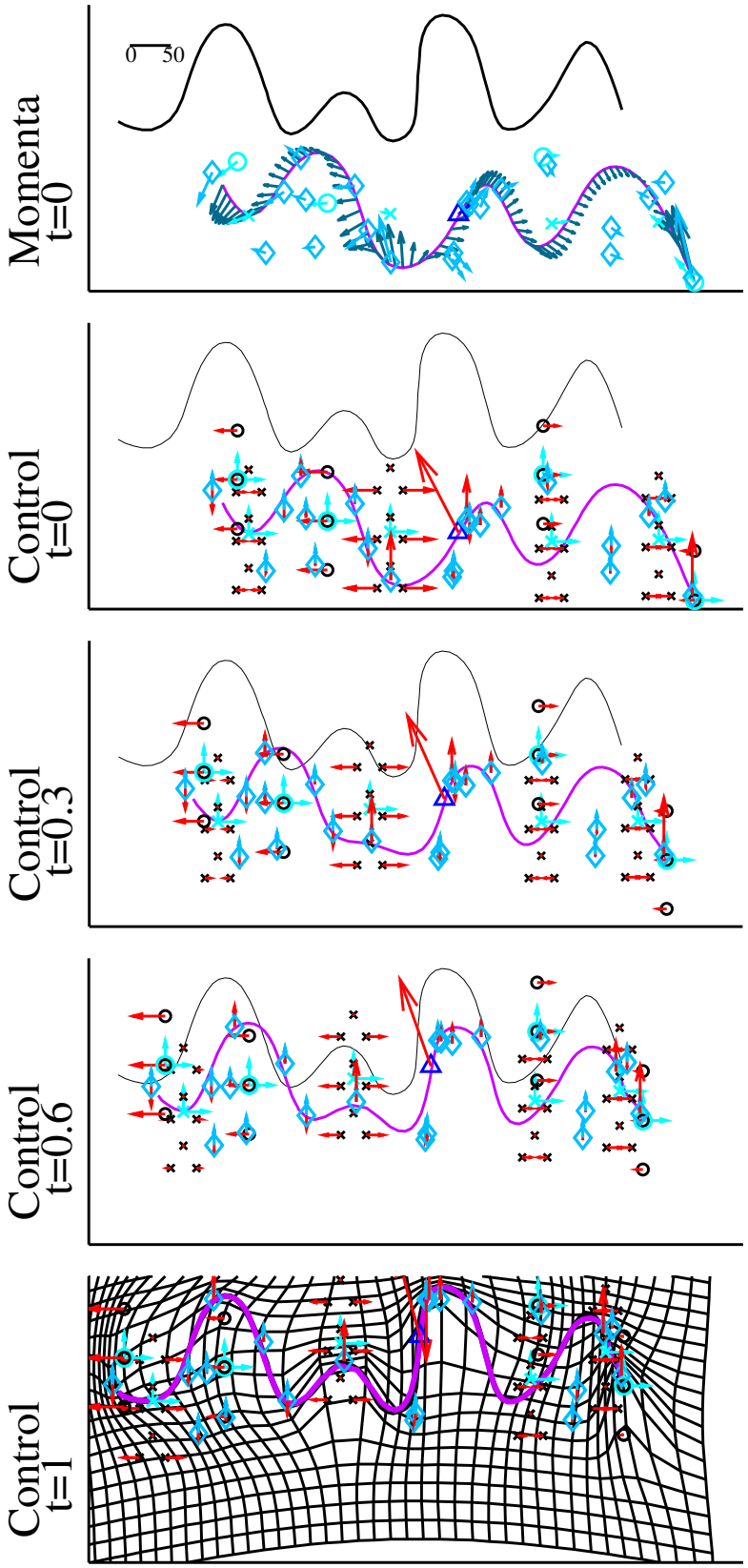


Figure 7.37: **Target 2, multi-scale data attachment term, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\diamondsuit$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

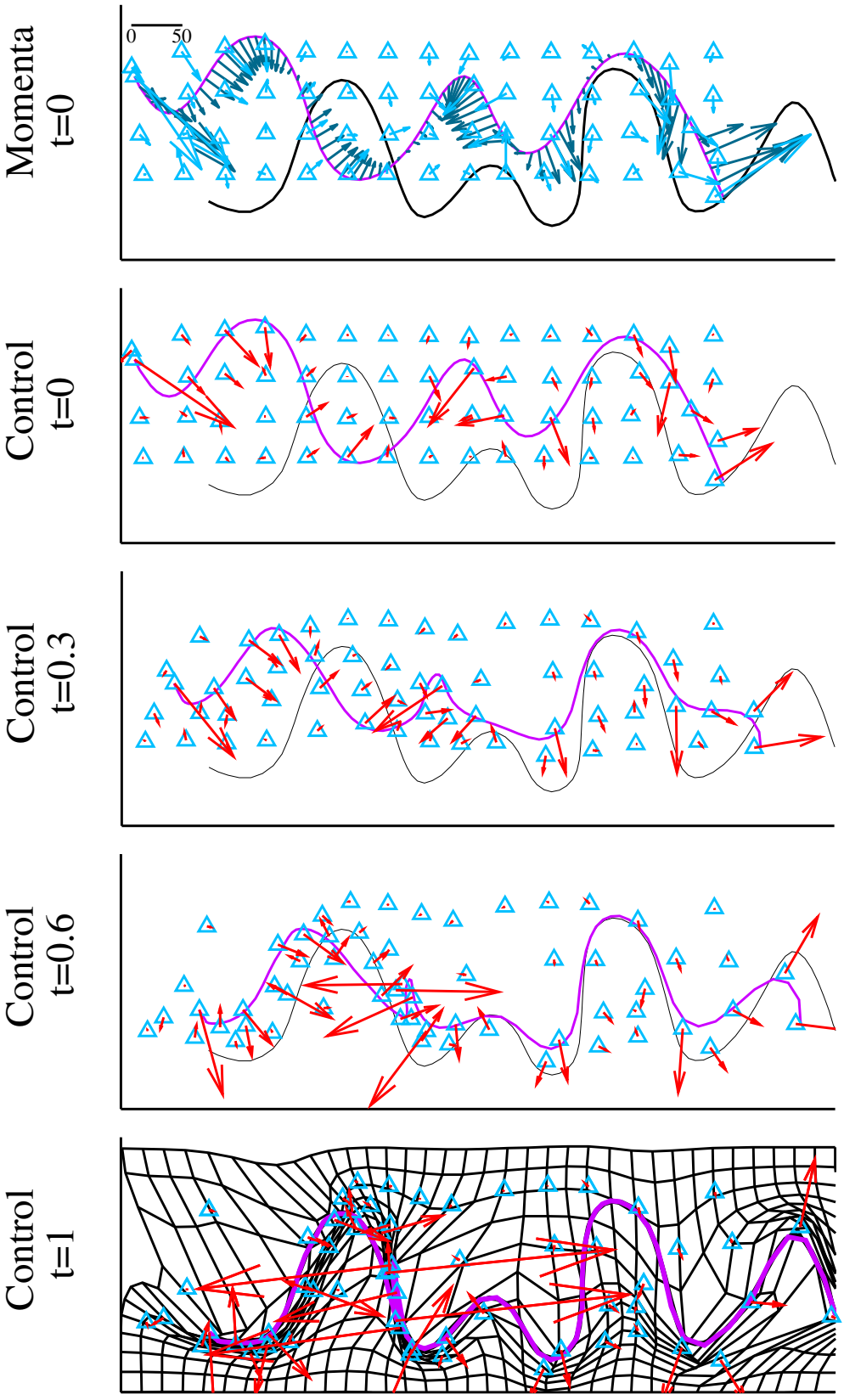


Figure 7.38: **Target 3, multi-scale data attachment term, module M^1.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: △ (centres of translations). First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

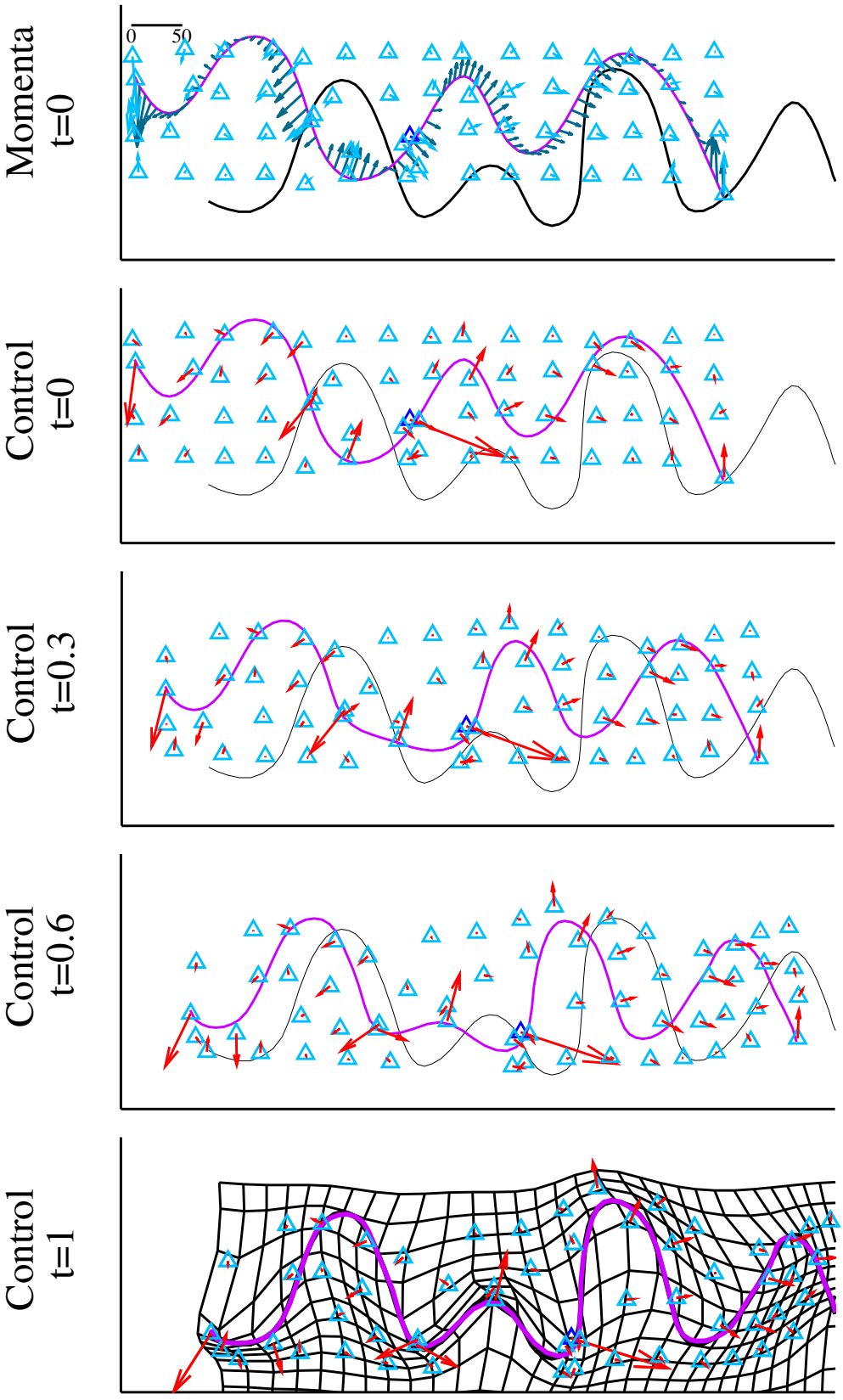


Figure 7.39: **Target 3, multi-scale data attachment term, module M^2.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations), scales: 600 and 40. First line: **Momentum** at $t = 0$. Three lower lines: **Control** at time t: red arrows (vector of translations).

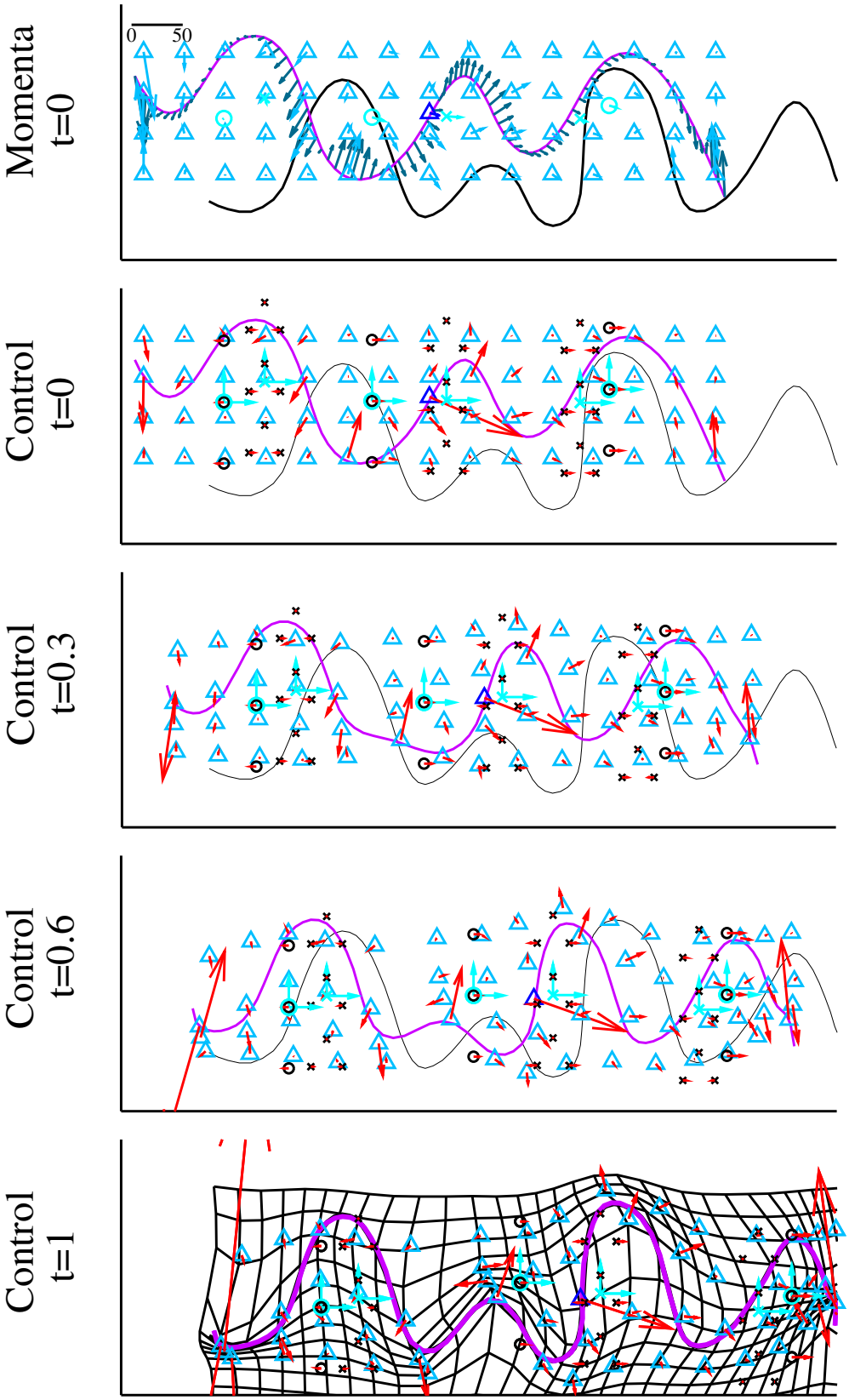


Figure 7.40: **Target 3, multi-scale data attachment term, module M^3.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

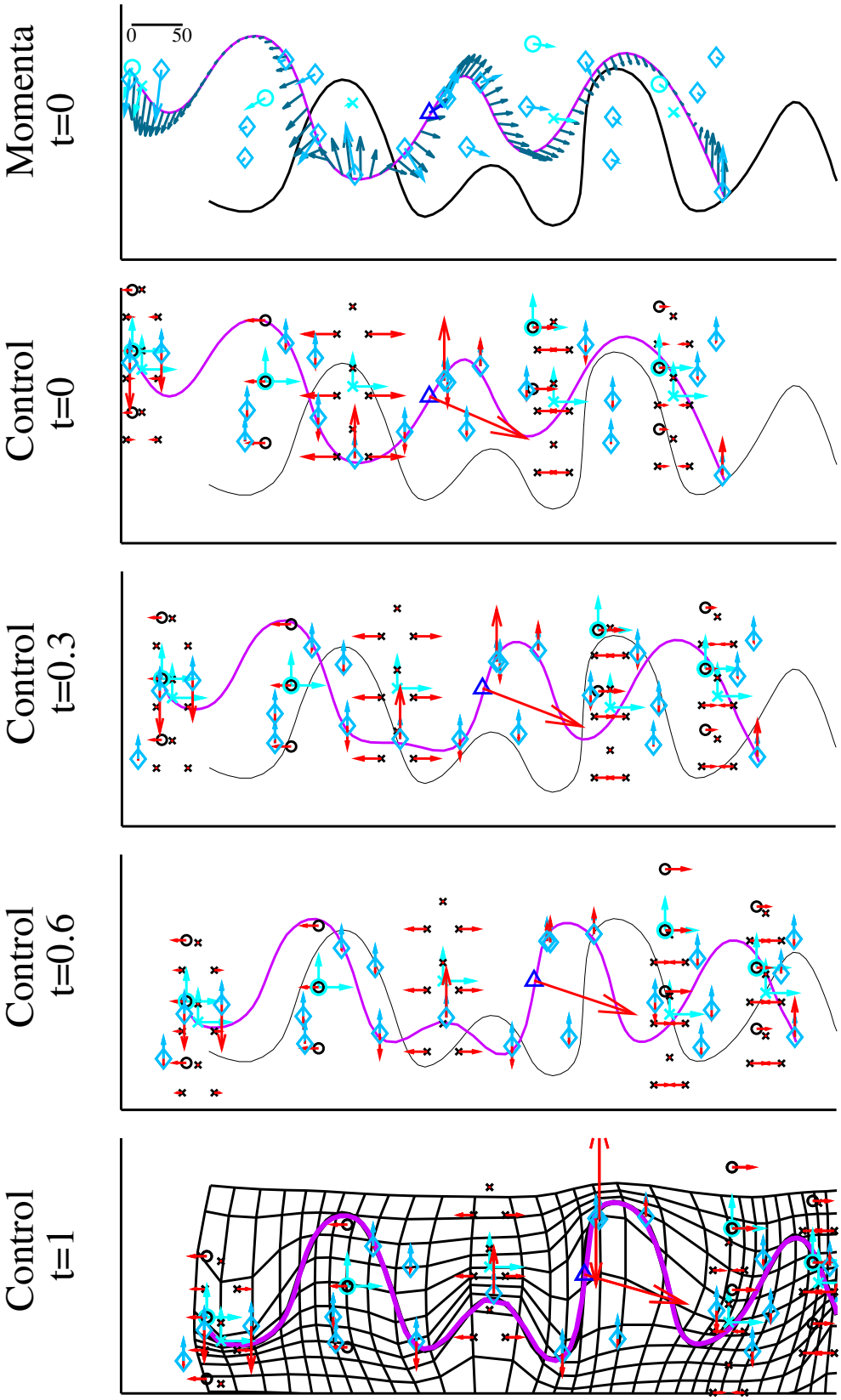


Figure 7.41: **Target 3, multi-scale data attachment term, module M^4.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\Diamond$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

7.2.4.1 Optimized geometrical descriptor

Let us first study the optimization of geometrical descriptor.
For the deformation module M^2, as the distance between centres of translation at scale 40 is equal to 40, they do not move much during the optimisation and then their optimized initial position cannot really be interpreted.
For M^3, centres translations at scale 40 are also close to each other and then do not move a lot during the optimization process. Centres of anisotropic spreadings and translations are very similar for first target shape and the third one (which are close to each other with respect to the second target). The only centre optimized similarly for the second target is a centre of anisotropic translation, around the creation of the small hump. For all other centres of anisotropic transforms, the optimisation is different. As pointed out previously, this is probably due to the fact that most of the global deformations are generated thanks to the isotropic translations at scale 40.
For the last deformation module, the optimized positions of centres of anisotropic spreadings and translations are similar for the three types of targets: with this constrained deformation module they are always needed in the same locations to enable the matching of the source shape onto the target one, regardless the rigid translation. Similarly, optimized positions of centres of isotropic translations are similar for the three experiments. This is probably due to two factors: first there are here less translations at scale 40 here than for M^2 and M^3 and then their centres need to move during the optimisation process. Besides as here translations are constrained to be vertical, the deformation they can generate is less rich. Then for different positions of their centres, it is not possible to generate similar transforms while it is in general possible for unconstrained translations: by constraining the deformation model, we increase the stability of the optimisation with respect to rigid displacements.

These experiments show that if the constraints are adapted, the optimization of initial geometrical descriptors can be interpreted since it does not depend on rigid displacements. However, when deformations are not constrained enough, we loose this interpretability. On the opposite, if the deformation is too constrained, a particular geometrical descriptor could be needed at different locations, and then its convergence would depend on initial conditions (and then would not be meaningful).

7.2.4.2 Trajectories of controls

For deformation module M^4, as horizontal and vertical displacements are dissociated, controls can be naturally interpreted. Besides trajectories of controls for all deformations except the translation at scale 600 are very similar for all experiments, which confirm that they are meaningful. Then for this deformation module, studying the control corresponding to a particular type of displacement at a particular location, allows to study how this displacement occurs. This is not the case for M^3. As we explained previously, the sum of unconstrained translations at scale 40 enables to generate a very rich set of deformations, and then parameters of anisotropic transforms cannot be as well interpreted. We see on figures 7.32, 7.36 and 7.40 that when their controls is non

zero, unconstrained translations around them often generate a similar transforms. Then here their control cannot really be interpreted as the strength with which this type of deformation is needed here. This is a general remark which is due to the non injectivity of the field generator. The same phenomenon occurs with deformation module M^2: translations at scale 40 also carry slightly the rigid translation. This can be seen by comparing controls of translations located in similar area for the registrations of the three targets: their controls are "biaised" in the direction of the translation at scale 600.

As a conclusion one should note that controls cannot be directly interpreted. However when the deformation model is constrained enough, this interpretation can be done without reserve. In general it is not the case, and controls can be used to interpret the type of deformations that are necessary, but not really to interpret the strength with which they are needed.

7.2.4.3 Comparison of initial momenta

One thing to be noted is that for deformation module M^2, silent initial momenta carried by the source shape for the three different targets are similar. This is also the case for deformation module M^4 but for deformation module M^3, the silent initial momenta are similar only between the centred target shape and the third one: only when initial geometrical descriptors are optimized similarly. This is at first sight surprising because the deformations needed are really different and a priori, the silent momenta should carry different *information*. However, the only difference is the rigid registration, which is encoded by the control of the translation at scale 600. Let us we denote $\tilde{M} = (\tilde{\mathcal{O}}, \tilde{H}, \tilde{\zeta}, \tilde{\xi}, \tilde{c})$ the deformation module generating a local translation at scale 600, ξ the infinitesimal action on our data curves and (e_1, e_2) an orthonormal basis of $\mathbb{R}^2$. From Remark 23 we obtain that for $u \in \{1, 2\}$, the gradient at f of the $u-$th component of the geodesic control $\tilde{h}$ of $\tilde{M}$ with respect to the silent momentum is $\xi_f \circ \tilde{\zeta}(e_u)$. Besides, there is no reason for this control to vary during the integration of the geodesic trajectory: as it affects all points in the same manner, the trajectories of least cost are the constant ones. Then the gradient of the $u-$th component of the rigid translation is roughly $\xi_f \circ \tilde{\zeta}(e_u)$ and since $\tilde{\zeta}(e_u)$ is roughly a rigid transform, the gradient will be the same for all points of the curves. As a consequence, all vectors of the initial silent momentum will contribute similarly to the rigid registration. Then the influence of the needed rigid translation on the initial momentum is distributed similarly on all points of the curve and it is negligible compared with influences of more local deformation patterns (which will not be spread on all momenta but only on a few of them). This explains why they are so similar to each other. As an illustration, we subtract this "rigid translation contribution" from initial momenta obtained thanks to the deformation module M^4 for the matching with the second target, and we compute the new geodesic trajectory corresponding to this new momentum: it is displayed in Figure 7.42. We see that the deformation is close to the one we obtained previously, except that the rigid translation is almost null.

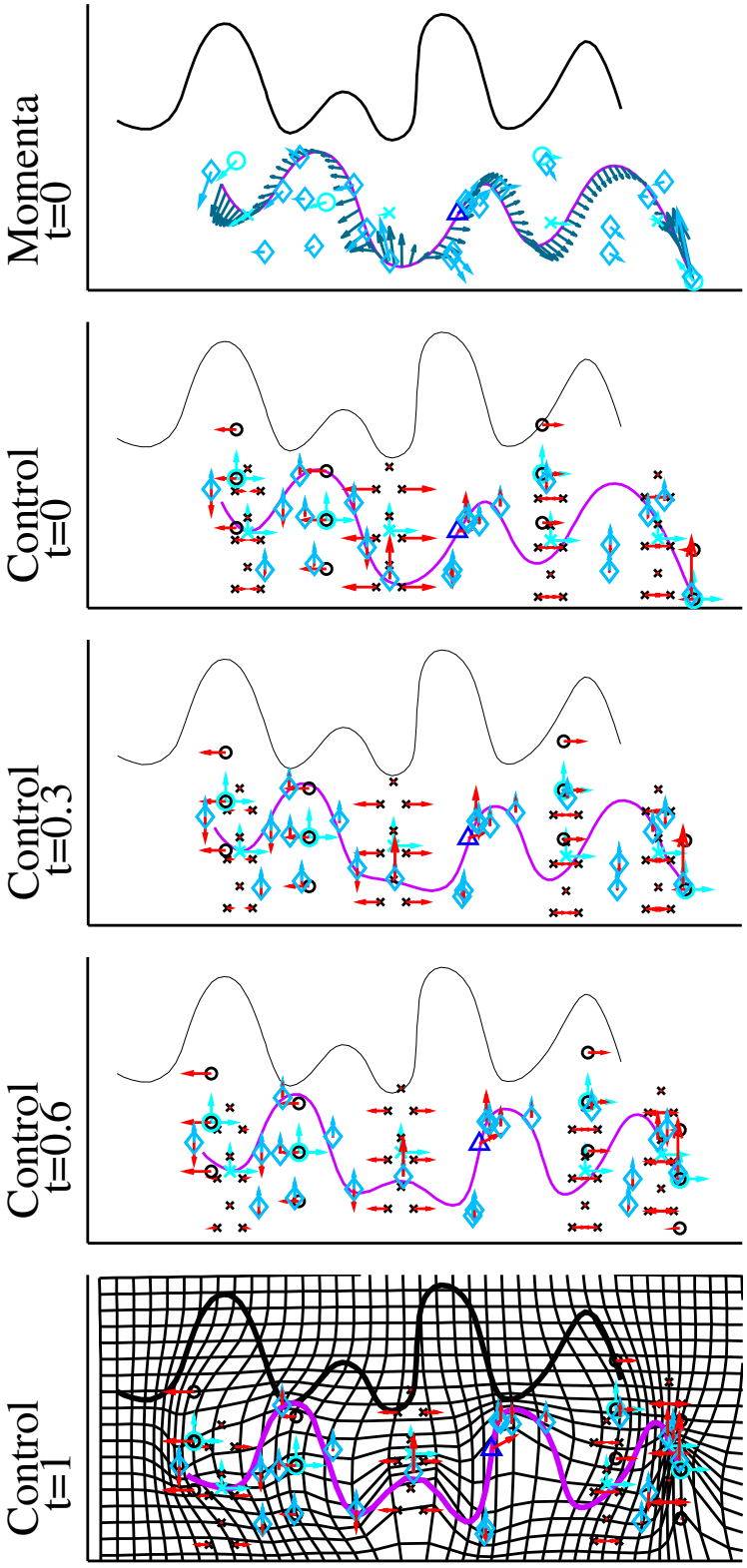


Figure 7.42: **Target 2, multi-scale data attachment term, module M^4, shooting with rigid part removed.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scale 600), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60), blue vectors attached to $\Diamond$ (isotropic translations with prior on direction, scale 40). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

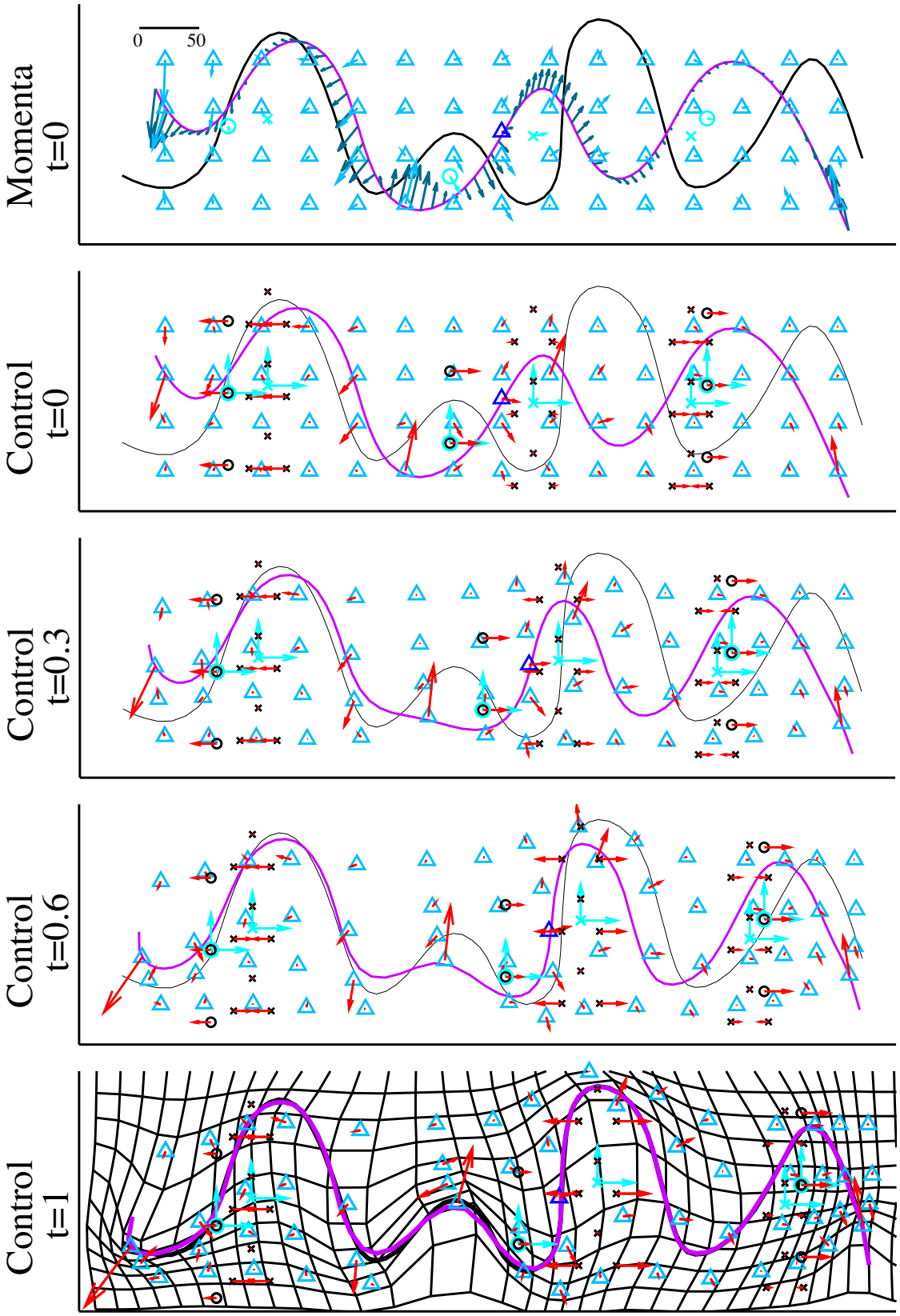


Figure 7.43: **Target 1, multi-scale data attachment term, module $\tilde{M}^3$.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

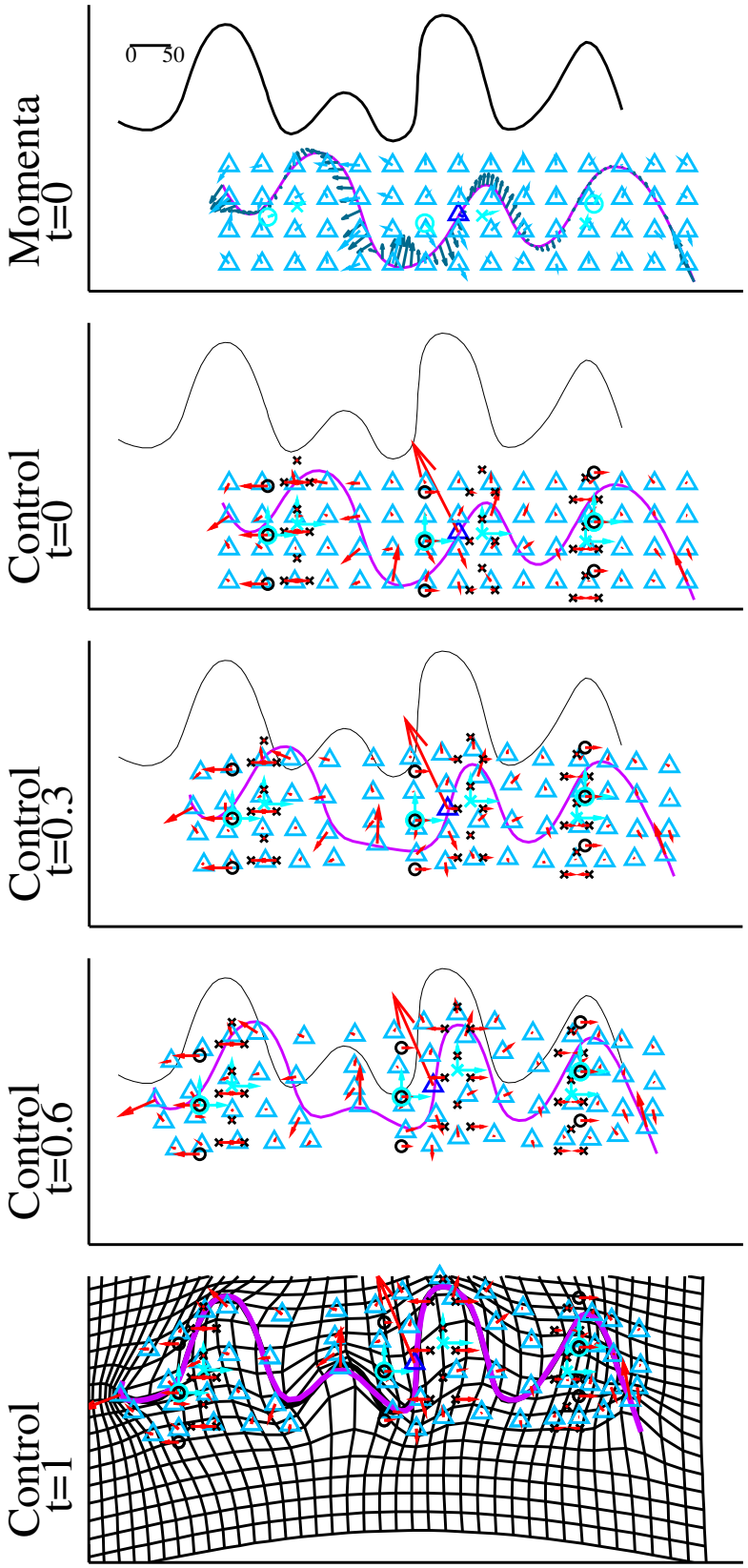


Figure 7.44: **Target 2, multi-scale data attachment term, module $\tilde{M}^3$.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

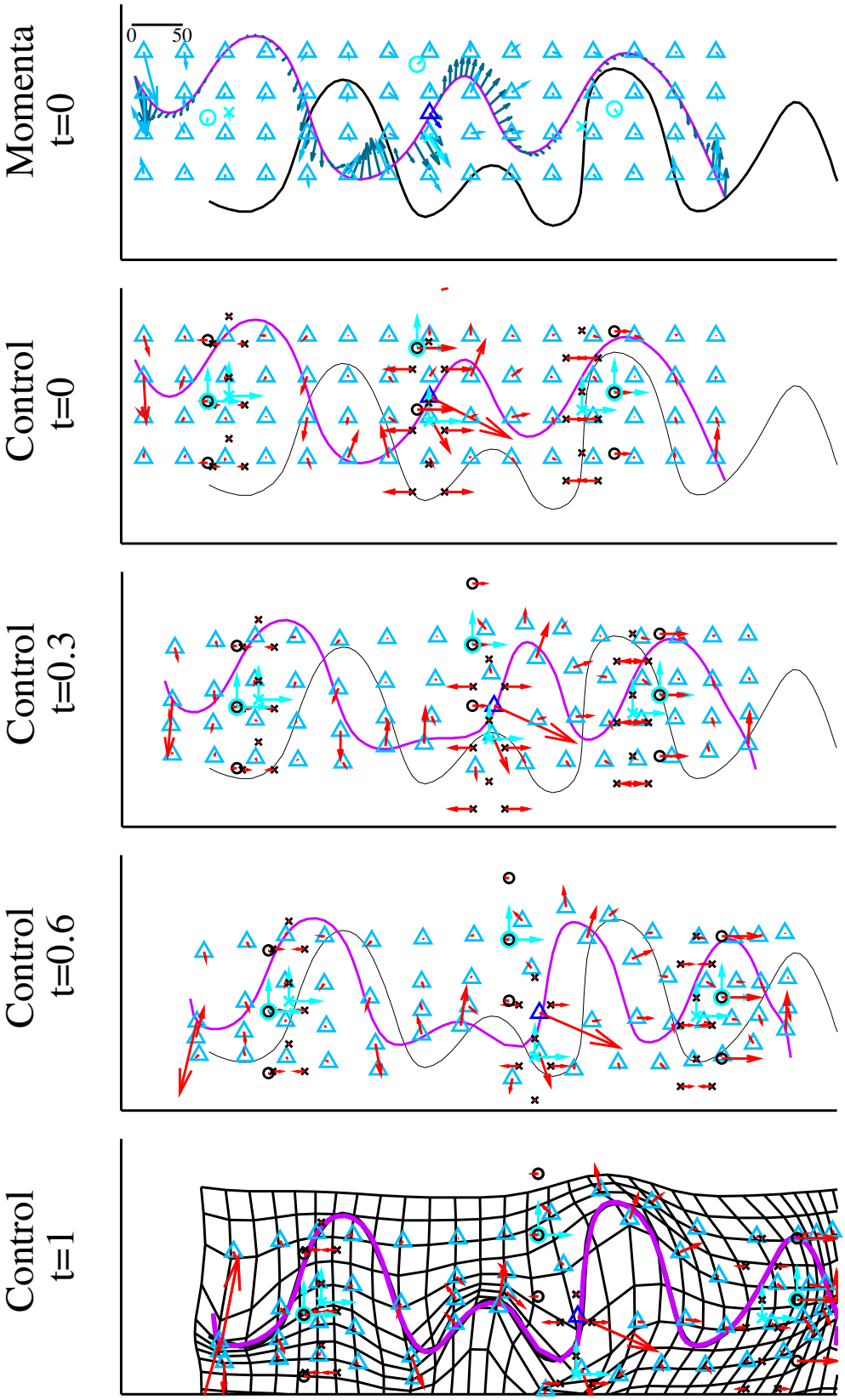


Figure 7.45: **Target 3, multi-scale data attachment term, module $\tilde{M}^3$.** Source at time t (in purple) and target (in black). **Geometrical descriptor** at time t: $\triangle$ (centres of translations, scales 600 and 40), blue vectors attached to $\times$ (anisotropic spreading, scale 60), blue vectors attached to $\circ$ (anisotropic translations with prior on direction, scale 60). First line: **Momentum** at $t = 0$. Three lower lines: Intermediate tools in black ($\times$ for spreading, $\circ$ for translations with prior on direction) and **Control** at time t in red.

7.2.5 Favouring some deformation patterns by changing the metric

We saw that when using the deformation module M^3, the anisotropic spreadings and translations had controls almost equal to zero along the trajectories and that the optimisation of their initial geometrical descriptors were not similar for the three positions of the target shape. We concluded that these two phenomena had a common cause: the non linear deformation patterns are generated by the local translations at scales 40. In this section we study the idea, presented in Section 5.2.3.3, that it is possible to favour the use of the deformation generated by one of the combined deformation modules by changing the combination rule for the costs.
The deformation module M^3 is built as the combination of four deformation modules and its cost is given by the sum of the costs of these four deformation modules. We build now a different module $\tilde{M}^3$ which is similar to $M^3 = (\mathcal{O}^3, H^3, \zeta^3, \xi^3, c^3)$ except that in the compound cost, the contributions of the module generating the anisotropic spreading, and the one generating the anisotropic translations are divided by 3: we can write $\tilde{M}^3 = (\mathcal{O}^3, H^3, \zeta^3, \xi^3, \tilde{c}^3)$. Then with this deformation module $\tilde{M}^3$, using anisotropic spreadings and translations "costs" three times less than with M^3. As a consequence, geodesic trajectories are not the same with $\tilde{M}^3$ as with M^3, and in particular to a given momentum corresponds now an other geodesic control as presented in Section 5.2.3.3. We present in Figures 7.43, 7.44 and 7.45 the results of the registrations with $\tilde{M}^3$ for the three target shapes. We can see that controls associated to the anisotropic spreadings and translations are now clearly non null. Besides the optimised positions of their geometrical descriptors are now similar for the three positions of the target shapes. This experiment shows first that the fact that the optimisations of their geometrical descriptors were different for the different target shapes was due to the fact that these deformations were not really *used* to generate the non linear patterns. It also confirms that we are able to favour one of the combined deformation modules by reducing its costs: the non-weighted sum that we have been defining for compound costs is not the only possibility and might not be the most adapted in some cases.

7.3 Performing jointly rigid and non linear registration to study variability among the population

In this section we present the construction of an atlas from a collection of 40 rabbit profiles (see Figure 7.46 for examples) with variable ear size and variable positioning of the shape in the ambient space. These shapes are encoded as varifolds (see 3.3.2, [CT13]) so that no point-correspondence is assumed. We propose here to combine rigid registration and local deformation in a single optimisation, whereas in the vase majority of cases shapes are rigidly registered as a pre-processing step before statistical analysis.

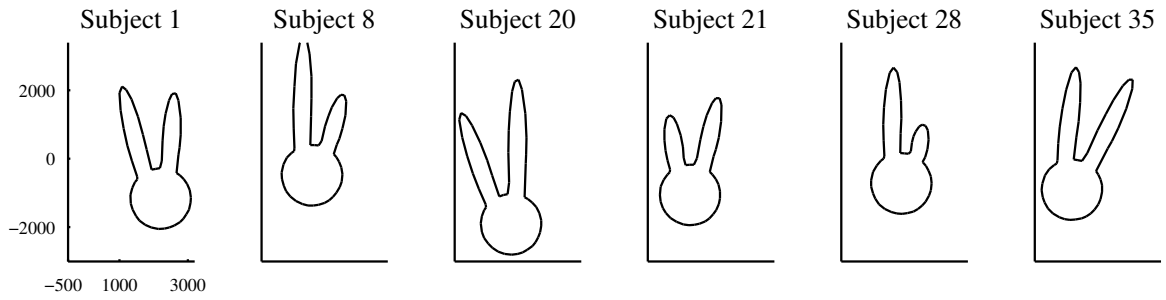


Figure 7.46: **Performing jointly rigid and non linear registration.** Target shapes

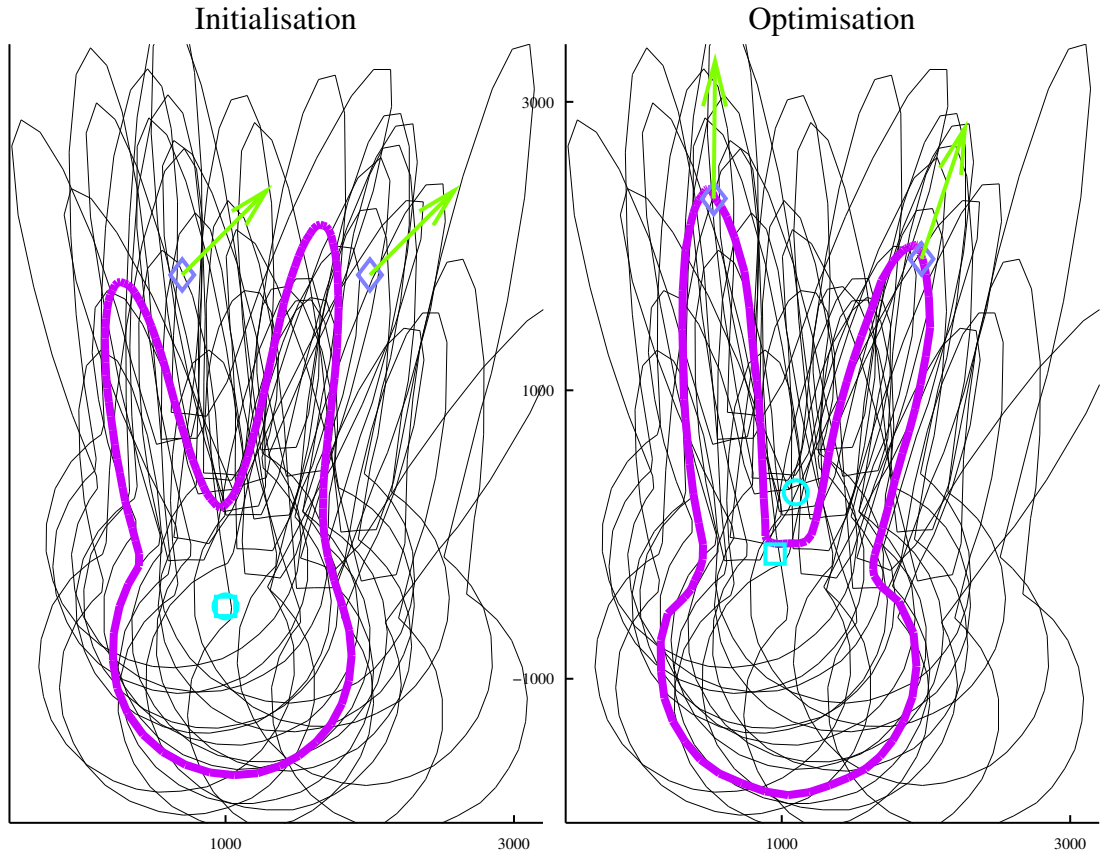


Figure 7.47: **Performing jointly rigid and non linear registration.** SDGS framework. **Template** at time $t = 0$ (purple shape) and other **geometrical descriptors**: rotation's one is the blue circle, the translation's ones is the blue square and translations with transported vectors' ones have their dot-component represented by blue diamonds and their vector components represented by green arrows. **Targets** in black. Left: Before optimisation. Right: After optimisation.

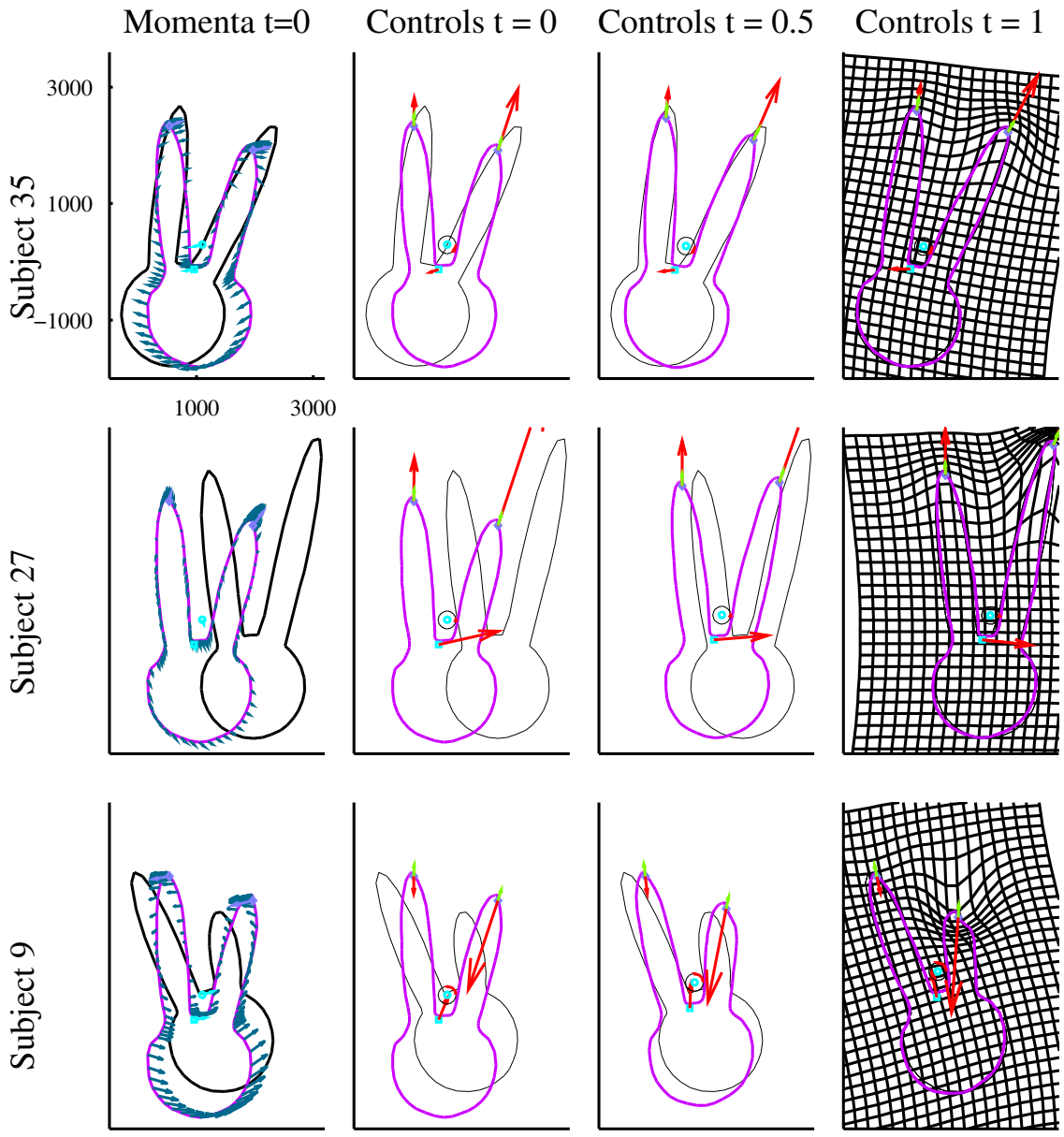


Figure 7.48: **Performing jointly rigid and non linear registration.** SDGS framework. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

7.3.1 Using an adapted deformation module

To this end, we propose to use a deformation module M which is the combination of three deformation modules: the first one generates vector fields that are a rotation at scale 5000 (see Section 4.2.3), the second one generates vector fields that are a translation at scale 5000 (see Section 4.2.2). The last deformation module generates a sum of two translations prior on directions, with directions updated by adjoint action, at scale 600 (See Section 4.2.2.3). The first two modules encode a rigid body transformation (at the scale of the shapes), and the third one encodes local non-linear deformation patterns. In Figure 7.47 are plotted template and other geometrical descriptors at $t = 0$, before and after optimisation. Note that the geometrical descriptor of the deformation module generating a sum of two translations with prior on directions is composed of two points and two vectors : the initial directions of these two vectors are shared by all subjects. The optimised initial directions are along ears : the variability amongst the population at this scale is in the direction of ears, which was expected. Besides, as a consequence, values of associated controls measure the growth or shrinking of ears and then this particular feature can be studied independently from the rigid registration. In Figure 7.48, on the first columns, are presented parametrisation of minimizing trajectories (template,other geometrical descriptors and momenta) for three subjects. Note that momenta associated to the sum of two translations with updated directions have two components: one associated to base points of translations, and one to vectors of translations. All these vectors are represented attached to the base point of the translation, but in different colors. In the three columns on the right hand side of this figure are represented the trajectory of the template, geometrical descriptors and controls for these three subjects. Controls associated to translations with updated direction are scalar so they are represented as lengths of vectors. The geometrical descriptor of the rotation is a blue circle, circled by a black one. The control associated to the rotation (scalar) is represented by the portion of this black circle which is coloured in red (the control is positive if the colouring is anti-clockwise and negative otherwise).

7.3.2 Comparison with the Sparse LDDMM framework [DPGJ11]

We compare our result with the one obtained by rigidly registering data beforehand and then applying the framework developed in [DPGJ11]: in this framework the vector field is built as a sum of local translations carried by control points. We use here 50 control points, and we set the scale of translations equal to 500. In Figure 7.49 we present the targets after the rigid registration, as well as the initial position of the template and control points, before and after optimisation. In Figure 7.50 are presented the parametrisation of optimal trajectories for several subjects (in this framework momenta are equal to controls) and the evolution of the template, control points and controls. One can see that non linear deformations are now necessary not only in the area of the ears, but also in the inferior part of the head. This fact is the direct consequence of the rigid-body registration, which does not align the inferior parts of the heads. It shows the sub-optimality of the greedy approach that consists in optimising the rigid part and the

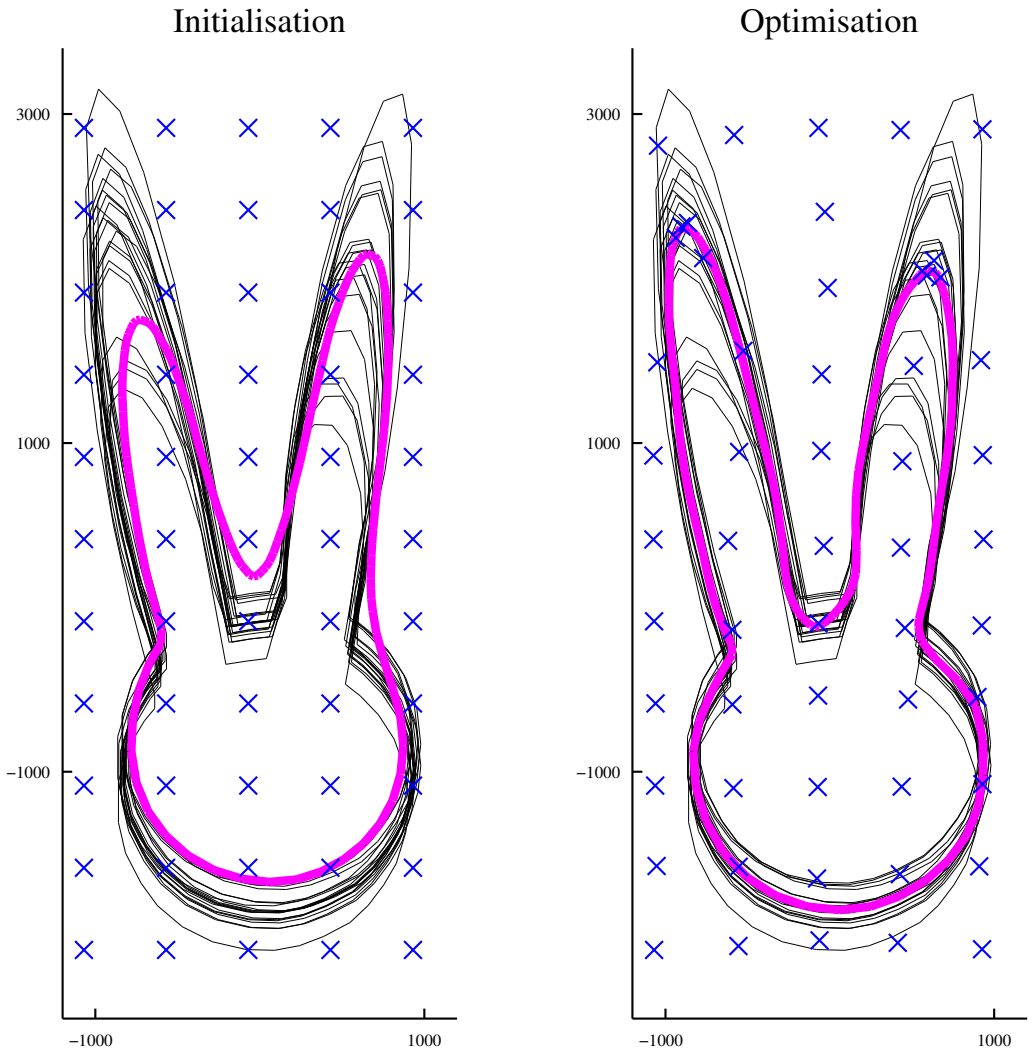


Figure 7.49: **Performing rigid before non linear registration.** Framework [DPGJ11]. **Template** at time $t = 0$ (purple) and **control points** (blue crosses). **Targets** in black. Left: Before optimisation. Right: After optimisation.

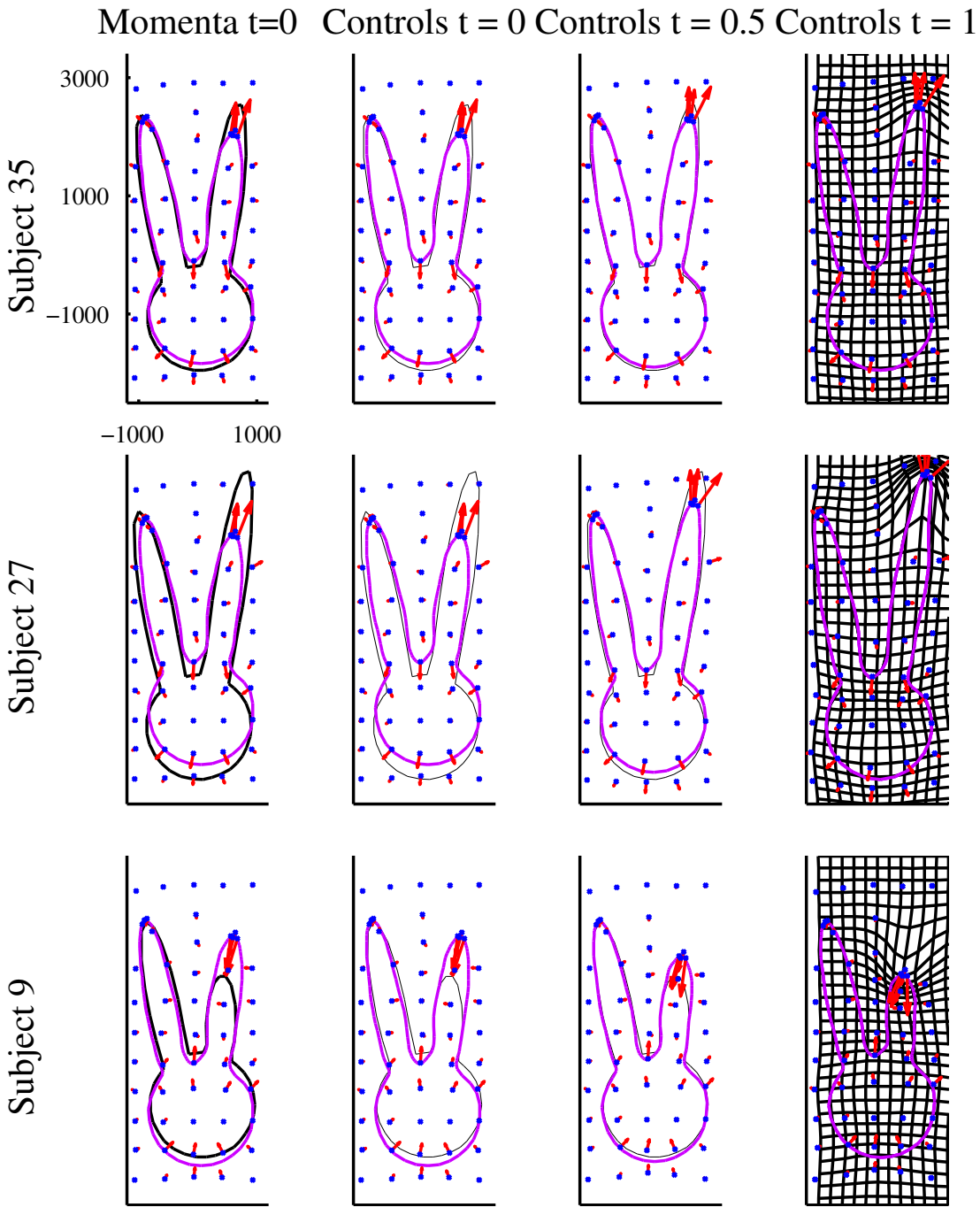


Figure 7.50: **Performing rigid before non linear registration.** Framework [DPGJ11]. **Template** at time t (in purple), **target shapes** are in black. **Control points** at time t (blue crosses). Left column: **Momenta** at $t = 0$ (blue arrows). Three right-hand columns: **Controls** at time t in red.

non-linear part in two consecutive step. Very few mathematical frameworks may deal with this issue, whereas it is well-known in the statistical shape analysis community. By contrast, the method that we present here allows the joint optimisation of linear and non-linear deformation patterns by the use of combined deformation modules at various scales. Besides we show the result obtained by rigidly registering data beforehand and then building the atlas of these shapes with the same deformation module, see Figures 7.51 and 7.52. One can see that the misalignment is corrected thanks to the large translation and the large rotation, and that the deformations of the ears are similar with and without initial rigid registration.

The resulting explanation of the variability seen in the data is much more satisfying by displaying a fixed head and ears of variable sizes.

7.3.3 With the SIGS framework

7.3.3.1 Presentation

In the SDGS framework presented in Section 7.3.1, we combine deformation module M with the silent deformation module induced by the data shape space. Then optimal deformations transporting the mean shape to each subject are parametrized by an initial momenta of this compound deformation module and its dimension is higher than the dimension of data shapes. As explained in Section 5.4.1, it is possible to restrict ourselves to deformations parametrized in smaller dimension: we consider optimal trajectories that can be generated by M and then search, amongst them, the ones that enable to transport the mean shape as close as possible to data shapes. In this framework we estimate one mean shape, one initial geometrical descriptor (of M, so only centers of translations, rotation and directions of translations with prior on directions) and 40 initial momenta. Here momenta are dual variable for deformation module M and then are of dimension of its geometrical descriptor: 12 (2 for the translation, 2 for the rotation and 4 for each translation with prior on direction). Initial position for the mean shape and geometrical descriptor are presented in Figure 7.53 before (left) and after (right) optimization. We also present in Figure 7.54 the parametrization of minimizing trajectories (template, initial geometrical descriptors and momenta) and trajectories of template, geometrical descriptors and controls. It can be seen that in this SIGS framework there is no momentum attached to the template.

7.3.3.2 Comparison with the SDGS framework

In Figure 7.55 we plot the decreasing of the data-attachment term for both SDGS and SIGS framework. One can see that at the beginning of the optimisation process, the SDGS framework seems more efficient. Nevertheless this is no longer true after around 300 iterations. This is confirmed by the comparison of Figure 7.47 with 7.53 and 7.48 with 7.53. One can also see that initial geometrical descriptors and trajectories of controls are close between the two frameworks (even though not equal). Initial momenta on the contrary are necessary different since in the SDGS framework there is an additional

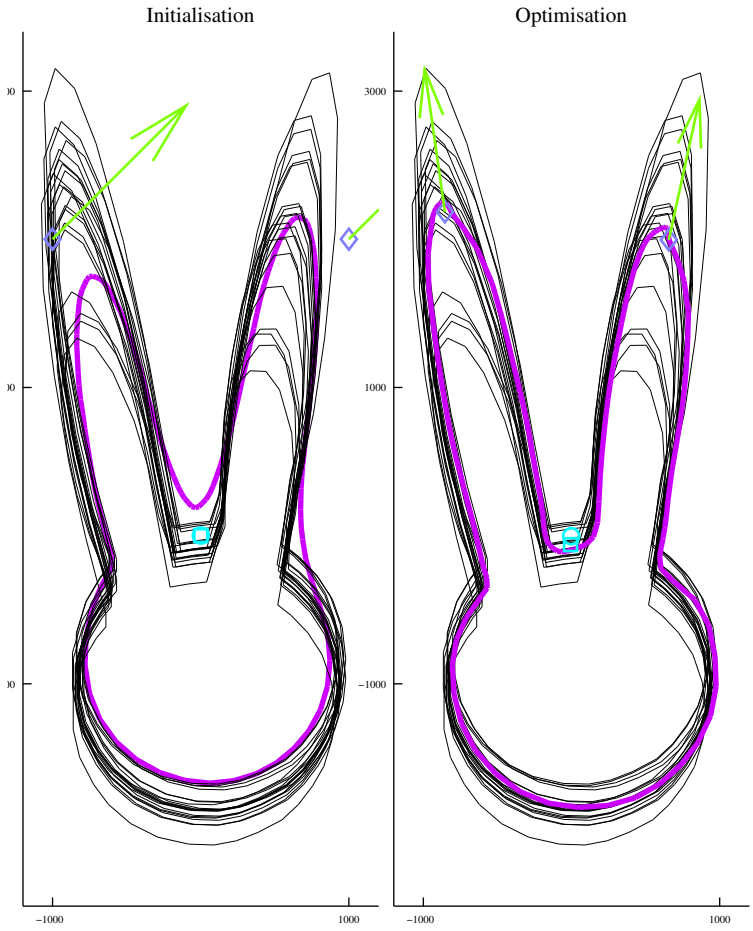


Figure 7.51: **Performing jointly rigid and non linear registration.** Shapes with previous rigid registration. **Template** at time $t = 0$ (purple shape) and other **geometrical descriptors**: rotation's one is the blue circle, the translation's ones is the blue square and translations with transported vectors' ones have their dot-component represented by blue diamonds and their vector components represented by green arrows. **Targets** in black. Left: Before optimisation. Right: After optimisation.

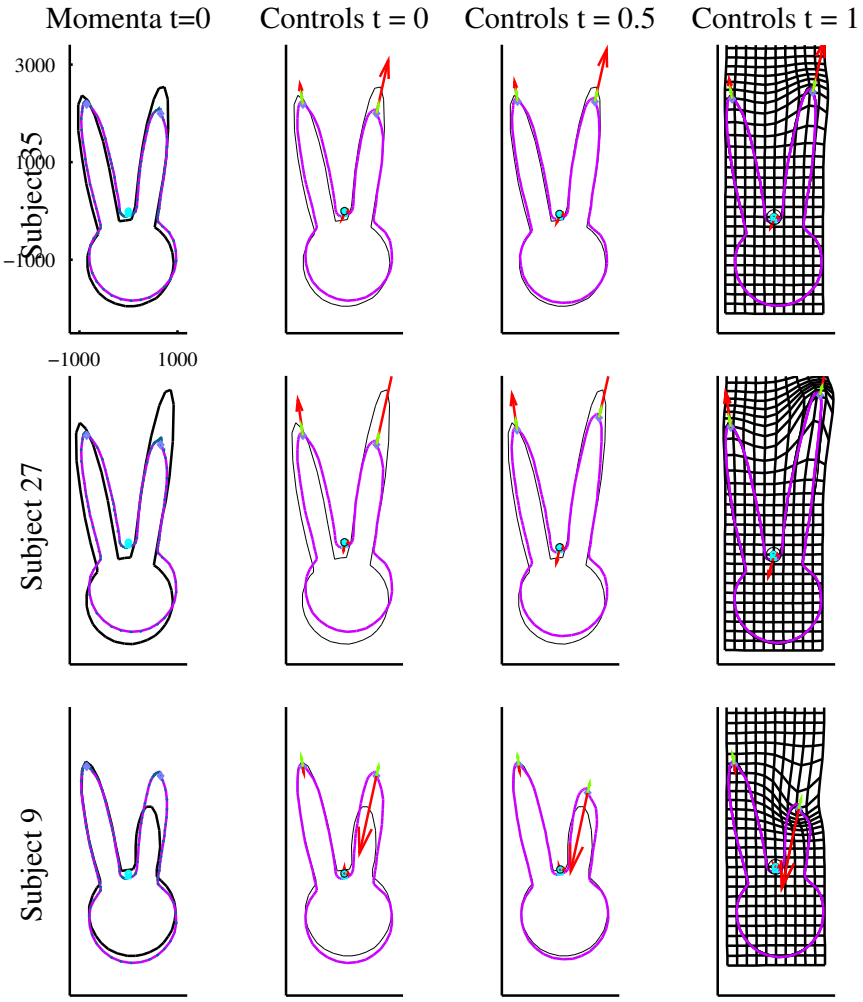


Figure 7.52: **Performing jointly rigid and non linear registration.** Shapes with previous rigid registration. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

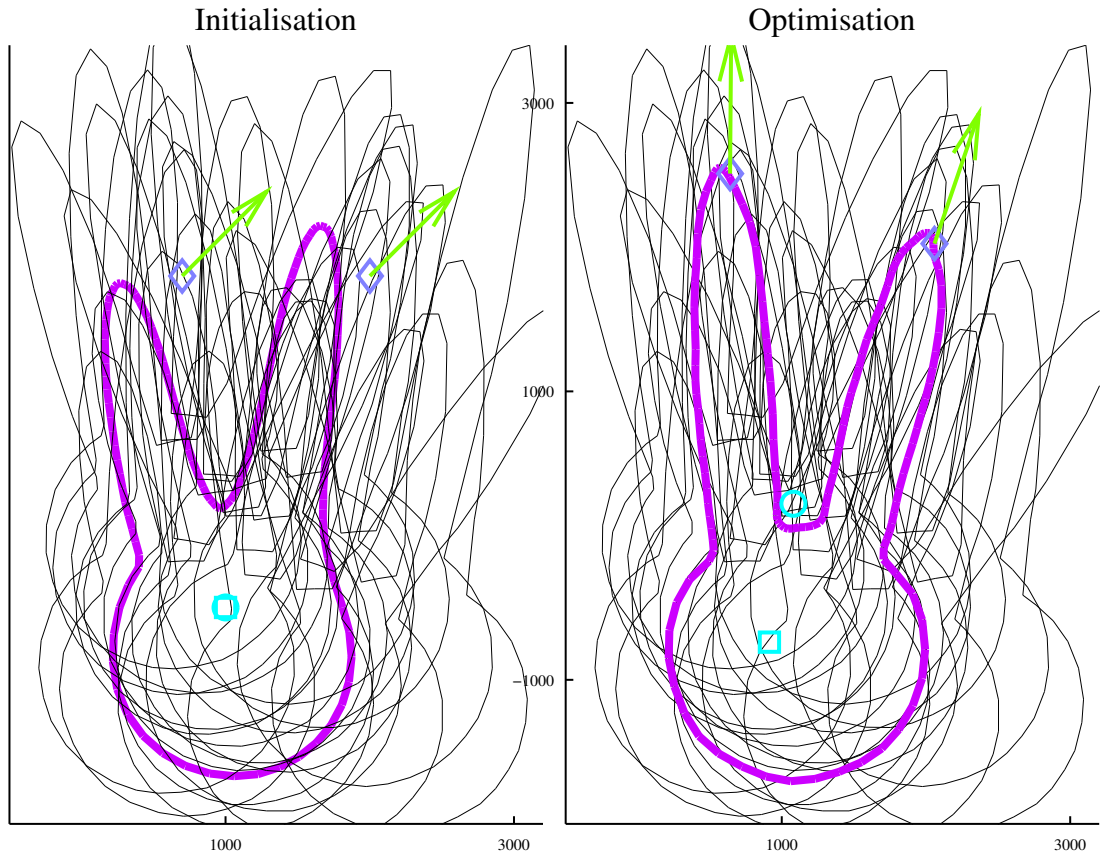


Figure 7.53: **Performing jointly rigid and non linear registration.** SIGS framework. **Template** at time $t = 0$ (purple shape) and other **geometrical descriptors**: rotation's one is the blue circle, the translation's ones is the blue square and translations with transported vectors' ones have their dot-component represented by blue diamonds and their vector components represented by green arrows. **Targets** in black. Left: Before optimisation. Right: After optimisation.

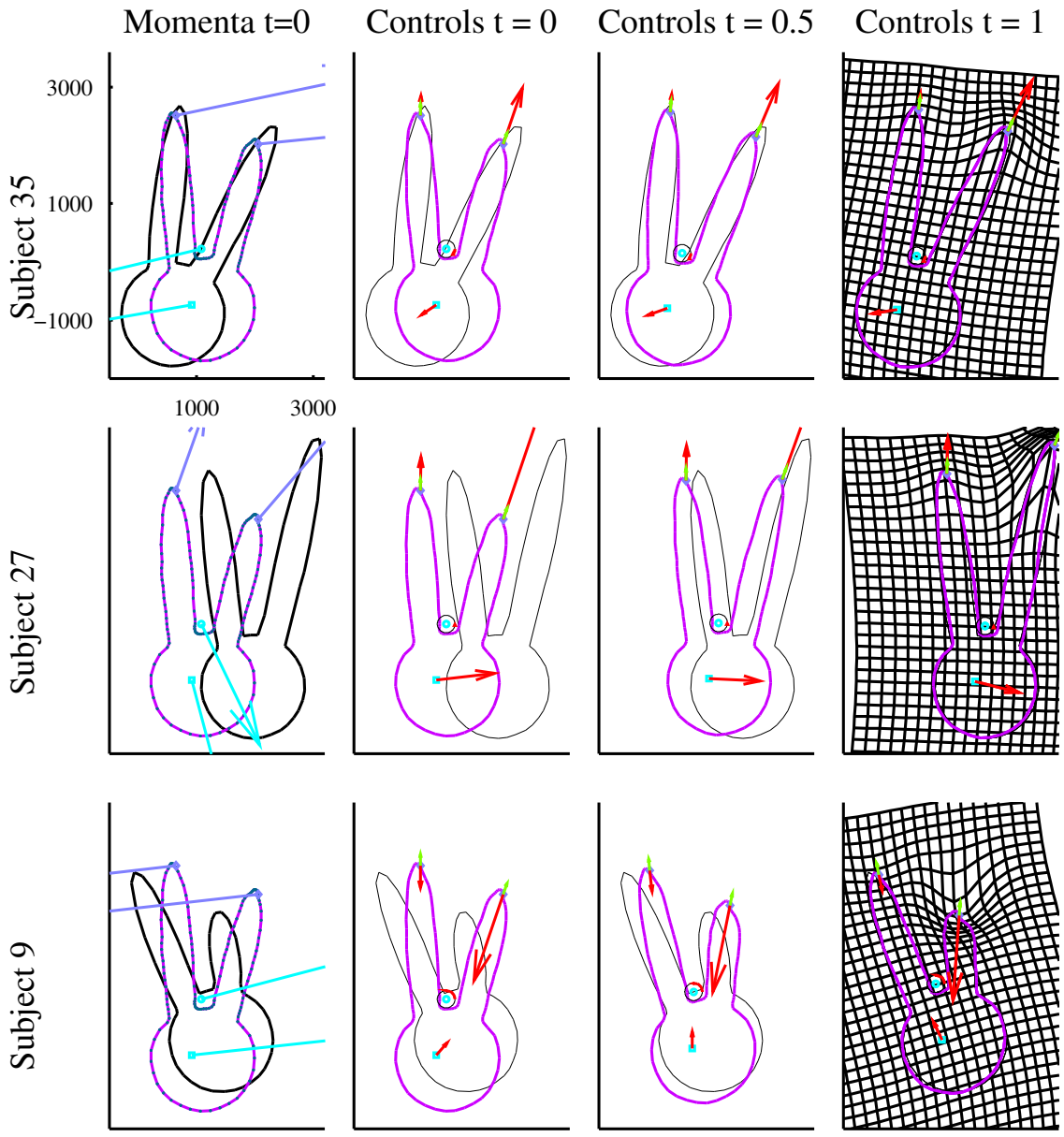


Figure 7.54: **Performing jointly rigid and non linear registration.** SIGS framework. **Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

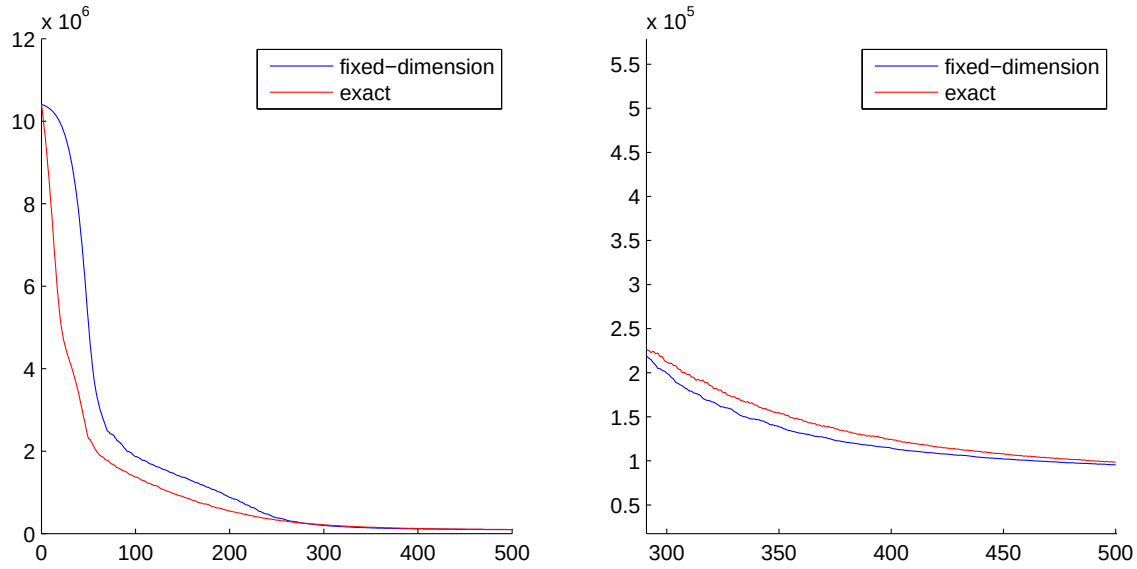


Figure 7.55: **Decreasing of the data attachment term for the two proposed frameworks.** x axis: number of iterations, y axis: data attachment term. **Left**: total curve. **Right**: zoom.

momentum attached to the mean shape. We can see that, as a consequence, initial momenta for the SIGS framework are of larger norm. This comes from the fact that each component of the control is equal to the application of the momenta to a speed the corresponding deformation module can generate (see Proposition 29). Then here controls associated to the translation, rotation or a translation with prior on direction will be influenced by points which are close to their centre with respect to their scale. Therefore in the SDGS framework momenta attached to geometrical descriptors which are close to one of these centres will increase the norm of the corresponding control. In the SDGS framework momenta attached to the mean shape will be taken into account on the computation of controls while in the SIGS framework, only momenta attached to centres and vectors of the translations, rotation and translation with prior on direction will count. Then in the second case they have to be larger in order to obtain controls with norms of the same order as in the SDGS framework.

7.3.4 Recovering the dimension of the population

The population of shapes we study is of dimension 5 as in fact they all derive from the same rabbit-head profile which has been rigidly translated (dimension 2), rotated (dimension 1) and whose ears' lengths have been changed (dimension 2). In order to study whether we can recover this dimension from initial momenta, we perform a PCA on initial momenta. Then for each n from 1 to 10, we project momenta on the orthogonal of the subspace generated by the first n principal directions and we consider the ratio, for each momentum, of the norm of this projection with the norm of the momentum. In Figure 7.56 we plot the maximum value taken by this ratio amongst the 40 momenta,

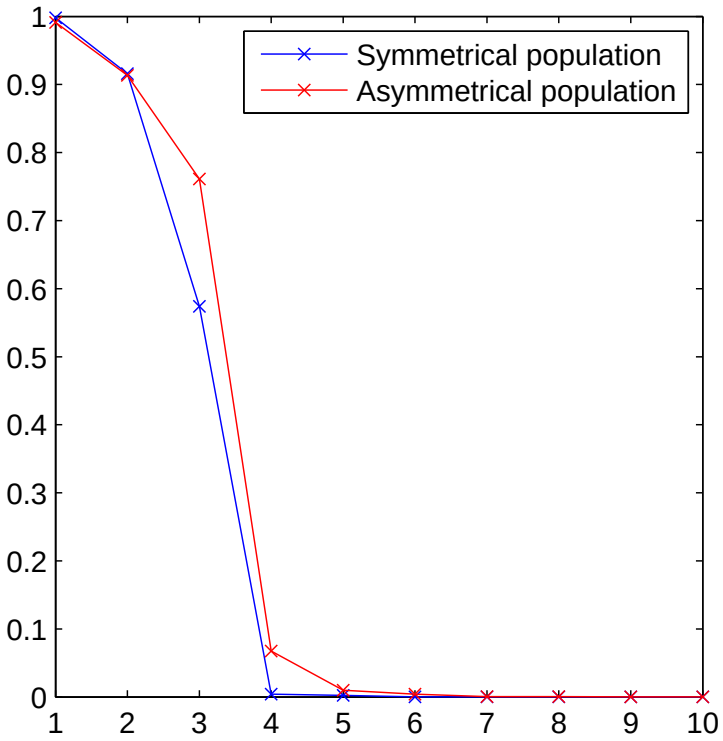


Figure 7.56: **Ratio of norms. x axis**: number n of principal directions , **y axis**: maximum ratio of the norm of the projection of momenta on the orthogonal of the subspace of first n principal directions and the norm of the momenta. In red is the plot for the population with the two ears of different sizes, and in blue for the population with the two ears of same size.

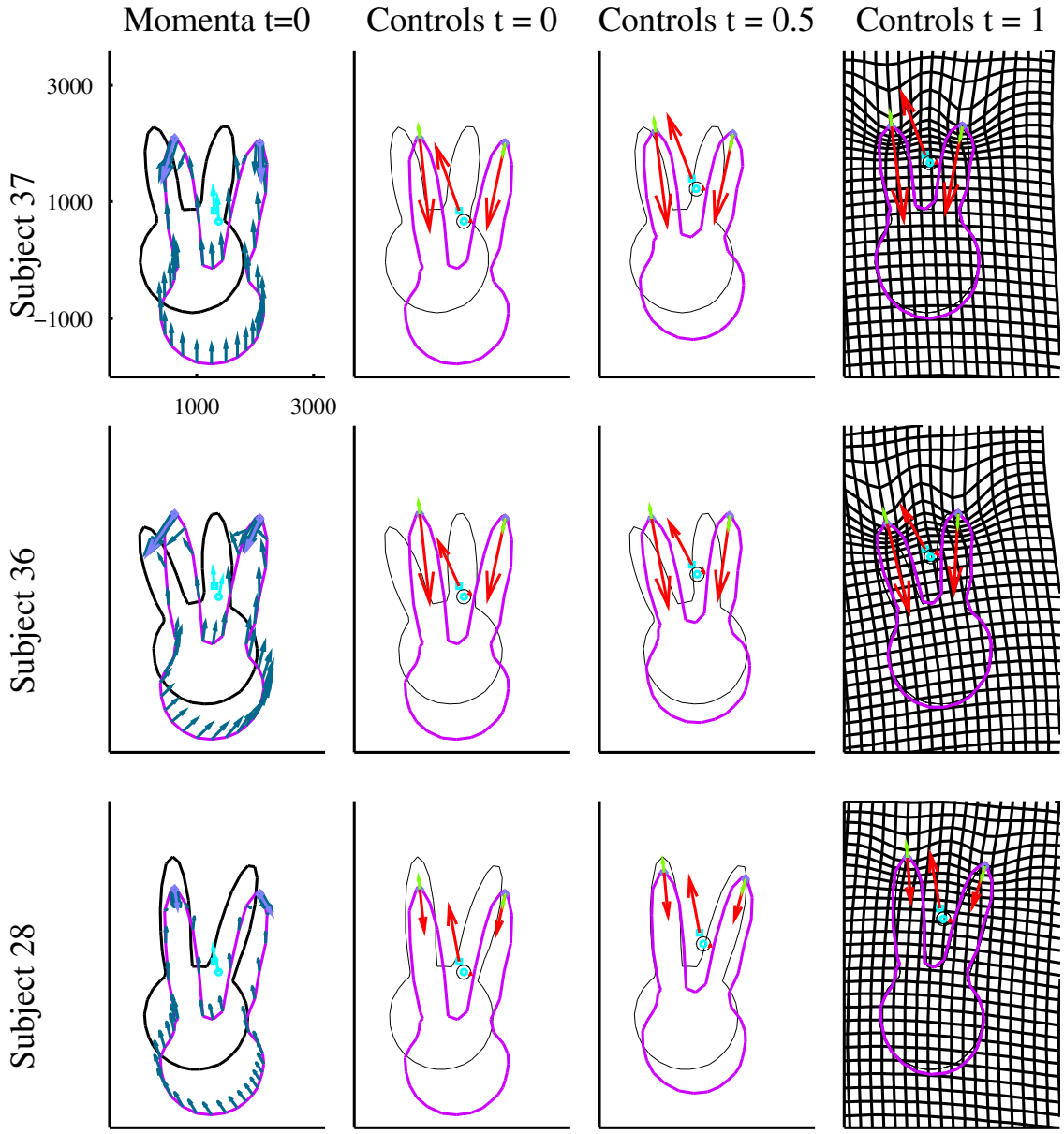


Figure 7.57: **Performing jointly rigid and non linear registration, symmetric population.** SDGS framework. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

for each number of principal directions n. We can see that for $n = 5$, for all subjects' momentum the residual norm of the projection on the 5-dimensional space is less than a percent of the norm which means that the 40 momenta roughly leave in this subspace of dimension 5. However, 5 is also the dimension of initial controls and then we need to be sure that the reason why momenta leave in a subspace of dimension 5 is because it is the dimension of the population and not because it is the subspace of the initial controls. In order to do so we study a new population of shapes which is built similarly but where the two ears of each subject are constrained to be of the same size: the dimension of this new population is 4. We compute an atlas of this population with the same deformation module, so that the dimension of initial controls is still 5. Three examples of geodesic trajectories can be seen in Figure 7.57. As previously we perform a PCA and study the norm of the projection on the orthogonal of the subspace of n principal directions for n from 1 to 10. We plot the maximum of the quotient of this residual norm and the norm of momenta in Figure 7.56. We can see that here only 4 dimensions are sufficient to generate these 40 momenta.

7.3.5 An example of non adapted parameters

We present now the same atlas computation as previously but with another scale for the translation and rotation: it is now 15000 instead of 5000. We show the results in Figures 7.58 and 7.59. It is at first surprising: vectors of the two translations with prior on direction are orthogonal to the direction of the ears. Besides at the end of the convergence the data attachment term is still 40 times larger than the regularity term: the problem comes from a local minimum, not from a too small weight for the attachment term. In fact we notice that vectors of the two translations with prior on direction are in the directions of the vector field that the rotation at scale 15000 generates at their centres. Besides we can see that the rotation at scale 15000 has controls almost equal to zero for all subjects. This enables to understand why the gradient descent converged to this configuration: it seems that the two translations with prior on direction generate locally the needed large rotation. However, whatever its scale is, the rotation needs the same control (and then the same cost) in order to realize a rotation at the same angle, so here the bigger scale of the rotation should not be a problem. However, with the same reasoning as in Section 7.2.5, we can see that the control of the large rotation is constant along geodesic trajectories and that its gradient with respect to initial momenta is given by the application of this rotation to geometrical descriptors with a control equal to one. Besides, since the rotation is equal to 0 at its centre and since its scale is too large with respect to the size of the data, the application of this rotation to geometrical descriptors with a control equal to one gives almost a null speed. Then here the problem comes from the fact that the gradient with respect to initial momenta of the control of the rotation is too small with respect to gradients of other trajectories of controls. Then during the optimisation process, the control of the rotation remains almost equal to zero and, to compensate, the directions of translations with prior on directions align themselves with this needed vector field.
In order to verify this hypothesis, using the idea introduced in Section 5.2.3.3 (similarly to Section 7.2.5), we perform an atlas of the same population, with a deformation module

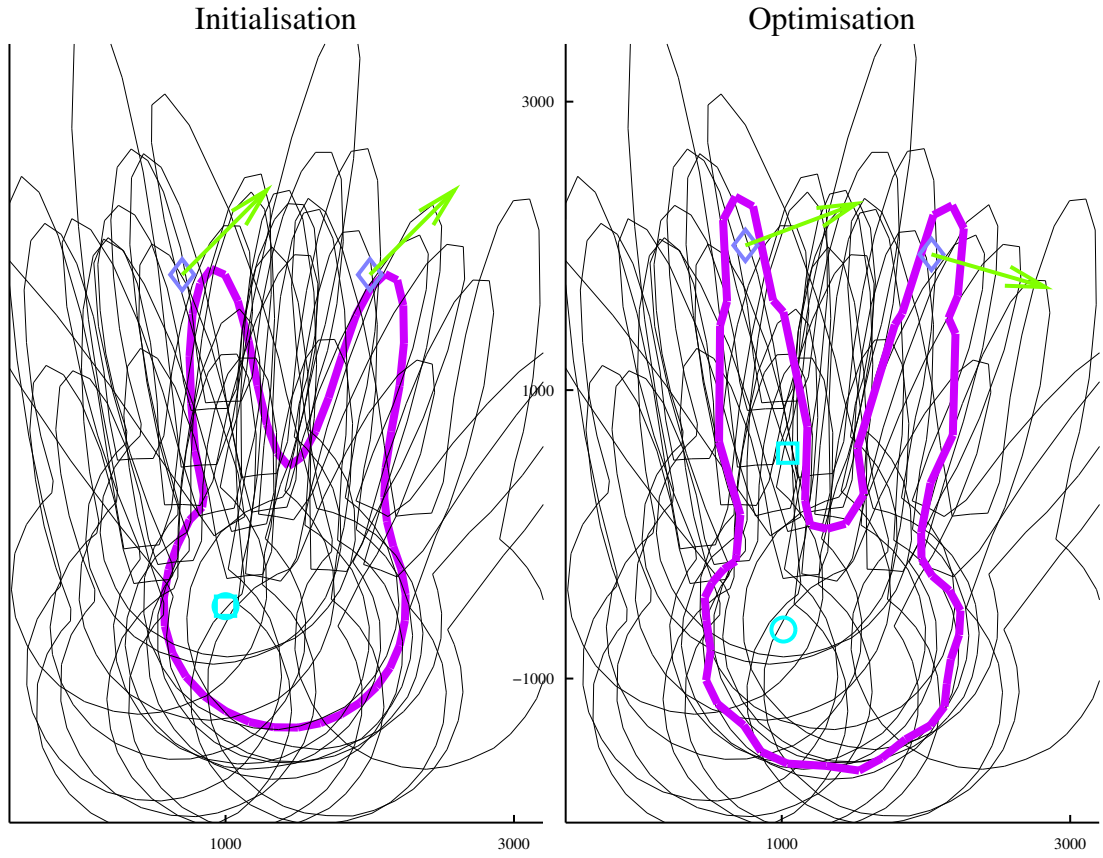


Figure 7.58: **Performing jointly rigid and non linear registration, symmetric population, large scale of rotation.** SDGS framework. **Template** at time $t = 0$ (purple shape) and other **geometrical descriptors**: rotation's one is the blue circle, the translation's ones is the blue square and translations with transported vectors' ones have their dot-component represented by blue diamonds and their vector components represented by green arrows. **Targets** in black. Left: Before optimisation. Right: After optimisation.

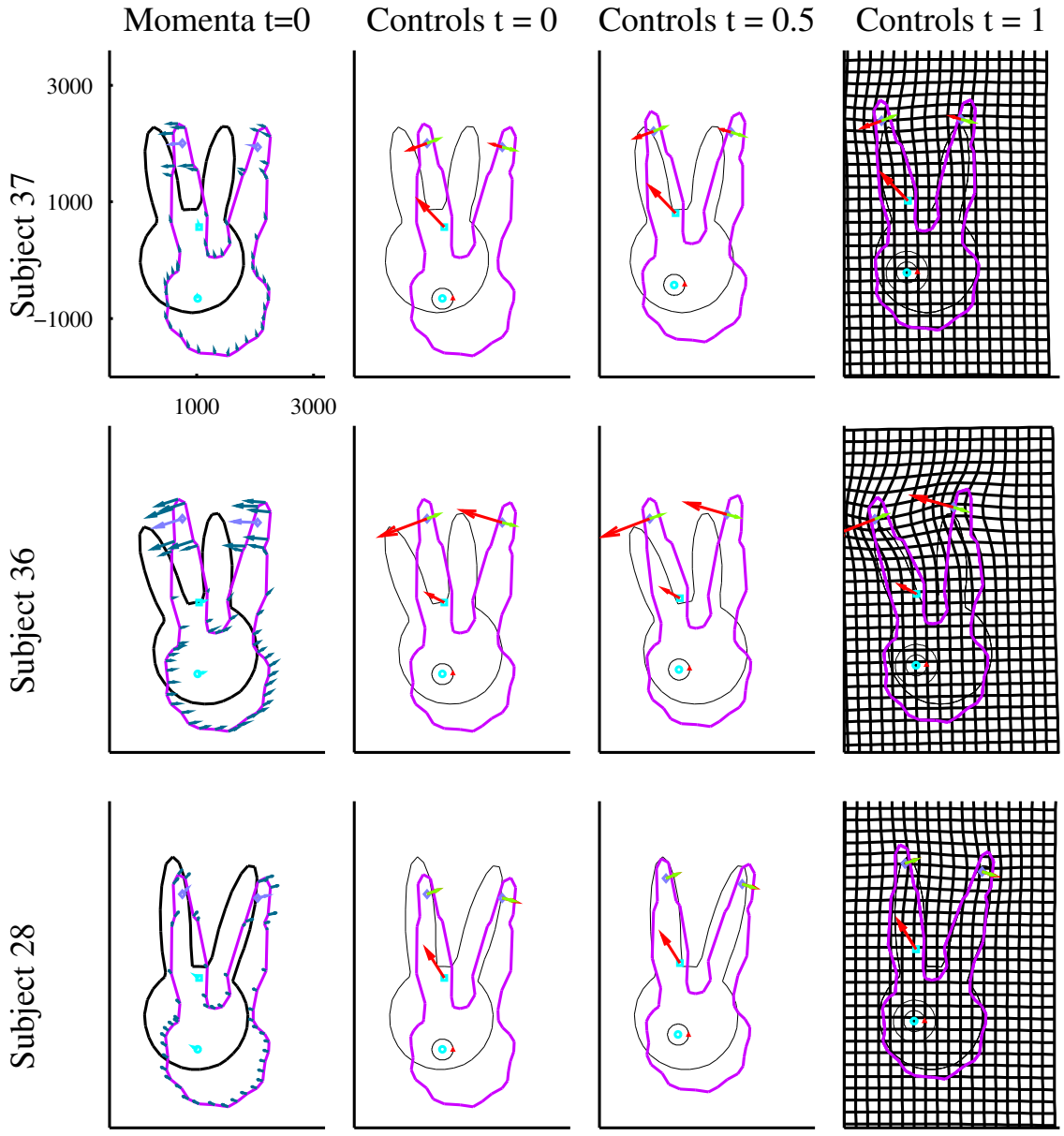


Figure 7.59: **Performing jointly rigid and non linear registration, symmetric population, large scale of rotation.** SDGS framework. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

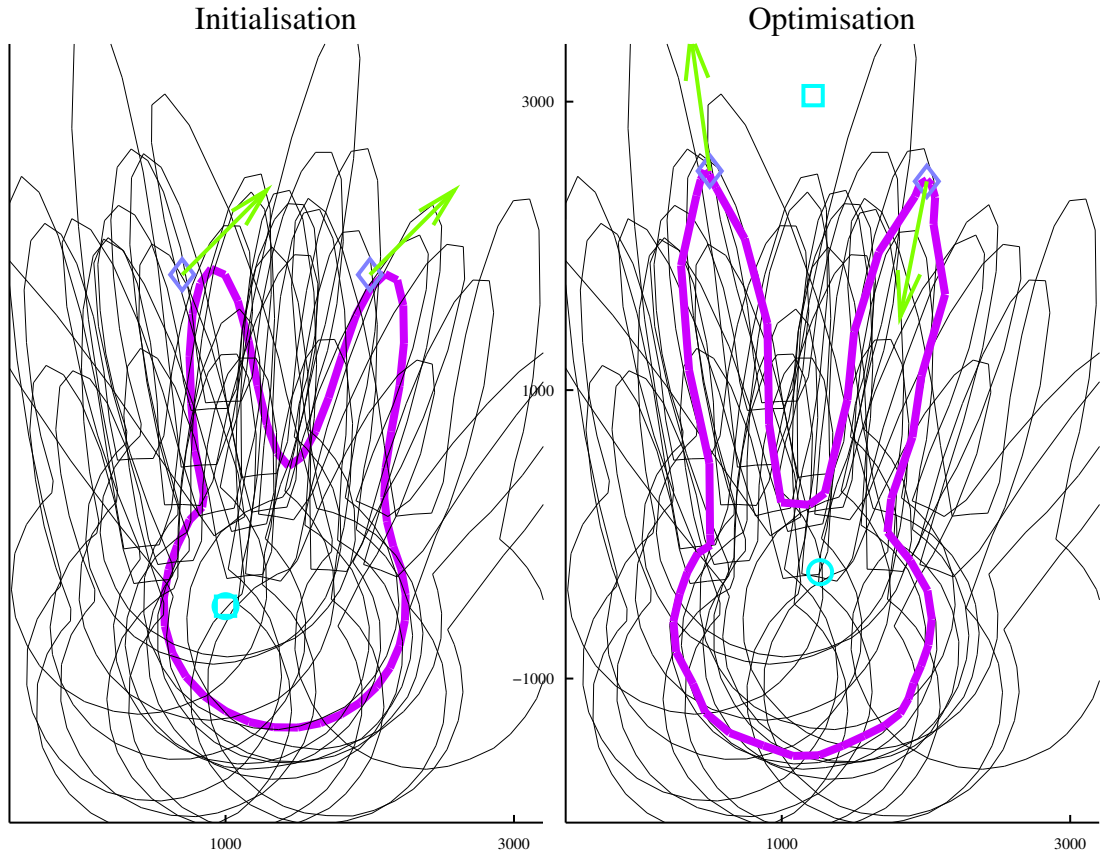


Figure 7.60: **Performing jointly rigid and non linear registration, symmetric population, large scale of rotation, change of metric.** SDGS framework. **Template** at time $t = 0$ (purple shape) and other **geometrical descriptors**: rotation's one is the blue circle, the translation's ones is the blue square and translations with transported vectors' ones have their dot-component represented by blue diamonds and their vector components represented by green arrows. **Targets** in black. Left: Before optimisation. Right: After optimisation.

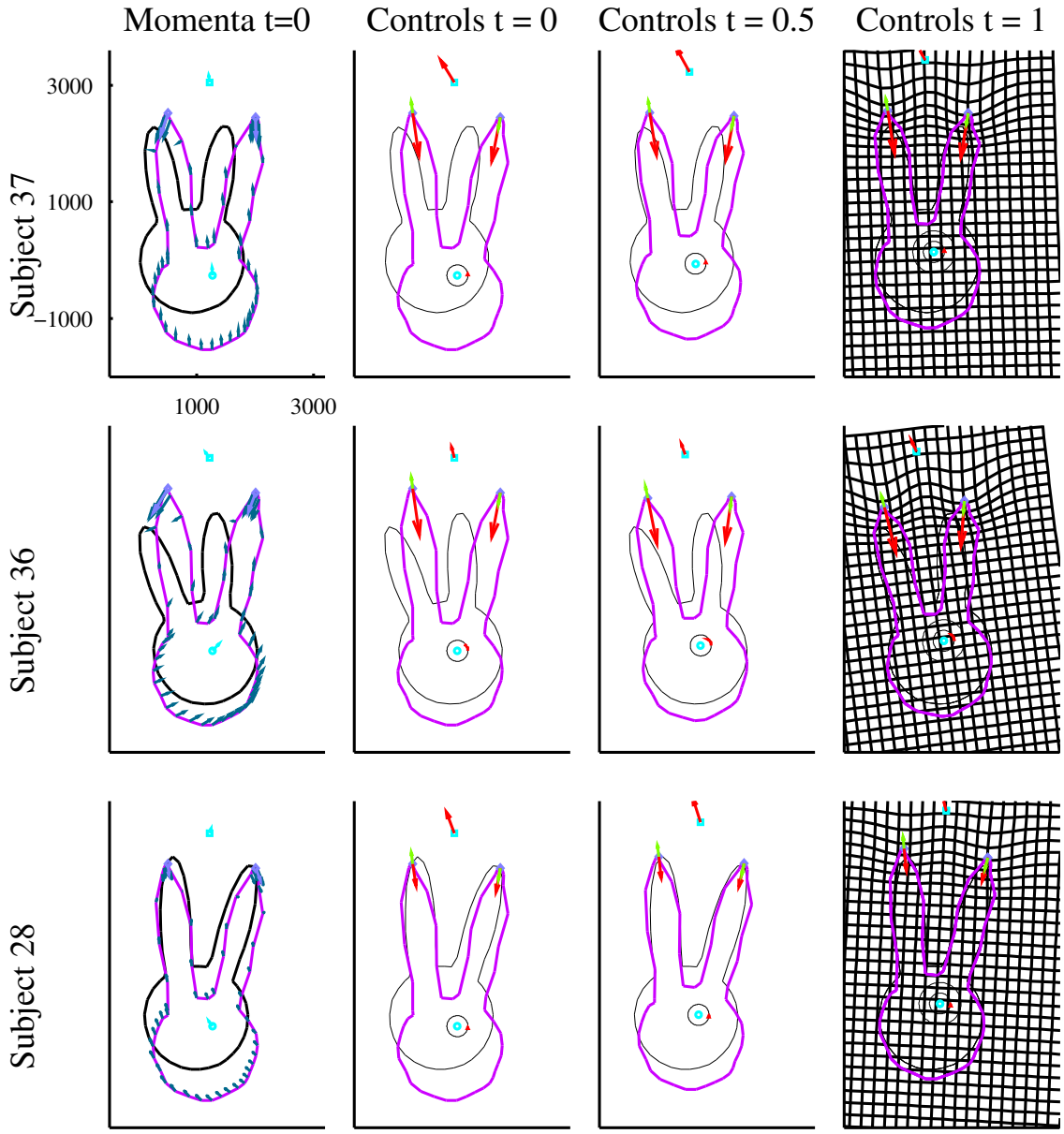


Figure 7.61: **Performing jointly rigid and non linear registration, symmetric population, large scale of rotation, change of metric.** SDGS framework. **Template** at time t (in purple), **target shapes** are in black. Other **geometrical descriptors** at time t: rotation's one at the blue circle, the translation's one at the blue square, translations with updated directions ones have their dot-component at blue diamonds and their vector-component represented by green arrows. Left column: **Momenta** at $t = 0$ in blue, attached to their geometrical descriptor (translations with update directions' ones have their vector-component represented in green, attached to the center of the translation). Three right-hand columns: **Controls** at time t : the red arrow attached to the blue square is the translation's one, the rotation's one is represented by the portion of the black circle coloured in red, and the one of translations with transported vectors are represented by the lengths and orientations of the red arrows attached to the blue diamond.

similar as previously except that we divide the cost of the rotation at scale 15000 by 10. Results can be seen in Figures 7.60 and 7.61: the control of the rotation is no longer null and vectors of translations with prior on direction are aligned with ears. We deduce that the problem came indeed from the weight of the rotation in the compound cost. This example shows that one needs to be cautious when choosing parameters of the deformation module and then interpreting optimized variables.

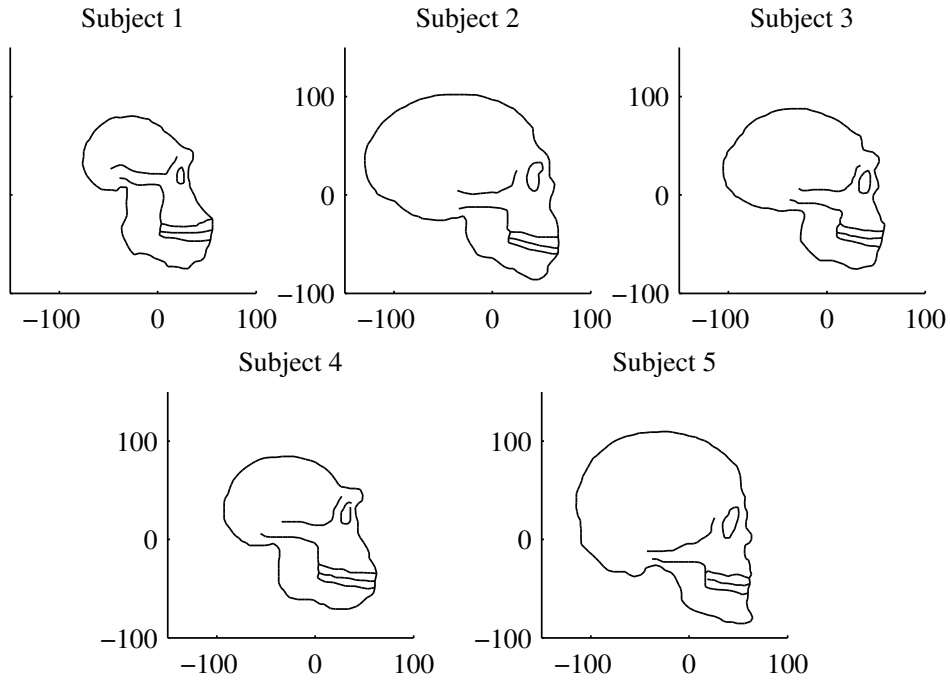


Figure 7.62: **Building an atlas with weak priors.** Target shapes

7.4 An example of atlas with a weak prior

In this last example we compute an atlas of the five shapes of skulls presented in 7.62. Similarly to previous examples, shapes are encoded as varifolds (see 3.3.2) so that no point-correspondence is assumed. Here, we do not have clear prior to include in the model. Therefore, we use a deformation module M obtained by combining 7 deformation modules: one generating a translation at large scale ($\sigma = 200$, see Section 4.2.2.1), one generating a rotation at large scale ($\sigma = 200$, see Section 4.2.3), one generating a sum of two rotations at middle scale ($\sigma = 100$, see Section 4.2.3), one generating a sum of two scalings at middle scale ($\sigma = 100$, see Section 4.2.3), one generating a sum of four rotations at small scale ($\sigma = 50$, see Section 4.2.3), one generating a sum of four scalings at small scale ($\sigma = 50$, see Section 4.2.3) and one generating a sum of 16 translations with prior on directions so that they are updated by adjoint action at small scale ($\sigma = 50$, see Section 4.2.2.3). The first two deformation modules (those at large scale) enable to perform rigid registration simultaneously with finer deformations. At the smallest scales, translations with prior on the direction have shared initial positions and directions for all subjects, and their directions are transported by the flow. Using this deformation module instead of one generating translation as in Section 4.2.2.1, allows to interpret initial vectors of these translations as directions of greatest variability among the population (at the small scale). In the first Section 7.4.1 we present the result obtained using this deformation module, in the following 7.4.3 one we present the result obtained using this deformation module in the SIGS setting (see Section 5.4.1), and in Section 7.4.4 we present the result obtained when we combine this deformation module

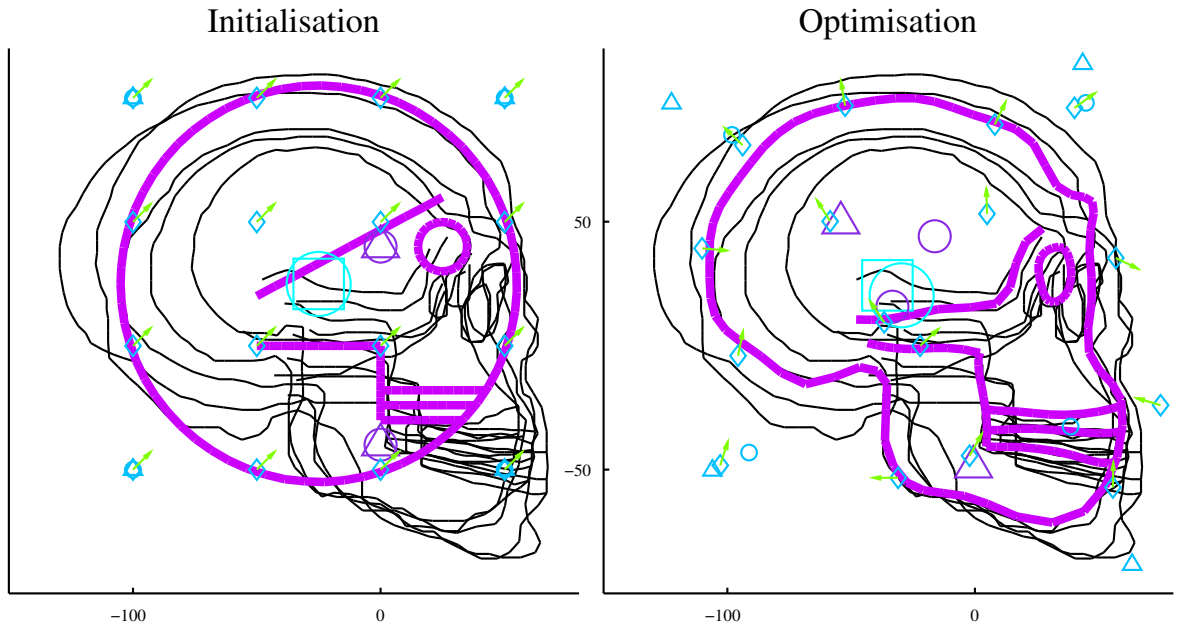


Figure 7.63: **SDGS framework. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

with a deformation module generating a sum of translations (see Section 4.2.2.1) at small scale.

7.4.1 SDGS framework

We first present the results obtained thanks to the SDGS framework (see Section 5.3.2). We recall that to perform this we combine the deformation module M with the silent deformation module (see Section 4.2.8) induced by the data shape space (here space of varifolds). We perform a gradient descent in order to estimate an optimal geometrical descriptors (made of an initial geometrical descriptor for deformation module M and an initial mean shape) and 5 initial momenta (one per subject). The initialisation of the geometrical descriptor can be seen in Figure 7.63 (left): the component corresponding to data shapes is initialized so that it has the same topology as data shapes, and other components are initialized on a regular lattice (one per type of deformation and scale). Initial momenta are initialized at zero. In Figure 7.63 (right) can be seen the initial value of geometrical descriptors after optimisation: they are updated during optimisation to regions of interest. Parametrization of minimizing trajectories (template, initial other geometrical descriptors and momenta) and trajectories of template, geometrical descriptors and controls can be seen in Figure 7.64 for three skulls. Controls associated to translations with prior on directions are scalars and are represented by lengths of

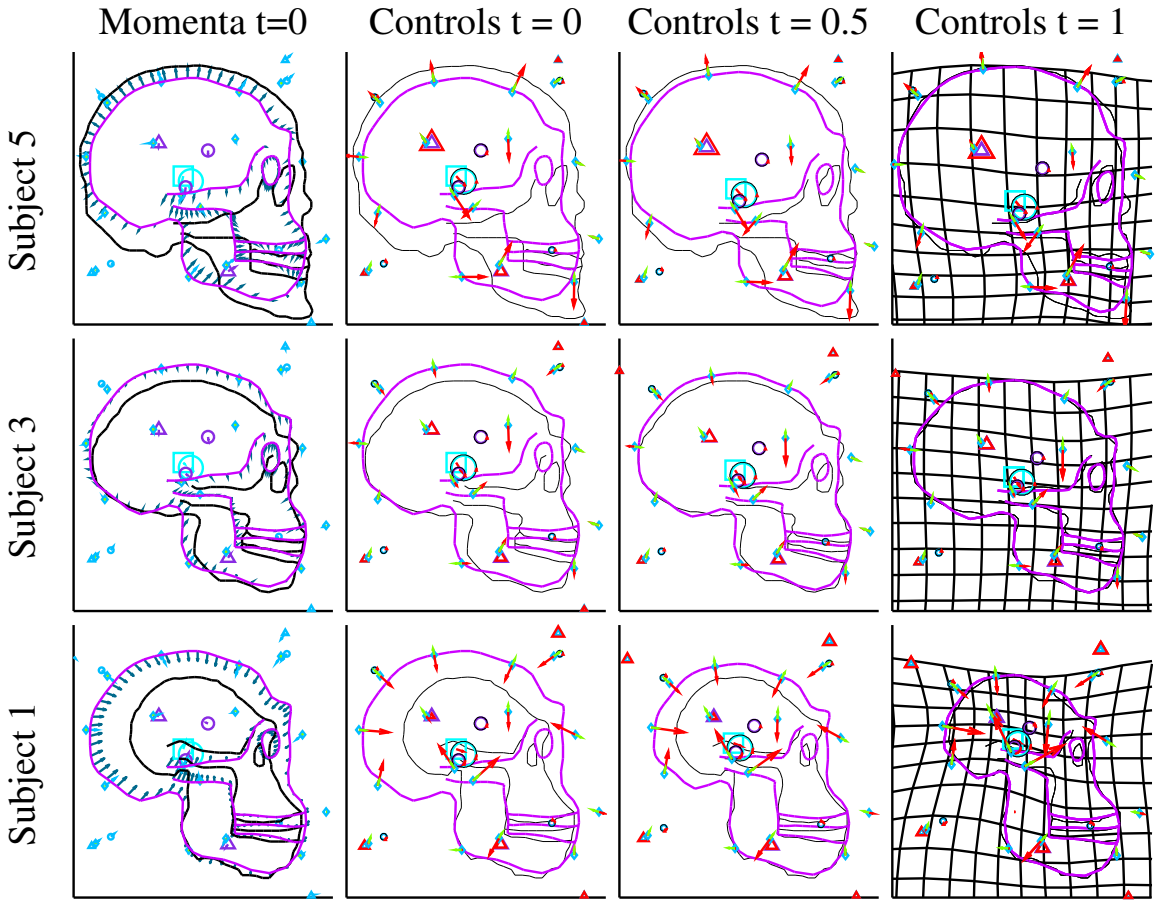


Figure 7.64: **SDGS framework. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

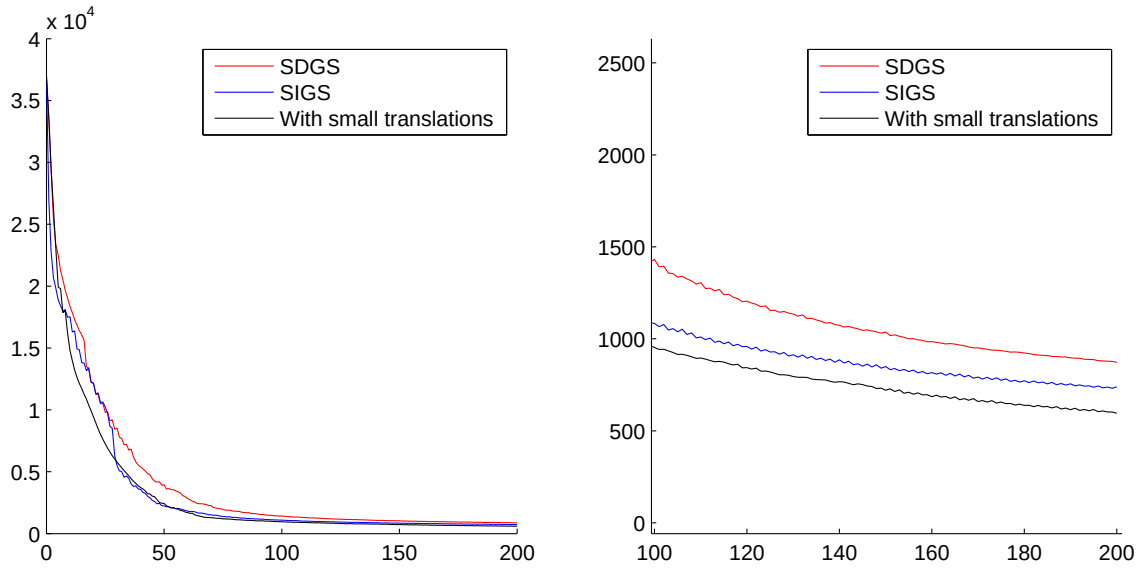


Figure 7.65: **Decreasing of the data attachment term for the three proposed frameworks.** x axis: number of iterations, y axis: data attachment term. **Left**: total curve. **Right**: zoom.

vectors attached to blue diamonds. Rotations have their geometrical descriptors represented by blue circles, and their controls (scalars) by the portion of these blue circles that are coloured in red (the control is positive if the colouring is anti-clockwise and negative otherwise). Geometrical descriptors of scalings are represented by blue triangles, and for each one a red triangle corresponding to the image of the blue one by this scaling enables to represent the scalar control. From these results one can see, for example, that the size of the cranium is a feature that varies importantly amongst the population as one center of the two local scalings at scale 100 is in this after optimisation, and trajectories of the corresponding control are very different for all skulls: negative for skull one (the red triangle is very small so the control is highly negative: his cranium is smaller than the template), close to zero for skull three (the red triangle is almost of the size of the blue one: his cranium is almost of same size as the template) and positive for skull five (the red triangle is bigger than the blue one: his cranium is bigger than the template). Other features of great variabilities can be deduced from this results such as, for instance, the shape of the back of the head or the area around the jaw and the nose, by inspecting the initial direction of vectors of the corresponding translations with prior on directions. The decreasing of the data attachment term (see Equation (5.12)) during the gradient descent is plotted in Figure 7.65.

7.4.2 Robustness with respect to initial conditions

In order to study the stability of these observed features with respect to initial conditions, we launched this experiment with four different sets of initial parameters. For the first three we use the same deformation modules but a different initialization for

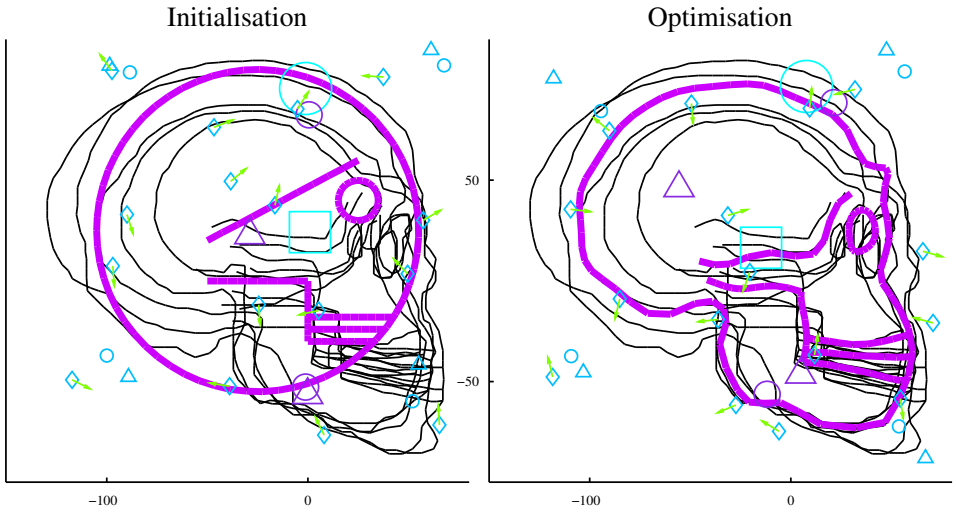


Figure 7.66: **Building an atlas without prior, a second initialization of geometrical descriptors. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

geometrical descriptors: we added a Gaussian noise with a variance equal to a quarter of the scale of each module. We present the results in Figures 7.66 to 7.71. One can see that the optimized template is very similar for these three experiments and the one we presented in the article. Besides a lot of deformation patterns are present in these four experiments, for instance a centre of scaling is always optimized in the centre of the cranes, and directions of translations at scale 50 show the same direction of high variability in most cases. Of course the stability of locations for geometrical descriptor has to be considered with respect to the corresponding scale. Besides if the control of one deformation module is always equal to zero in the trajectories, it is normal if the optimisation of the corresponding geometrical descriptor is not robust.

We launched a last experiment with different initial scales (but same initial geometrical descriptors): we used a translation at scale 180 (instead of 200), scalings at scales 110 (instead of 100) and 40 (instead of 50), rotations at scales 210 (instead of 200), 80 (instead of 100) and 40 (instead of 50), and translations with directions updated by adjoint action at scale 60 (instead of 50). The results are presented in Figures 7.72 and 7.73. Once again the deformation patterns are quite stable.

Note that we cannot really compare the results we obtain with a deformation module which would be too different because we would use another *vocabulary* to describe the variability of the population.

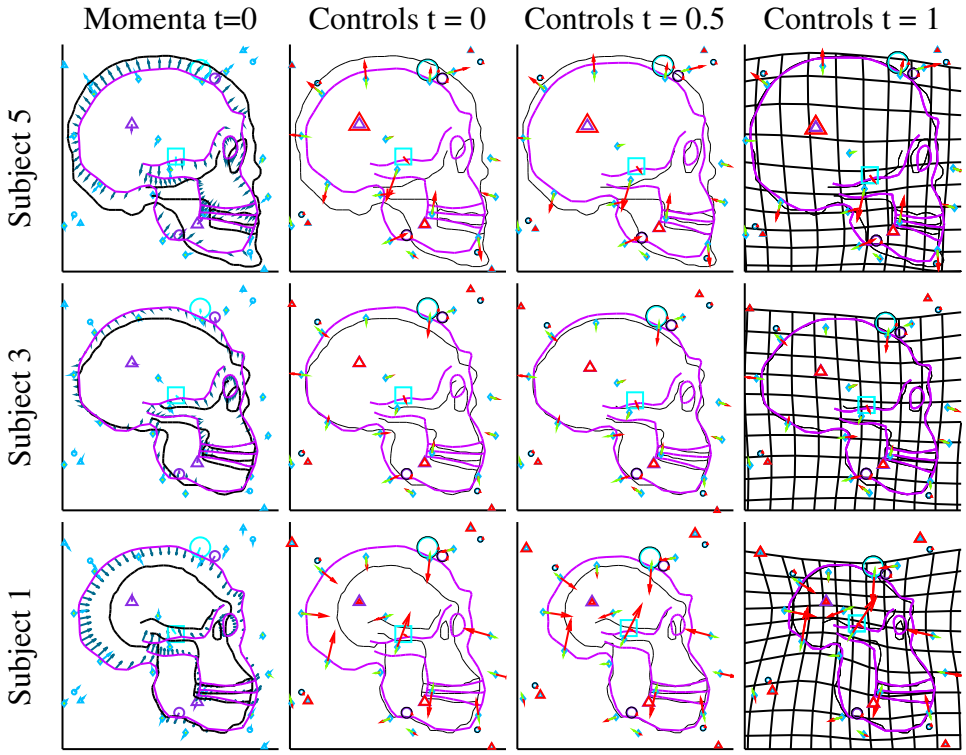


Figure 7.67: **Building an atlas with weak prior, a second initialization of geometrical descriptors. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

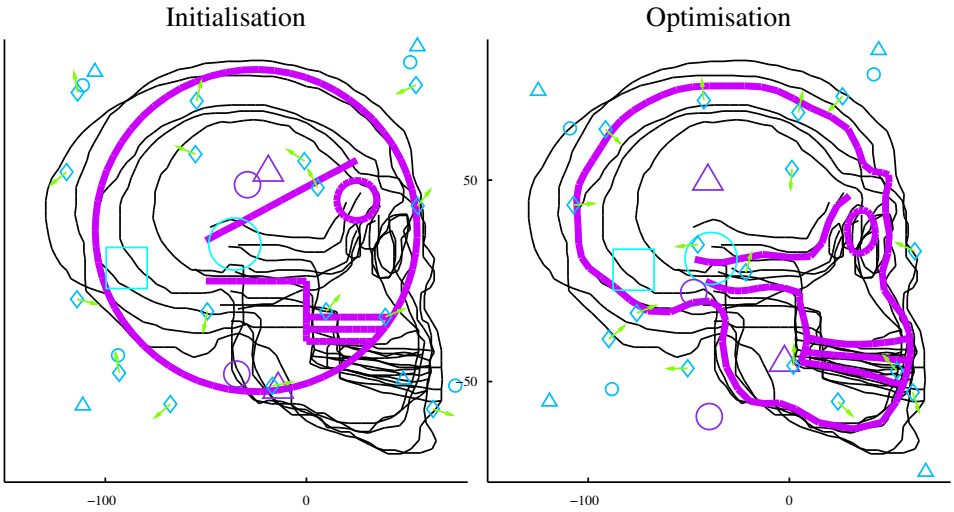


Figure 7.68: **Building an atlas without prior, a third initialization of geometrical descriptors. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

7.4.3 SIGS framework

7.4.3.1 Presentation

In the SDGS framework presented previously, we combine deformation module M with the silent deformation module induced by the data shape space. Then optimal deformations transporting the mean shape to each subject are parametrized by an initial momenta of this compound deformation module and its dimension is higher than the dimension of data shapes. As explained in Section 5.4.1, it is possible to restrict ourselves to deformations parametrized in smaller dimension: we consider optimal trajectories that can be generated by M and then search, amongst them, the ones that enable to transport the mean shape as close as possible to data shapes. In this framework we estimate one mean shape, one initial geometrical descriptor (of M, so only centers of translations, rotations, scalings and directions of translations with prior on directions) and 5 initial momenta. Here momenta are dual variable for deformation module M and then are of dimension of its geometrical descriptor: 92 (2 for each translation, scaling, rotation and 4 for each translation with prior on direction). As previously these geometrical descriptors are initialized on a regular lattice, and momenta are initialized at zero. Initial position for the mean shape and geometrical descriptor are presented in Figure 7.74 before (left) and after (right) optimization. We also present in Figure 7.75 the parametrization of minimizing trajectories (template, initial geometrical descriptors and momenta) and trajectories of template, geometrical descriptors and controls. It can be seen that in this SIGS framework there is no momentum attached to the template.

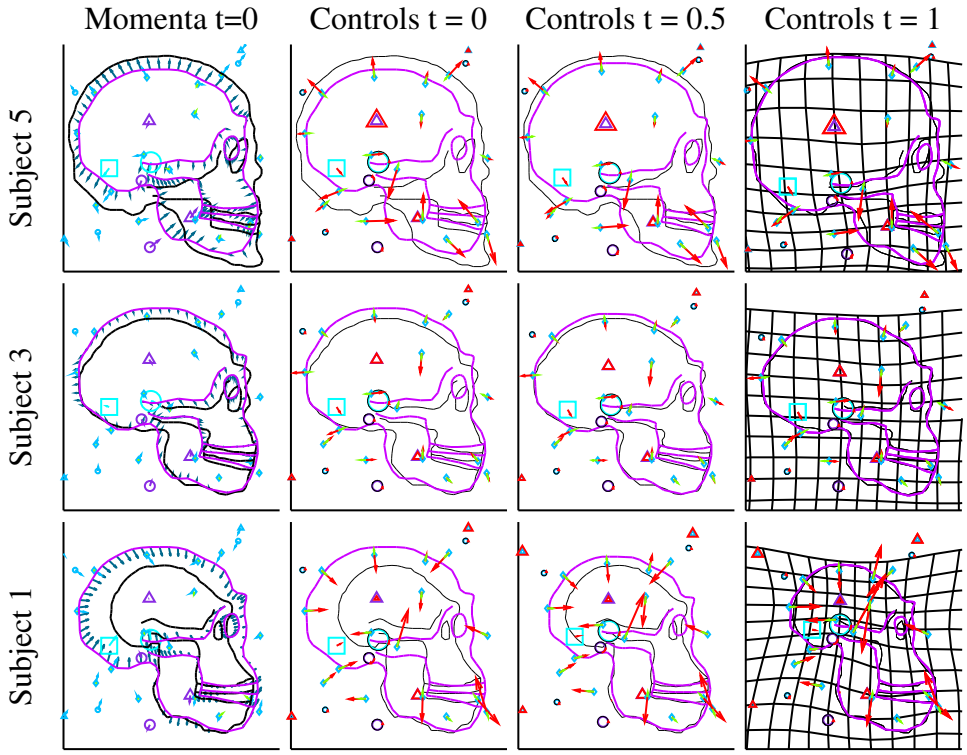


Figure 7.69: **Building an atlas with weak prior, a third initialization of geometrical descriptors. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

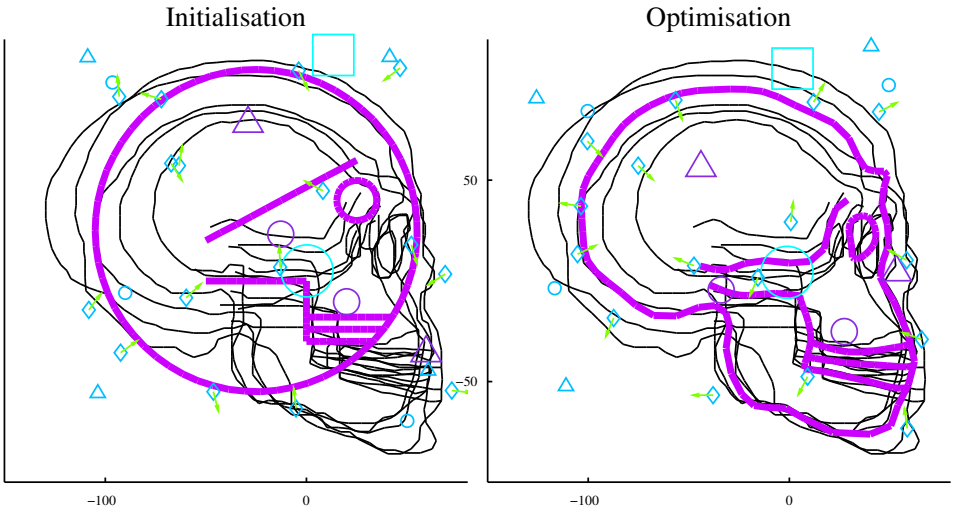


Figure 7.70: **Building an atlas without prior, a fourth initialization of geometrical descriptors. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

7.4.3.2 Comparison with the SDGS framework

By comparing Figures 7.63 with 7.74 and 7.64 with 7.75 we can see that initial geometrical descriptors and trajectories of controls are close between the two frameworks (even though not equal). Initial momenta on the contrary are necessary different since in the SDGS framework there is an additional momentum attached to the mean shape. For the same reasons as those presented in Section 7.3.3.2, initial momenta for the SIGS framework are of larger norm.

The decreasing of the data attachment term (see Equation (5.12)) during the gradient descent is plotted in Figure 7.65. We can see that this SIGS framework not only allows to decrease the data-attachment term, but also enables to obtain a smaller one for a same number of iterations. First this shows that the variability among this population can be well captured thanks to the this framework. The fact that better registration can be performed here thanks to the SIGS framework is probably due to the fact that in this framework the dimension of the parameters (initial momenta) to optimize are smaller than in the SDGS framework, and then this parameter is more easily estimated.

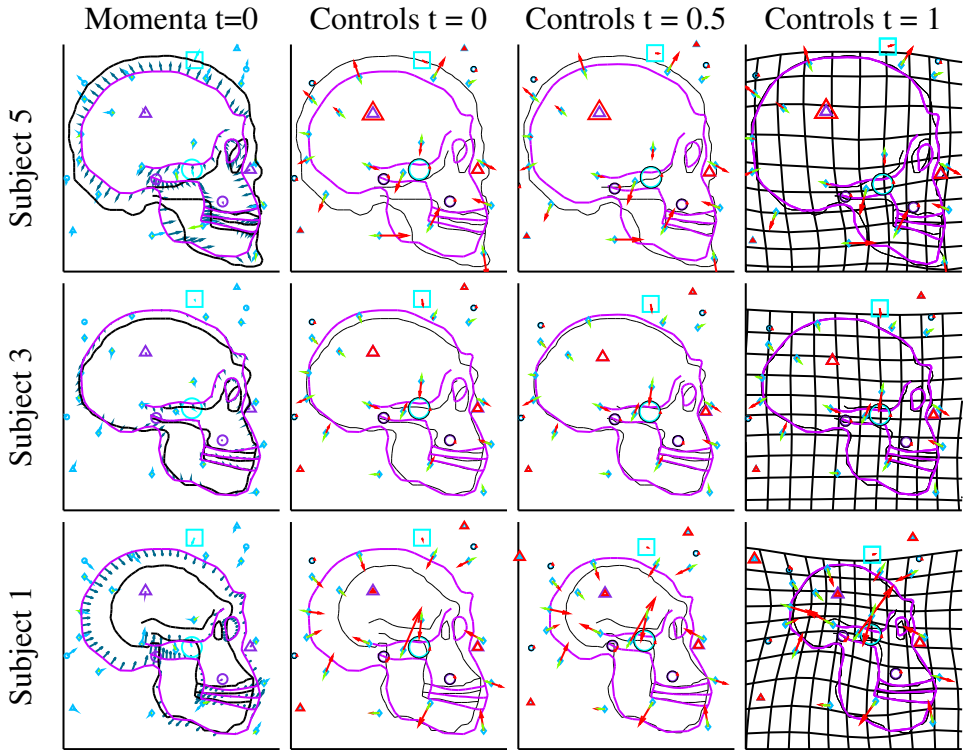


Figure 7.71: **Building an atlas with weak prior, a fourth initialization of geometrical descriptors. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

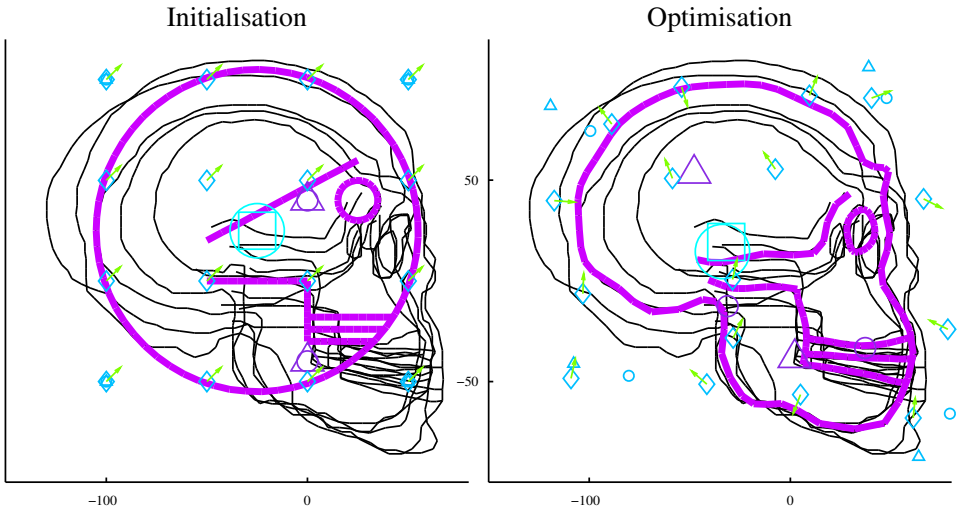


Figure 7.72: **Building an atlas without prior, different scales. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (210, 180,110,80,60,40). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

7.4.4 With additional small translations

We now combine M with a deformation module generating a sum of 36 translations at scale 30, and we compute the atlas using the SDGS framework. Unconstrained local translations enable to generate rich deformation fields and then by combining this new deformation module we will allow additional, almost unconstrained, local displacements.

Initial geometrical descriptors before and after optimisation are plotted in Figure 7.76. One can see that initial centres of translations do not migrate much during the optimisation process, this is due to the fact that they are relatively close to each other (and to areas of interest) with respect to their scale (the space between to consecutive centres is equal to the scale). In Figure 7.75 are presented the parametrization of minimizing trajectories (template, initial geometrical descriptors and momenta) and trajectories of template, geometrical descriptors and controls for three subjects. Trajectories of controls for scaling and rotations and the large translation are close to trajectories obtained without these small translations. Centres and directions of translations with prior on the direction are also close to the first framework, but one can see that local displacements at small scale is carried both by these translations with prior on directions and by the unconstrained ones.

In Figure 7.65 we plot the decreasing of the data attachment term during the gradient descent. One can see that this new deformation module enables the data attachment term to decrease more than when the deformation module M is used. This is probably due to a better approximation to fine necessary displacements. Nevertheless the difference is small with respect to initial data attachment term, suggesting that the first

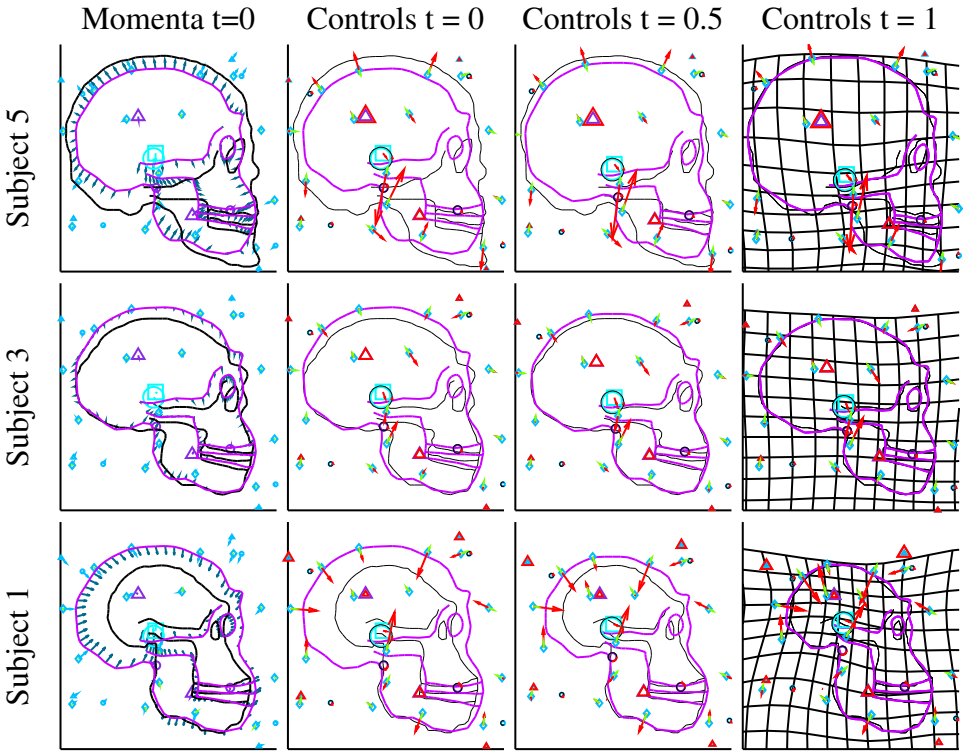


Figure 7.73: **Building an atlas with weak prior, different scales. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\diamondsuit$ (translations with updated directions). The color and size of markers represent the (210, 180,110,80,60,40). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

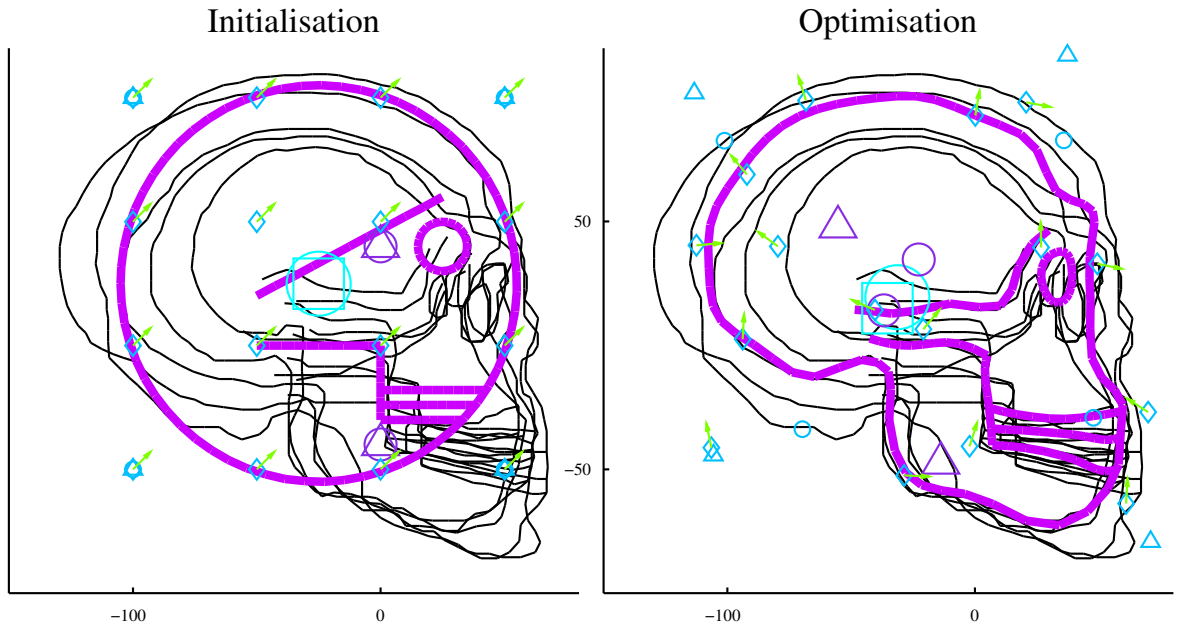


Figure 7.74: **SIGS framework. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

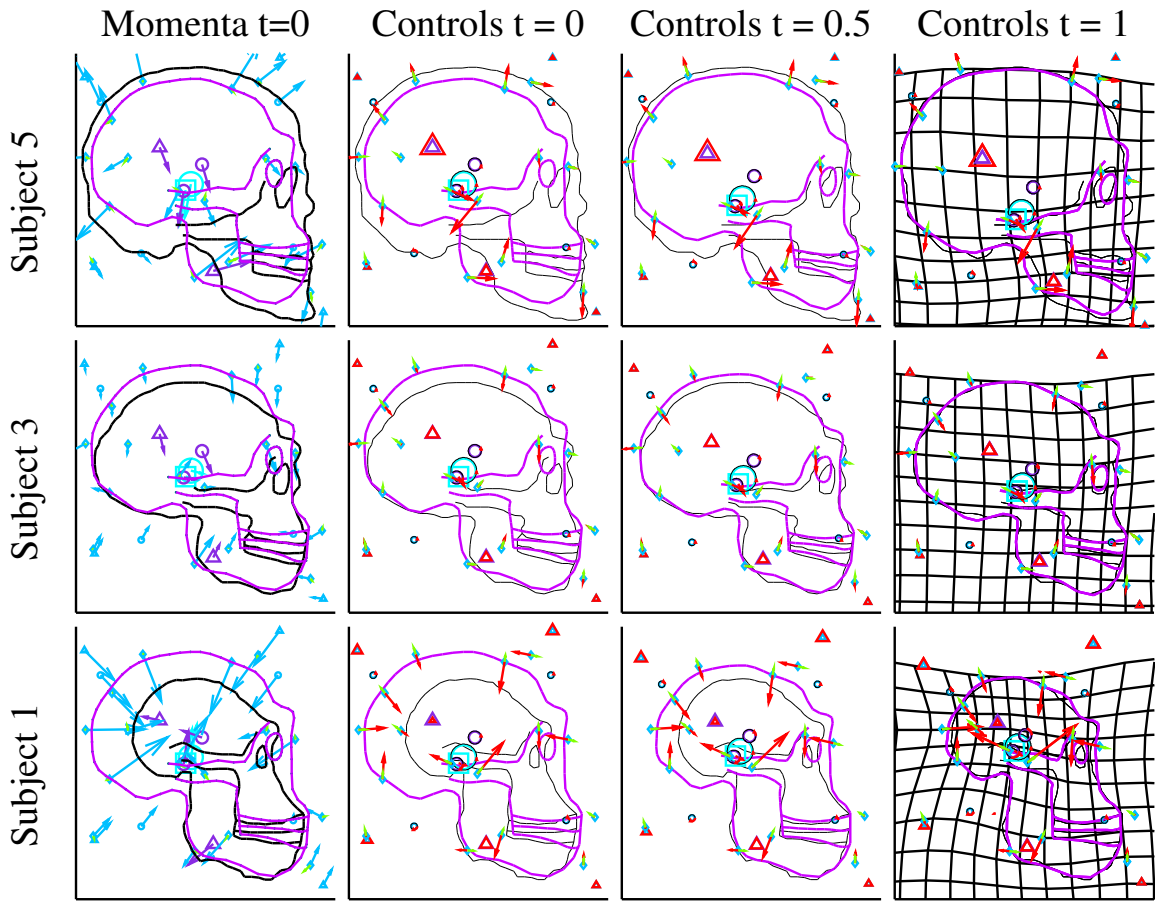


Figure 7.75: **SIGS framework. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vector for the translation, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

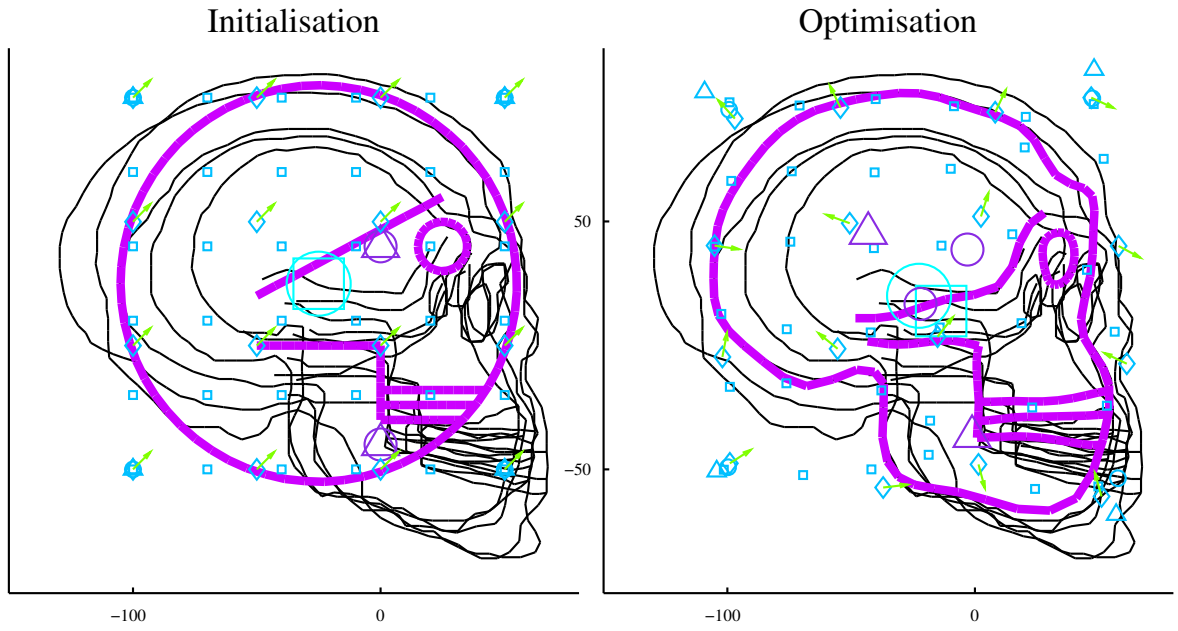


Figure 7.76: **With additional small translations. Template** at time $t = 0$ (purple shape) and **geometrical descriptors**: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50,30). Black shapes are the targets. Left: Before optimisation. Right: After optimisation.

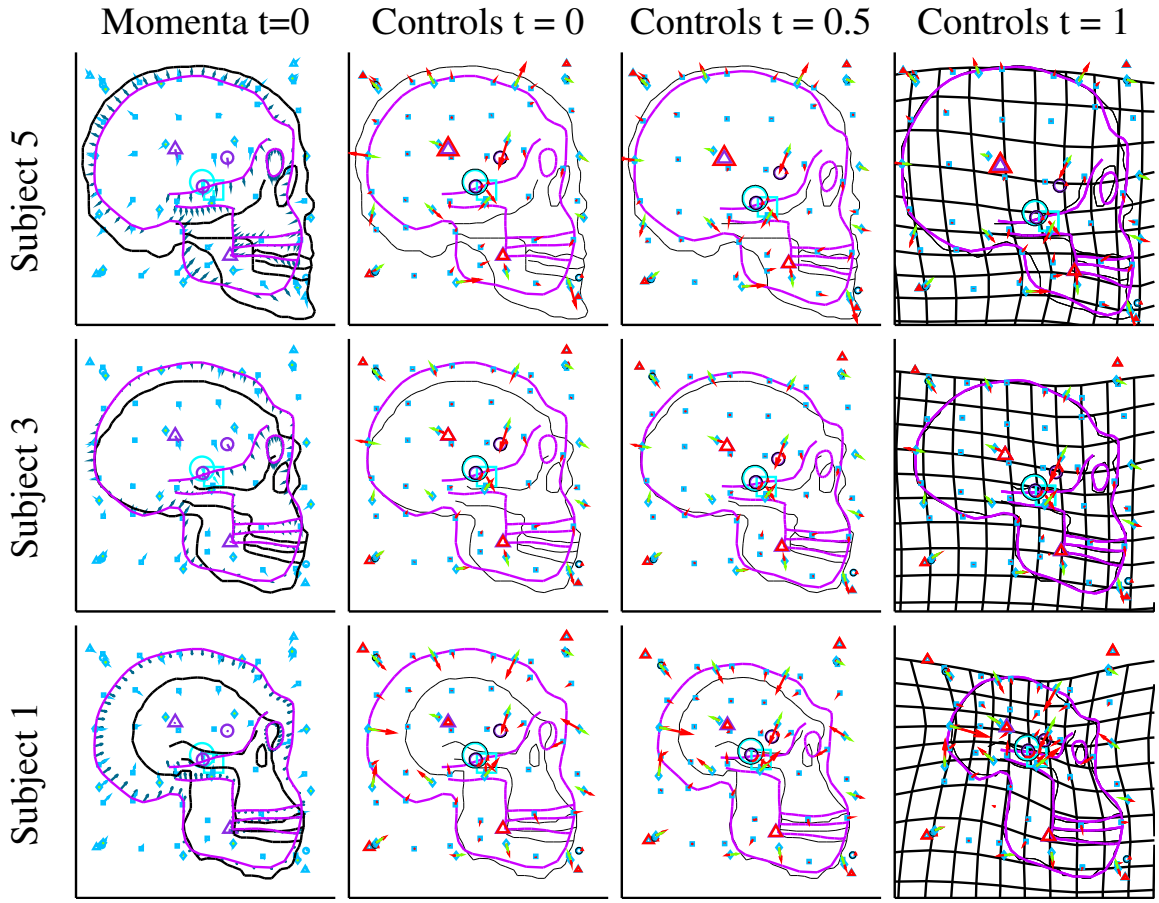


Figure 7.77: **With additional small translations. Template** at time t (in purple), **target shapes** are in black. **Geometrical descriptors** at time t: $\square$ (translations), $\circ$ (rotations), $\triangle$ (scalings) and green vectors attached to $\Diamond$ (translations with updated directions). The color and size of markers represent the scale (200,100,50,30). Left column: Momenta at $t = 0$. Three right-hand columns: **Controls** at time t in red (vectors for the translations, portion of the blue circle coloured in red for rotations, ratio between the blue and red triangles for scalings and length of the red vectors for translations with updated direction).

deformation module M enables to capture the variability amongst the population in a satisfactory manner.

Chapter 8

Conclusion and perspectives

Contents

In this thesis we presented a new deformation model enabling to incorporate naturally and easily complex constraints. This enables to study shapes from a chosen point of view, which corresponds to a prior in the deformation model. We showed that our framework enables to study a large diversity of shapes, thanks to a large diversity of priors. In particular, it is possible to incorporate very strong priors (for instance in Section 7.1) as well as very weak ones (for instance in Section 7.4). Our framework can be used to study differences between shapes with two methods: the matching framework for registering a shape onto an other one, and the atlas computation to study the variability amongst a population of shapes. In both these methods the idea is to fix a deformation module (implementing the *prior*) and to estimate an initial geometrical descriptor as well as one (or one per subject in the atlas computation) modular large deformation generated by this geometrical descriptor. The geometrical descriptor corresponds to a geometrical characterization of the shape variability, using the *vocabulary* allowed with the chosen deformation module. The geometrical descriptor is composed of two components: an active one which will generate the deformation, and a silent one which is a shape of the data shape space. In the matching problem the silent one is equal to the source shape and then is not optimized, while in the case of the atlas computation, the silent one is the template (corresponding to a *mean shape*) and is optimized. Modular large deformations are parametrized thanks to a dual variable of the total initial geometrical descriptor called the momentum. We emphasized in Chapter 5 that the momentum of the silent component of the initial geometrical descriptor has a fundamental role in the generation of large deformations, even though this component does not directly creates a vector field. Our framework raises many questions and perspectives, let us present them.

8.1 Application to shapes in dimension 3

In all examples that we presented in Chapter 7, the studied shapes are curves embedded dimension 2. However, as explained in Section 4.2, building deformation modules in higher dimension is as simple as building them in dimension 2. This is the object of current considerations. An interesting further work would also be to incorporate image data in order to use icono-geometric shapes similarly to [FDY+14].

8.2 Role of the initial momentum

In Sections 5.2.1 and 5.2.3 we studied the link, for a fixed geometrical descriptor o, between the initial momentum η and the associated geodesic control $h^{o,\eta}$. Remark 23 links in a very intuitive way the coefficients of the geodesic control in an orthonormal basis with the momentum. Besides it allows to re-deduce Proposition 26 which states that if $F_{o,\eta} \subset \xi_o \circ \zeta_o(H)$ is the orthogonal in $\xi_o \circ \zeta_o(H)$ (for the metric g_o given by the cost) of the velocity $\xi_o \circ \zeta_o(h^{o,\eta})$ generated by o and η, then η is orthogonal for the co-metric to $F_{o,\eta}$. We also saw in Proposition 25 that the geodesic vector field $\zeta_o(h^{o,\eta}$ can be considered as a projection of the dual variable $\xi_o^*(\eta)$ brought in $\zeta_o(H)$ thanks

to its kernel. All these properties allow to understand how the initial momentum and the initial vector field are linked. However, as our framework is a Sub-Riemannian one (and not Riemannian), the initial vector field does not determine the geodesic trajectory. We introduced several examples in Sections 5.4.1.2, 5.4.2.2 and 5.4.2.3 showing that the dimension of geodesics, and the influence of the initial momentum on them is not trivial. The complete geometrical understanding of the role of this initial momentum is still to be performed. In particular if the deformation module we use is the combination of several base-deformation modules, it would be interesting to understand whether the influence of the different types of vector fields which are combined (through the combination of deformation modules) can be recovered from the initial momentum. It is important to note that studying the influence of a particular type of vector field is not equivalent to studying the influence of a particular deformation module since different deformation modules may generate locally similar vector fields. Another point of view on the initial momentum could be the statistical one: by studying the distribution of initial momenta of a population after an atlas computation could enable to understand their role in the geodesic trajectories. In particular we showed in the simple example of Section 7.3 that a Principal Component Analysis performed on initial momenta allowed to recover the dimension of the population. In a more general setting, performing a Principal Geodesic Analysis [Fle04, FLPJ04] might help extracting a meaningful sub-manifold of initial momenta.

8.3 Influence of the cost

In the definition of a deformation module, the choice of the cost is central but its influence has still to be studied. In particular, in the definition of the compound deformation module, the compound cost is built as the sum of costs, but it could be defined as a weighted sum. We explained in Section 5.2.3.3 how weighting this sum influences the geodesic cost associated to a given geometrical descriptor and a given momentum. Besides we gave two examples in Sections 7.2.5 and 7.3.5 suggesting than one can favour the "use" of one of the combined deformation module by lowering its weight in the compound cost.
In a more generic framework, one could multiply the matrix representing the metric g_o by a chosen matrix, allowing more complex compound costs. In order to determine these weights a statistical framework should be developed.
Note also that if a deformation module satisfies UEC, and if we modify its cost by adding a smooth function of (o, h) which is a positive semidefinite quadratic form on H, we obtain again a deformation module satisfies UEC. This way it could be possible to take into account another type of regularity.

8.4 Selection of models and interpretation of variables

In Section 7.2 we studied the stability of optimized geometrical descriptors and geodesic trajectories for the matching problem. We showed that optimized initial geometrical descriptors cannot really be interpreted when the deformation module which is used is not totally adapted in terms of constraints. For instance if a geometrical descriptor is not used to generate a vector field, its position might move during the optimisation process without its optimized position to be meaningful in terms of generated vector field. On the opposite, we saw that if the constraints are adapted, then the initialization of initial geometrical descriptor seems more stable, in particular to rigid translations of data. The same phenomenon occurs with trajectories of vector field. In practice determining whether a particular geometrical descriptor's position is meaningful or not can be intuited. But it seems necessary to develop a rigorous and systematic method to determine it. In a more general setting, it would be very useful to elaborate a method allowing to assess how adapted a deformation module is to a certain population of shapes. Then different deformation modules could be used to compute an atlas of a population of shapes, and knowing which one is the most adapted would give an additional knowledge on this population of shapes. Besides, this would remove the subjective choice of the deformation module.

8.5 Building a vocabulary

The question of the optimization of initial geometrical descriptor for an atlas case is different from the matching case. Indeed in the second case one needs to estimate one position for them allowing to transport a source shape onto a target shape, while for the atlas computation one needs to estimate one position which allows to transport a common template shape onto all subjects. Then in the atlas computation, the notion of vocabulary is particularly relevant: by choosing a particular deformation module to compute the atlas of a population of shapes, we use a particular vocabulary to describe its variability. We showed in Section 7.1 that several choices of vocabulary could lead to several template estimation. Similarly to the matching case, it would be interesting elaborate a method to compare the relevance of two choices for this deformation module. In the matching case this method should rely on the quality of the registration, while in the atlas case it should takes into account the quality and the relevance of the variability description. This could be performed thanks to probabilistic framework, by estimating the likelihood of a population through the repartition of initial momenta. For instance in Section 7.1.1, even though the two priors allow to describe the shape variability, it seems unlikely to have folding and unfolding patterns while each subject has only one hump.

8.6 Statistical point of view

In this thesis we have favoured a geometrical study of our framework but it should be completed with a statistical study. First, in Section 5.2.1 we introduced a rigorous probabilistic framework to develop the inexact matching problem when initial and final geometrical descriptors are fixed. In Section 5.2.2 we generalized it to the case where only part of initial and final geometrical descriptors are known (the silent component) and that we want to estimate the unknown one (the active component). Then we again generalized it in Section 5.3 to the computation of atlas, where the initial geometrical descriptor it totally unknown, and only part of the final ones are known. However in order to develop the probabilistic reasoning of Section 5.2.1, we introduced a Gaussian probabilistic law centred in the initial geometrical descriptor, and in the maximization of the likelihood we could remove the normalization term of this Gaussian measure because, given the initial geometrical descriptor, it is constant. But it does depend on the value of the initial geometrical descriptor: here we neglect this influence but a more rigorous framework such as the one developed in [AAT07] should be built. We should also follow the lead of this article to build a coherent statistical framework for parameters of our deformation modules. In particular it seems necessary to estimate scales of deformations as well as the number of replications of local transforms that we use (for instance the sum of local translations or scalings).

Bibliography

[AAT07] Stéphanie Allassonnière, Yali Amit, and Alain Trouvé. Towards a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1):3–29, 2007.

[AB14] Andrei Agrachev and Ugo Barilari, Davide *Introduction to Riemannian and Sub-Riemannian geometry*. 2014.

[ACAP09] Vincent Arsigny, Olivier Commowick, Nicholas Ayache, and Xavier Pennec. A fast and log-euclidean polyaffine framework for locally linear registration. *Journal of Mathematical Imaging and Vision*, 33(2):222–238, 2009.

[AG04] Brian Avants and James C Gee. Geodesic estimation for large deformation anatomical shape averaging and interpolation. *Neuroimage*, 23:S139–S150, 2004.

[AK15] Stéphanie Allassonniere and Estelle Kuhn. Convergent stochastic expectation maximization algorithm with efficient sampling in high dimension. application to deformable template model estimation. *Computational Statistics & Data Analysis*, 91:4–19, 2015.

[AKT+10] Stéphanie Allassonnière, Estelle Kuhn, Alain Trouvé, et al. Construction of bayesian deformable models via a stochastic approximation algorithm: a convergence study. *Bernoulli*, 16(3):641–678, 2010.

[APA05] Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Polyrigid and polyaffine transformations: a novel geometrical tool to deal with non-rigid deformations–application to the registration of histological slices. *Medical image analysis*, 9(6):507–523, 2005.

[Arg14] Sylvain Arguillere. *Géométrie sous-riemannienne en dimension infinie et applications à l'analyse mathématique des formes*. PhD thesis, Paris 6, 2014.

[Arg15a] Sylvain Arguillere. The general setting for shape analysis. *arXiv:1504.01767*, 2015.

[Arg15b] Sylvain Arguillère. The general setting for shape analysis. *arXiv preprint arXiv:1504.01767*, 2015.

[Aro50] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[ATTY15] Sylvain Arguillere, Emmanuel Trélat, Alain Trouvé, and Laurent Younes. Shape deformation analysis from the optimal control viewpoint. *Journal de mathématiques pures et appliquées*, 2015.

[B+89] Fred L Bookstein et al. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[BBM14] Martin Bauer, Martins Bruveris, and Peter W Michor. Overview of the geometries of shape spaces and diffeomorphism groups. *Journal of Mathematical Imaging and Vision*, 50(1-2):60–97, 2014.

[BMTY05] M Faisal Beg, Michael I Miller, Alain Trouvé, and Laurent Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61(2):139–157, 2005.

[Boo97] Fred L Bookstein. *Morphometric tools for landmark data: geometry and biology*. Cambridge University Press, 1997.

[BRV12] Martins Bruveris, Laurent Risser, and François-Xavier Vialard. Mixture of kernels and iterated semidirect product of diffeomorphisms groups. *Multiscale Modeling & Simulation*, 10(4):1344–1368, 2012.

[Cha13] Nicolas Charon. *Analysis of geometric and functional shapes with extensions of currents: applications to registration and atlas estimation*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2013.

[CRM96] Gary E Christensen, Richard D Rabbitt, and Michael I Miller. Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on*, 5(10):1435–1447, 1996.

[CT13] Nicolas Charon and Alain Trouvé. The varifold representation of non-oriented shapes for diffeomorphic registration. *arXiv preprint arXiv:1304.6108*, 2013.

[DGM98] Paul Dupuis, Ulf Grenander, and Michael I Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of applied mathematics*, pages 587–600, 1998.

[DPC+14] Stanley Durrleman, Marcel Prastawa, Nicolas Charon, Julie R Korenberg, Sarang Joshi, Guido Gerig, and Alain Trouvé. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage*, 101:35–49, 2014.

[DPGJ11] Stanley Durrleman, Marcel Prastawa, Guido Gerig, and Sarang Joshi. Optimal data-driven sparse parameterization of diffeomorphisms for population analysis. In *Information Processing in Medical Imaging*, pages 123–134. Springer, 2011.

[FDY+14] Ana B Graciano Fouquier, Stanley Durrleman, Jérôme Yelnik, Sara Fernández-Vidal, and Eric Bardinet. Iconic-geometric nonlinear registration of a basal ganglia atlas for deep brain stimulation planning. In *2nd International MICCAI Workshop on Deep Brain Stimulation Methodological Challenges (DBSMC)*, 2014.

[Fle04] P Thomas Fletcher. *Statistical variability in nonlinear spaces: Application to shape analysis and DT-MRI*. PhD thesis, Citeseer, 2004.

[FLPJ04] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

[GCK12] Ulf Grenander, Yun-shyong Chow, and Daniel M Keenan. *Hands: A pattern theoretic study of biological shapes*, volume 2. Springer Science & Business Media, 2012.

[GCW+13] Pietro Gori, Olivier Colliot, Yulia Worbe, Linda Marrakchi-Kacem, Sophie Lecomte, Cyril Poupon, Andreas Hartmann, Nicholas Ayache, and Stanley Durrleman. Bayesian Atlas Estimation for the Variability Analysis of Shape Complexes. In *MICCAI 2013 : Medical Image Computing and Computer Assisted Intervention*, pages 267–274, Nagoya, Japan, September 2013.

[GK93] Ulf Grenander and Daniel Macrae Keenan. On the shape of plane images. *SIAM Journal on Applied Mathematics*, 53(4):1072–1094, 1993.

[Gla05] Joan Glaunes. *Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie numérique*. PhD thesis, 2005.

[Gre93] Ulf Grenander. *General pattern theory-A mathematical study of regular structures*. Clarendon Press, 1993.

[Gre96] Ulf Grenander. *Elements of pattern theory*. JHU Press, 1996.

[GSS07] Ulf Grenander, Anuj Srivastava, and Sanjay Saini. A pattern-theoretic characterization of biological growth. *Medical Imaging, IEEE Transactions on*, 26(5):648–659, 2007.

[GVM04] Joan Glaunès, Marc Vaillant, and Michael I Miller. Landmark matching via large deformation diffeomorphisms on the sphere. *Journal of mathematical imaging and vision*, 20(1-2):179–200, 2004.

[Jac13] Henry Jacobs. Symmetries in lddmm with higher order momentum distributions. *arXiv preprint arXiv:1306.3309*, 2013.

[JDJG04] Sarang Joshi, Brad Davis, Matthieu Jomier, and Guido Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage*, 23:S151–S160, 2004.

[KBCL09] David George Kendall, Dennis Barden, Thomas K Carne, and Huiling Le. *Shape and shape theory*, volume 500. John Wiley & Sons, 2009.

[Ken84] David G Kendall. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121, 1984.

[KT97] Aaron CW Kotcheff and Chris J Taylor. Automatic construction of eigenshape models by genetic algorithm. In *Information Processing in Medical Imaging*, pages 1–14. Springer, 1997.

[Kur76] S Kurcyusz. On the existence and nonexistence of lagrange multipliers in banach spaces. *Journal of Optimization Theory and Applications*, 20(1):81–110, 1976.

[KWB08] Ali R Khan, Lei Wang, and Mirza Faisal Beg. Freesurfer-initiated fully-automated subcortical brain segmentation in mri using large deformation diffeomorphic metric mapping. *Neuroimage*, 41(3):735–746, 2008.

[LDJ05] Peter Lorenzen, Brad C Davis, and Sarang Joshi. Unbiased atlas formation via large deformations metric mapping. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 411–418. Springer, 2005.

[MCAG93] Michael I Miller, Gary E Christensen, Yali Amit, and Ulf Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Sciences*, 90(24):11944–11948, 1993.

[MSBP15] Kristin McLeod, Maxime Sermesant, Philipp Beerbaum, and Xavier Pennec. Spatio-temporal tensor decomposition of a polyaffine motion model for a better analysis of pathological left ventricular dynamics. *IEEE transactions on medical imaging*, 34(7):1562–1575, 2015.

[MV07] Graham McNeill and Sethu Vijayakumar. Linear and nonlinear generative probabilistic class models for shape contours. In *Proceedings of the 24th international conference on Machine learning*, pages 617–624. ACM, 2007.

[MYT14] Michael I Miller, Laurent Younes, and Alain Trouvé. Diffeomorphometry and geodesic positioning systems for human anatomy. *Technology*, 2(01):36–43, 2014.

[PFY+99] Stephen M Pizer, Daniel S Fritsch, Paul A Yushkevich, Valen E Johnson, and Edward L Chaney. Segmentation, registration, and measurement of shape variation via image object shape. *IEEE transactions on medical imaging*, 18(10):851–865, 1999.

[PGV07] Nataliya Portman, Ulf Grenander, and Edward R Vrscay. New computational methods for the construction of "darcyan" biological coordinate systems. In *International Conference Image Analysis and Recognition*, pages 143–156. Springer, 2007.

[Por08] Nataliya Portman. Estimation of growth parameters for the drosophila wing disc from the sequence of its micrographs using the growth as random iterated diffeomorphisms model. 2008.

[Por10] Nataliya Portman. The modelling of biological growth: A pattern theoretic approach. 2010.

[PTC00] Stephen M Pizer, Andrew L Thall, and David T Chen. M-reps: A new object representation for graphics. In *ACM Transactions on Graphics*. Citeseer, 2000.

[PV11] Nataliya Portman and Edward R Vrscay. Existence and uniqueness of solutions to the grid macroscopic growth equation. *Applied Mathematics and Computation*, 217(21):8318–8327, 2011.

[RDSP15] Marc-Michel Rohé, Nicolas Duchateau, Maxime Sermesant, and Xavier Pennec. Combination of polyaffine transformations and supervised learning for the automatic diagnosis of lv infarct. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 190–198. Springer, 2015.

[RFS03] Daniel Rueckert, Alejandro F Frangi, and Julia A Schnabel. Automatic construction of 3-d statistical deformation models of the brain using nonrigid registration. *IEEE transactions on medical imaging*, 22(8):1014–1025, 2003.

[RVW+11] Laurent Risser, F Vialard, Robin Wolz, Maria Murgasova, Darryl D Holm, and Daniel Rueckert. Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping. *Medical Imaging, IEEE Transactions on*, 30(10):1746–1759, 2011.

[SLNP11] Stefan Sommer, François Lauze, Mads Nielsen, and Xavier Pennec. Kernel bundle epdiff: evolution equations for multi-scale diffeomorphic image registration. In *Scale Space and Variational Methods in Computer Vision*, pages 677–688. Springer, 2011.

[SLNP13] Stefan Sommer, François Lauze, Mads Nielsen, and Xavier Pennec. Sparse multi-scale diffeomorphic registration: the kernel bundle framework. *Journal of mathematical imaging and vision*, 46(3):292–308, 2013.

[SNDP13] Stefan Sommer, Mads Nielsen, Sune Darkner, and Xavier Pennec. Higher-order momentum distributions and locally affine lddmm registration. *SIAM Journal on Imaging Sciences*, 6(1):341–367, 2013.

[SNLP11] Stefan Sommer, Mads Nielsen, François Lauze, and Xavier Pennec. A multi-scale kernel bundle for lddmm: towards sparse deformation description across space and scales. In *Information Processing in Medical Imaging*, pages 624–635. Springer, 2011.

[SPR12] Christof Seiler, Xavier Pennec, and Mauricio Reyes. Capturing the multiscale anatomical shape variability with polyaffine transformation trees. *Medical image analysis*, 16(7):1371–1384, 2012.

[SSDG05] Anuj Srivastava, Sanjay Saini, Zhaohua Ding, and Ulf Grenander. Maximum-likelihood estimation of biological growth variables. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 107–118. Springer, 2005.

[T+42] Darcy Wentworth Thompson et al. On growth and form. *On growth and form.*, 1942.

[Thi98] J-P Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical image analysis*, 2(3):243–260, 1998.

[Tog98] Arthur W Toga. *Brain warping*. Academic press, 1998.

[Tro95] Alain Trouvé. An infinite dimensional group approach for physics based models in pattern recognition. *preprint*, 1995.

[Tro98] Alain Trouvé. Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision*, 28(3):213–221, 1998.

[VPPA09] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage*, 45(1):S61–S72, 2009.

[WBR+07] Lei Wang, Faisal Beg, Tilak Ratnanather, Can Ceritoglu, Laurent Younes, John C Morris, John G Csernansky, and Michael I Miller. Large deformation diffeomorphism and momentum based hippocampal shape discrimination in dementia of the alzheimer type. *IEEE transactions on medical imaging*, 26(4):462–470, 2007.

[YAM09] Laurent Younes, Felipe Arrate, and Michael I Miller. Evolutions equations in computational anatomy. *NeuroImage*, 45(1):S40–S50, 2009.

[You10] Laurent Younes. *Shapes and diffeomorphisms*, volume 171. Springer Science & Business Media, 2010.

[You12] Laurent Younes. Constrained diffeomorphic shape evolution. *Foundations of Computational Mathematics*, 12(3):295–325, 2012.

[YPJM01] Paul Yushkevich, Stephen M Pizer, Sarang Joshi, and James Stephen Marron. Intuitive, localized analysis of shape variability. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 402–408. Springer, 2001.