FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

PhD Thesis

Stefan Sommer

Anatomy in Curved Space

Non-linear Modeling of Deformation and Shape for Medical Imaging

The PhD School of Science

Datalogisk Institut (DIKU), Københavns Universitet
Department of Computer Science, University of Copenhagen

Academic advisors: Mads Nielsen and François Lauze

December 18th, 2011

Abstract

This thesis presents contributions in three topics in non-linear modeling of deformation and shape with applications to medical imaging:

To obtain *compact* description of deformation while keeping the *capacity* of the deformation model, we present two results on registration and deformation modeling. We introduce the *kernel bundle* framework which extends the LDDMM framework to represent deformation at multiple scales while preserving much of the mathematical structure underlying the original framework. We explore the mathematical properties of the multi-scale construction and derive evolution equations with the bundle. The kernel bundle in particular allows application of sparse priors *across scales*, and we use this property to obtain compact representations while keeping the capacity of the deformation model and its ability to generalize to test data. The method is evaluated on annotated lung CT images and a fast GPU optimized registration algorithm is developed and tested.

In addition, sparse deformation representation with LDDMM is restricted by representing only translational motion. We introduce *higher order kernels* in the framework to allow modeling of locally affine deformation. The higher order kernels fit naturally into the mathematical construction of the LDDMM, and this enables us to derive evolution equations and a matching algorithm using first order information. We show how the increased description capacity allows registration with very few parameters, and we apply the kernels to register MR scans of patients suffering from Alzheimer's disease.

Performing statistics in non-linear spaces, in particular on Riemannian manifolds, requires computational tools to compute directions, distances, and projections. We present algorithms for computing the differential of the Exponential map and second order derivatives on Riemannian manifolds leading to an algorithm for computing *exact Principal Geodesic Analysis*, a generalization of PCA to manifolds which is exact as it does not use the common tangent space linearization. We evaluate the results obtained with the exact algorithm against the standard PGA method and provide new insight into when modelling non-linearity is beneficial.

To reduce annotation variation in point based models, we introduce the *bicycle chain shape model* for 2D-shape representation. The model imposes constraints on the pairwise point distances which leads to a non-linear shape space when keeping the constraints consistently enforced. We develop tools for performing statistics on the embedded Riemannian manifold comprising the model, and we apply the method to represent and perform statistics on a dataset of human vertebrae X-rays.

Resumé

Denne PhD-afhandling omhandler ikke-lineær modellering af deformation og former med anvendelser i medicinsk billedbehandling. Nye resultater præsenteres indenfor tre hovedområder:

For at kunne beskrive deformation *kompakt* samtidig med at deformationsmodellens beskrivelsesevne bibeholdes, præsenterer vi to resultater til deformationsbeskrivelse og registrering. Vi introducerer *kerne-bundt* (kernel bundle) udvidelsen af LDDMM, der giver muligheden for at repræsentere deformation på flere skalaer samtidig med at store dele af den matematiske struktur, der ligger til grund for den oprindelige model, bibeholdes. Vi undersøger de matematiske egenskaber med multi-skala konstruktionen og udleder flow-ligninger. Kerne-bundtet tillader specielt brug af sparse priors på de enkelte skalaer, og vi benytter denne egenskab til at opnå kompakte repræsentationer, samtidig med kapaciteten af deformationsmodellen og dens evne til at generalisere til testdata bibeholdes. Metoden evalueres på annoterede lunge-CT scanninger, og en hurtig registreringsalgoritme til grafikprocessorer bliver udviklet og testet.

Kompakt deformationsbeskrivelse med LDDMM er yderligere begrænset ved kun at repræsentere translatering. Vi introducerer *højereordenskerner* (higher order kernels) i modellen for at kunne modellere lokalt affine deformationer. Højereordenskerner passer på en naturlig måde i den matematiske konstruktion bag LDDMM, og ved at udnytte dette udleder vi flow-ligninger og en registreringsalgoritme, der inkluderer førsteordens information. Vi viser hvordan den øgede kapacitet af modellen tillader registrering med meget få parametre, og vi benytter kernerne til at registrere MR scanninger af patienter med Alzheimers sygdom.

Statistik i ikke-lineære rum, specielt på Riemannske mangfoldigheder, kræver algoritmer til at udregne retninger, afstande og projektioner. Vi udvikler algoritmer til beregning af differentialet af eksponentialafbildningen og andenordensafledte på Riemannske mangfoldigheder. Dette fører til en algoritme til at beregne *eksakt PGA*, som er en udvidelse af PCA til mangfoldigheder og som er eksakt, idet den ikke bruger tangentrumslinearisering. Vi sammenligner resultater beregnet med den eksakte algoritme med resultater fra den gængse PGA metode. Dette giver ny indsigt i hvornår det kan betale sig at modellere ikke-linearitet.

For at reducere variation ved manuel annotering til punkt-baserede 2D-kurvemodeller, introducerer vi *cykelkæde* form-modellen (bicycle chain shape model). Modellen indfører krav til de parvise afstande mellem punkter, hvilket fører til et ikke-lineært formrum, når kravene indføres konsistent. Vi udvikler algoritmer til at udføre statistik på den indlejrede mangfoldighed, som udgør modellen, og vi benytter metoden til at repræsentere og udføre statistik på et datasæt med røntgenbilleder af vertebrae fra mennesker.

Contents

Introduction

Vertebrae may fracture as a result of osteoporosis; Alzheimer's disease can lead to brain atrophy; a reduction in hippocampal size has been observed with patients suffering from depression or schizophrenia. These examples and a number of additional diseases cause anatomical changes in the human body that can be observed using medical imaging techniques and that can be explored using computer algorithms.

The ability to perform statistics is fundamental for empirical research, and the overall goal of the work presented in this thesis is to allow statistical exploration of imaging data showing changes in human anatomy. For example, a precise characterization of the shape change of the hippocampus may lead to improved understanding of schizophrenia; if wear in certain areas of vertebrae indicates osteoporosis, early diagnosis and assessment of the effect of treatments may be possible; knowledge of Alzheimer's disease may be gained by a more detailed view of the brain atrophy.

Based on examples as the above, we wish to allow statistics to be performed on the shape of organs, and, to pursue this goal, this thesis concerns modeling and statistical methods on deformation and shape. Statistics on geometric objects is far from as well established as performing statistics on numbers, and quantification of change requires models, metrics, non-linear statistical methods, computational representations, and numerical algorithms. In addition, domain specific knowledge is often needed, and different objects to be studied require different models and algorithms. The work presented here constitutes steps for specific methods seeking to allow statistics on anatomical changes, and the papers contribute to components in the entire pipeline from model to algorithm.

Summary of Contributions

The results presented add to the state-of-the-art by the following contributions:

Multi-Scale Deformation Modeling
: The LDDMM registration framework has many important mathematical and modeling properties but sparse deformation description with LDDMM is limited in representing deformation at only one scale. We introduce the *kernel bundle* framework for multi-scale deformation representation and explore the underlying mathematical structure. The kernel bundle in particular allows application of sparse priors *across scales*, and we use this property to obtain compact representations while keeping the capacity of the deformation model and its ability to generalize to test data. The method is evaluated on annotated lung CT scans, and we present a GPU optimized registration algorithm.

Higher Order Kernels for LDDMM
: Finite dimensional representations of LDDMM do not directly support all affine motions; only a combination of translations can approximate non-translational deformations. This limitation restricts the ability to represent deformation sparsely. We show how *higher order kernels* through the partial derivative reproducing property fit naturally into the LDDMM framework and how the new kernels allow compact representation of locally affine deformation such as local rotation and dilation. Through experiments, we demonstrate how the increased descrip-

tion capacity allows registration with very few parameters, and we apply the kernels to register MR scans of patients suffering from Alzheimer's disease.

Numerical Algorithms and Non-linear Statistics
Performing statistics in non-linear spaces, in particular on Riemannian manifolds, requires computational tools to compute directions, distances, and projections. We present algorithms for computing the differential of the Exponential map and second order derivatives on Riemannian manifolds. These results lead to an algorithm for computing *exact Principal Geodesic Analysis*, a generalization of PCA to manifolds which is exact as it does not use the common tangent space linearization. We evaluate the results obtained with the exact algorithm against the standard PGA method and provide new insight into when modelling non-linearity is beneficial.

2D-Shape Modeling
To reduce annotation variation in point based models, we introduce the *bicycle chain shape model* for 2D-shape representation. The model imposes constraints on the pairwise point distances which leads to a non-linear shape space when keeping the constraints consistently enforced. We develop tools for performing statistics on the embedded Riemannian manifold comprising the model, and we apply the method to represent and perform statistics on a dataset of human vertebrae X-rays.

Structure of the Thesis

The main body of this thesis consists of papers presenting the research in which I have been involved during my PhD studies. The present introduction will be followed by a brief discussion of the relation between the presented papers and the current state of the research fields that the papers concern. Following this, each of the papers are included as published or submitted for review. Only page numbers have been converted in order to fit the numbering of the thesis. The papers are ordered according to the topic they concern: registration and deformation modeling; non-linear statistics and algorithms; and 2D shape modeling. The thesis will end with a short summary, concluding remarks, and outlook.

Papers

The thesis comprises the six papers listed below all of which I am the first author. Three of the papers have been peer-reviewed, presented at conferences and published in conference proceedings. The remaining three papers are submitted for journals and currently under review. Paper #1, "Sparse Multi-Scale Diffeomorphic Registration: the Kernel Bundle Framework", is the result of an invitation to submit to the Scale-Space and Variational Methods special issue in the Journal on Mathematical Imaging and Vision, and it combines and extends the three conference papers [1, 2, 3]. These three conference papers are not included in the thesis since the material presented in the papers is largely covered in Paper #1.

Further, I have contributed to three additional papers where I am not the first author. The six papers that are not part of the thesis are listed below as not included

papers. Combined, the six papers included and the six papers not included constitute the result of my 3 years of PhD studies 2009-2011 at the Image Group, Department of Computer Science, University of Copenhagen.

Included Papers:

Sparse Multi-Scale Diffeomorphic Registration: the Kernel Bundle Framework

Stefan Sommer, François Lauze, Mads Nielsen, and Xavier Pennec. Paper invited for submission to the Scale-Space and Variational Methods special issue in the Journal on Mathematical Imaging and Vision (JMIV). Submitted to JMIV, December 2011.

Higher Order Kernels and Locally Affine LDDMM Registration

Stefan Sommer, Mads Nielsen, Sune Darkner, and Xavier Pennec. Submitted to SIAM Journal on Imaging Sciences, December 2011.

Accelerating Multi-Scale Flows for LDDKBM Diffeomorphic Registration

Stefan Sommer. GPUCV workshop at ICCV 2011, Barcelona, Spain, 2011.

The Differential of the Exponential Map, Jacobi Fields and Exact Principal Geodesic Analysis

Stefan Sommer, François Lauze, and Mads Nielsen. Submitted to Foundations of Computational Mathematics, May 2011.

Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations

Stefan Sommer, François Lauze, Søren Hauberg, and Mads Nielsen. European Conference on Computer Vision (ECCV) 2010, Heraklion, Greece, 2010.

Bicycle Chain Shape Models

Stefan Sommer, Aditya Tatu, Chen Chen, Dan R. Jørgensen, Marleen de Bruijne, Marco Loog, Mads Nielsen, and François Lauze. Mathematical Methods in Biomedical Image Analysis (MMBIA) at CVPR 2009, Miami Beach, Florida, 2009.

Papers Not Included:

A Multi-Scale Kernel Bundle for LDDMM: Towards Sparse Deformation Description Across Space and Scales

Stefan Sommer, Mads Nielsen, François Lauze, and Xavier Pennec. Information Processing in Medical Imaging (IPMI) 2011, Irsee, Germany, 2011.

Kernel Bundle EPDiff: Evolution Equations for Multi-Scale Diffeomorphic Image Registration

Stefan Sommer, François Lauze, Mads Nielsen, and Xavier Pennec. Scale Space and Variational Methods in Computer Vision (SSVM) 2011, Ein-Gedi, Israel, 2011.

Sparsity and Scale: Compact Representations of Deformation for Diffeomorphic Registration

Stefan Sommer, Mads Nielsen, and Xavier Pennec. Mathematical Methods in Biomedical Image Analysis (MMBIA) at WACV 2012, Breckenridge, Colorado, 2012.

Natural Metrics and Least-Committed Priors for Articulated Tracking

Søren Hauberg, Stefan Sommer, and Kim Steenstrup Pedersen. Accepted for publication in Elsevier Journal on Image and Vision Computing, submitted March 2011.

Gaussian-like Spatial Priors for Articulated Tracking

Søren Hauberg, Stefan Sommer, and Kim Steenstrup Pedersen. European Conference on Computer Vision (ECCV) 2010, Heraklion, Greece, 2010.

On Restricting Planar Curve Evolution to Finite Dimensional Implicit Subspaces with Non-Euclidean Metric

Aditya Tatu, François Lauze, Stefan Sommer and Mads Nielsen. Journal of Mathematical Imaging and Vision, Springer 2010.

Acknowledgments

I wish to thank in particular my supervisors, Professor Mads Nielsen and Assistant Professor François Lauze both at the University of Copenhagen, for guiding me through the last three exciting years. The major parts of the research presented in this thesis have been produced as joint work with Mads and François. Our strategic meetings have been invaluable in guiding my research and opening my eyes to the ways of the academic world. Last but not of less importance, I have enjoyed their ways of motivating students including dinners in Nice and Antibes and open-air rides through the Rockies.

My stay at the Asclepios research group at INRIA Sophi-Antipolis during the winter 2010-2011 has been particularly inspiring. I wish to thank Professor Xavier Pennec and the entire group for hosting me. It is hard to think of a better source of inspiration than a prolonged discussion on math and geometry with Xavier; the work on deformation modeling presented here comes mainly as results of these talks. Further, I had the opportunity to attend the "Geometry for Anatomy" workshop in Banff, Alberta in the fall of 2011. Immense inspiration and the ideas for two papers arose from this workshop, and I would like to thank the organizers, Ghassan Hamarneh, Stephen Pizer, and Hao Zhang, for letting me participate.

Last, I would like to thank Søren Hauberg, Sune Darkner, and the rest of the Image Group at the University of Copenhagen for being great colleagues, great inspirators, and for making a great amount of fun.

Stefan Sommer
Frederiksberg, December 2011.

1.

Overview

The purpose of this chapter is to lay out the context of the papers comprising the thesis and discuss the problems they address. We will start with high level comments on non-linear modeling and the difficulties arising when diverting from linearity. Next, we will discuss the three main topics of the thesis: registration and deformation modeling; non-linear statistics and algorithms; and 2D shape modeling. These parts will concern the purposes and results of the papers and give overall comments on the current state of the fields. The "related work" sections in the papers provide short reviews of the fields, and the intention is not to repeat those reviews here.

1 Non-linear Modeling

We wish to model, measure, and do statistics on organs with concrete examples being bones, lungs, and the human brain. It is often hard to find accurate linear models for the geometry of organs, and it turns out that, in many cases, representations of the geometry as points in Euclidean space fail to be adequate for statistical purposes. For example, in Section 4 we consider outlines of human vertebrae, and a vertebra may be represented by N points in $\mathbb{R}^2$ lying on the outline of a lateral X-Ray of the spine, confer Figure 1.5(a). The arithmetic mean in $\mathbb{R}^{2N}$ between a collection of such points may however not look like a vertebra at all.

Non-linear modeling appears in the search for meaningful and theoretically well-founded models that are accurate and compact. Prior knowledge can be used to restrict the modeling space to objects which are actually *meaningful* to represent. For example, we may define a subset of $\mathbb{R}^{2N}$ that we find represents realistic vertebrae and restrict the model to this subspace. Such a subspace will most likely be non-linear. Furthermore, it is often hard to define e.g. distances and distributions directly in the global modeling space. Instead, modeling can be performed infinitesimally, and the infinitesimal constructions can be integrated to provide global structures. The LDDMM deformation model that we discuss in Section 2 provides an example of how this approach can lead to theoretically *well-founded* models. Both restriction of the modeling space and infinitesimal approaches can reduce the dimensionality of the model leading to increased *compactness* without reducing the *accuracy* of the representation. Correspondingly, the increased compactness for a given accuracy can lead to increased accuracy when comparing against a linear representation with equivalent dimensionality.

Non-linearity does, however, come at a price. Figure 1.1 shows common steps

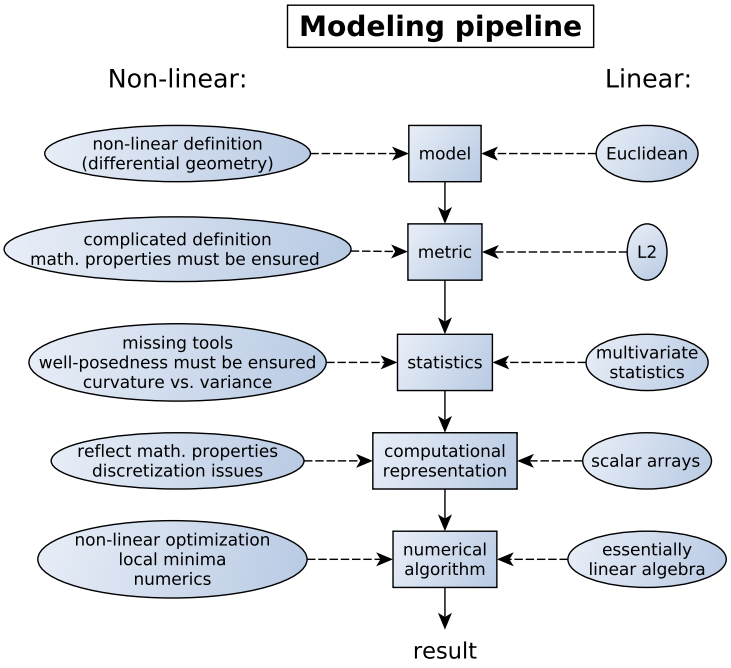


Figure 1.1: Sketch of a modeling pipeline with elements needed for each step of non-linear modeling (left) and linear modeling (right). To summarize: for linear models, most of the required elements can be taken straight off the shelf and used; for non-linear models, every step is complicated.

needed to go from problem definition to working tools that allow analysis in non-linear spaces. For each element of this pipeline, research in mathematics, statistics, and computer science has already developed solid *linear* tools. In contrast, for non-linear modeling, many of the elements comprise active research areas. Non-linear modeling is hard because the mathematics of non-linearity is involved; because the added freedom requires careful selection of metrics; because statistics in non-linear spaces is largely undefined; because non-linear constraints are hard to represent in computational representations; and because actual algorithms may drown in numerical errors, local minima, and prohibitive need for computing resources. The left column of Figure 1.1 also shows the need for interdisciplinarity when working with non-linear modeling. Geometry, statistics, numerical analysis, and domain specific knowledge all constitute important parts.

The work constituting this thesis spans from model to implemented algorithm touching several parts of the pipeline show in Figure 1.1. The papers concern data of varying nature and address different non-linear problems; they are therefore best considered distinct but related contributions to non-linear modeling.

One important general observation has reappeared when performing the research presented in the papers: *curvature is relative to spread*, or, with an equation which should be interpreted informally,

$$\text{non-linearity} = \text{curvature} \times \text{spread} \ .$$

If either the curvature of the modeling space is low or the data is very localized, there is a good chance that a linear model *in practice* will perform just as well as a more precise but complicated non-linear equivalent. On the other hand, if both curvature and spread is high, even the presently known non-linear statistical models become problematic. As we will see, modeling vertebrae using the bicycle chain model described in Paper #6 and Section 4 constitutes an example of the former case; applying the PGA procedure described in Section 3 on human motion data is arguably an example of the latter case.

2 Registration and Deformation Modeling

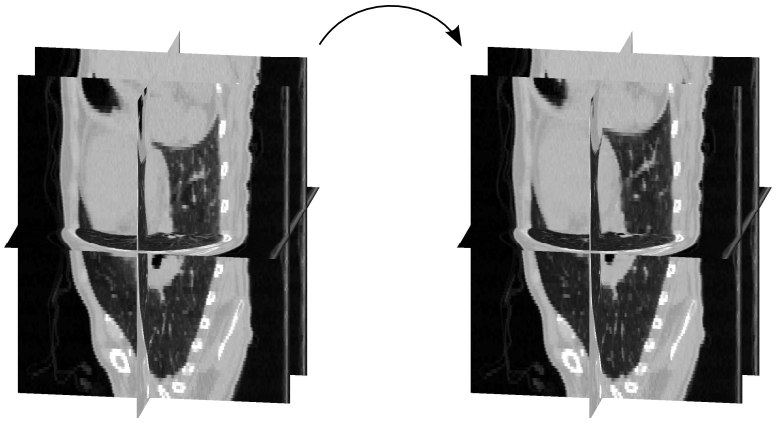
Finding correspondences between geometric objects, organs in particular, is often of interest: if we acquire CT scans of lungs, there will be a natural variation caused by the respiratory process. We can remove this variation by finding correspondences between points in the scans. Similarly, if we acquire baseline and follow-up MR scans of the brain of a patient suffering from Alzheimer's disease, we can use correspondences between the scans to see possible progressing atrophy.

Registering geometric objects, images in particular, has been the subject of a huge amount of work over the last decades. In this thesis, we focus on the deformable template view and the LDDMM framework. The LDDMM framework provides a complete deformation model with a Lie group structure on spaces of deformations and Riemannian metrics. The metrics measure the cost of infinitesimal deformations, and large deformations are generated by integrating infinitesimal motions. The benefits are the ability to measure distances between points, lines, surfaces, distributions, and images; explicit control of the smoothness enforced in the registration; and possibility of performing well-founded statistics on the registration results. For the latter property, the explicit mathematical foundation of LDDMM promises statistics measuring real patterns in the data instead of possible artifacts of the registration algorithm.

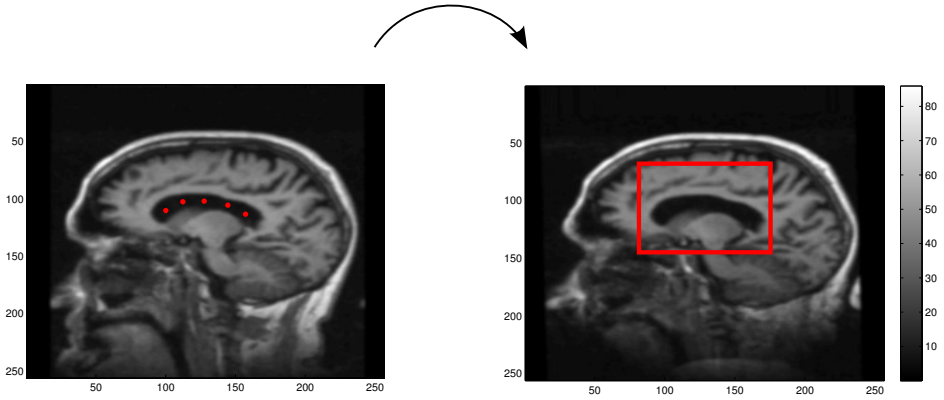
Paper #1 and Paper #2 provide short introductions to the key concepts in LDDMM: the Lie group formulation, the tangent space with reproducible kernel Hilbert space (RKHS) structure, the metric, and the EPDiff evolution equations. We refer to the papers for description of these concepts, and provide here the main outline for our approach to LDDMM.

The deformation model in LDDMM is *infinitely* dimensional, and most implementations reflect this by seeking discretizations as fine as possible. The deformation we wish to obtain when registering images may be of significantly lower dimensionality than provided by dense discretizations, and the dimensionality may vary over different spatial locations in the images. For example, large-scale deformation may be needed when registering brains from different subjects while e.g. atrophy may occur at small scales during the progression of Alzheimer's disease.

We wish to depart from dense discretization by looking for *sparser* representations of deformation that contain *interpretable* information. We denote the basis elements of such representations *deformation atoms*. In LDDMM, the fundamental notion of kernel parametrizes infinitesimal deformations, and kernels at different spatial locations are used as deformation atoms. Since the kernel has a fixed scale and since it only encodes translation information, sparse representations should ideally be accompanied by increased *capacity* of the deformation description provided by each atom. Our work on deformation modeling revolves around this program of allowing sparsity through



(a) Registering lungs in inhale and exhale phases; 3D CT. In Paper #1, Paper #3, and [1, 3], we use annotated scans for registration with our *kernel bundle* multi-scale registration framework.



(b) Registering baseline and follow up scans; 2D MRI of an Alzheimer's patient. The baseline image (left) contains deformation atoms in the form of *first order kernels*, confer Paper #2. The ventricle area to be registered is marked in the follow up image (right).

Figure 1.2: Registration, finding correspondences between geometric objects, is important for a multitude of applications. CT and MR data from [4, 5].

increased capacity with the introduction of the multi-scale *kernel bundle* framework and by developing *higher order kernels*. The former has the effect of varying the spatial range of each deformation atom; the latter has the effect of locally increasing the description capacity of the atoms.

In the conference paper [1], we present simple examples illustrating why sparse description of deformation occurring at different scales at different spatial locations will require multiple scales in the deformation model. Based on these observations, we introduce the *kernel bundle* or LDDKBM extension of LDDMM. The aim is to allow multi-scale representation while keeping much of the mathematical structure of LDDMM intact. In [2], we continue exploring the mathematical properties of the multi-scale model by deriving the KB-EPDiff evolution equations.

Including multiple scales in LDDMM has also been treated by Risser et al. [6, 7]

and Bruveris et al. [8]. While the approach of Bruveris et al. corresponds to the kernel bundle method with two scales, the approach of Risser et al. is different in not representing the momentum field at different scales. In contrast, the kernel bundle is designed specifically to represent deformation individually at different scales. This in particular allows the momentum field to vanish at some scales while being non-zero at others at the same spatial locations, a property we denote *sparsity across scales*. It has been shown [9] that optimal deformations with the original kernel bundle formulation will coincide with optimal deformations with the approach of Risser et al. Thus, though the kernel bundle is able to represent sparsity across scales, the cross-scale sparsity will not occur without adding more information to the system. In [3], we pursue this by applying sparse priors to the individual scales in kernel bundle framework, and we demonstrate in the experiments section that sparsity across scales is indeed achieved. Scale information may also arise from the data term, and we are currently continuing the multi-scale program by searching for the right way to combine both multi-scale representation, multi-scale regularization and prior information, and multiple scales present in the data.

We were invited to extend the work presented in [2] for the Scale-Space and Variational Methods special issue in the Journal on Mathematical Imaging and Vision. This resulted in Paper #1 which combines and extends the three conference papers [1, 2, 3] to one account of the kernel bundle framework.

Continuing with the goal of obtaining sparser representations through increased description capacity, we develop *higher order kernels* for LDDMM in Paper #2. With the common LDDMM representations and even with multiple scales, sparse representations are limited to represent *translational* movements at each deformation atom. Contractions, expansions, rotations and other affine transformations must be approximated using several atoms. This limits the range of deformation that sparse representations can describe. With higher order kernels, we extend each deformation atom to locally represent first order information and thus describe locally affine transformations. We illustrate the application of the higher order kernels by registering images using low numbers of deformation atoms, and we use the method to register MR scans of patients with progressing atrophy caused by Alzheimer's disease.

Finally, in Paper #3, we take a more implementation specific focus and develop a GPU implementation of the kernel bundle algorithm. The resulting two-orders of magnitude speedup from a single-threaded CPU implementation shows that including multiple-scales does not rule out fast computation of the registration results with large amounts of landmarks. In the paper, the kernel bundle framework is referred to as LDDKBM.

3 Non-linear Statistics and Algorithms

When using non-linear models for geometric objects, we are confronted with the problem of performing non-linear statistical analysis. In non-linear statistics, we cannot directly use the inner product that is present in Euclidean spaces; with infinitesimal models, this is reflected in the contrast between the global nature of the inner product and the local nature of infinitesimal variations. In addition, we must often take care of infinite dimensionality and artifacts such as non-uniqueness or non-existence of means. In this thesis, we apply non-linear statistical methods to shape analysis

and human motion models but the need for statistical methods outside the Euclidean settings is not limited to these examples, and much work has gone into generalizing well-known Euclidean concepts to different non-linear spaces. The field is however far from completely explored.

Almost no concept from Euclidean statistics has a straightforward generalization to non-linear spaces. Here, we discuss issues in generalizing three of the most important concepts before relating them to the two papers in this thesis concerning non-linear statistics. Confer [10] for a recent review of the field.

Means

The common arithmetic mean $m = \frac{1}{N}\sum_{i=1}^{N} x_i$ has for Euclidean data the property of minimizing the variance between the data points $x_i \in \mathbb{R}^d$ and m, i.e.

$$m = \text{argmin}_{x\in\mathbb{R}^d} \sum_{i=1}^{N} \|x_i - x\|^2 \ . \tag{1.1}$$

By definition, the arithmetic mean is unique and it always exists. In non-linear spaces, addition is most often not well-defined; even if addition is defined, it will not be compatible with the metric structure and the arithmetic mean will not provide a distance minimizing property (1.1).

The most frequently used non-linear equivalent of the arithmetic mean is the Fréchet [11] mean that on a metric space M with distance $d_M(\cdot,\cdot)$ generalizes (1.1) to

$$m = \text{argmin}_{x\in M} \sum_{i=1}^{N} d_M(x_i, x)^2 \ . \tag{1.2}$$

Note that m here is a set of global minimizers of the variance. Looking only at local minimizers of (1.2), Karcher [12] shows that existence and uniqueness is ensured for sufficiently local data when M is a Riemannian manifold. For slightly less local data, uniqueness may fail even in simple cases. Local minimizers of (1.2) will be used when discussing the PGA generalization of PCA below. In the experiments presented in the papers on non-linear statistics, we observed that, in rough terms, either the data is sufficiently localized for the situation to be essentially linear or the data is non-localized making analysis centered around the mean problematic, confer also Figure 1.3.

Gaussian Distributions

Gaussian distributions can be generalized to manifolds in several ways: by using the Laplace-Beltrami operator to obtain solutions to the heat equation or Brownian motion; by projecting Euclidean Gaussian distributions in the tangent space of the mean to the manifold using the Exponential map; by projecting embedding space Gaussians to the manifold; and by maximizing global entropy [10]. An important point is that we do not have the Euclidean convenience of one distribution that satisfies all of these properties at once.

Principal Component Analysis

The Euclidean Principal Component Analysis (PCA) procedure aims to find low dimensional subspaces capturing the variance of a dataset. This can equivalently

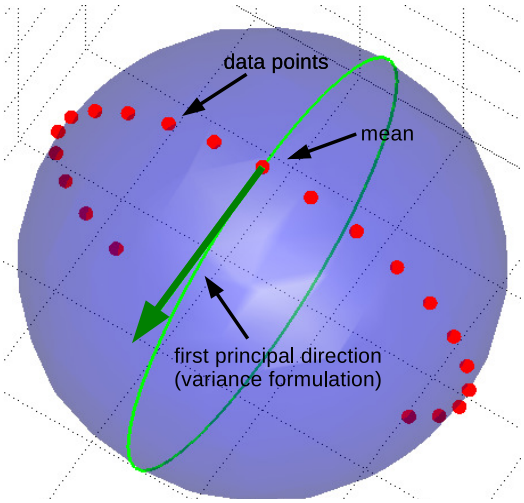
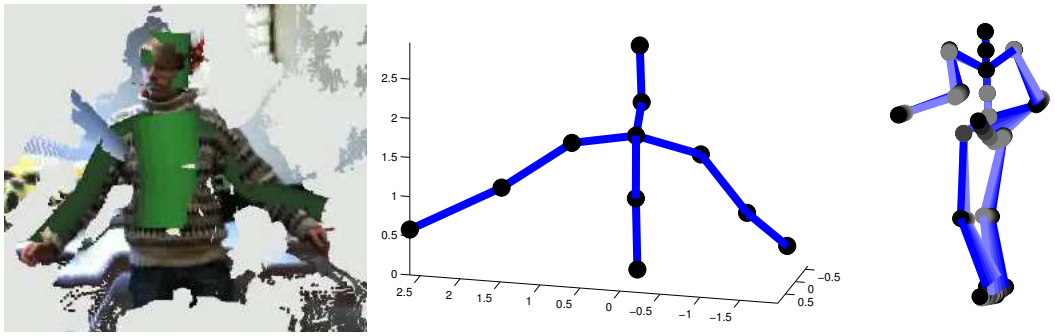


Figure 1.3: Even when the mean (1.2) is unique, centering analysis can be problematic for non-local data. Here, data points on a sphere lie on a great circle passing through the mean. Nevertheless and non-intuitively, exact PGA computes the first principal direction to be orthogonal to the great circle. Observing such phenomena is possible using the algorithms developed in Paper #4.

be formulated either as minimizing residual errors or maximizing captured variance.

Principal Geodesic Analysis (PGA, [13, 14]) and Geodesic PCA [15, 16] both provide abstractions of PCA to manifolds. PGA centers the analysis to the tangent space of local minimizers of (1.2) and aims for maximizing variance in linear subspaces of the tangent space. Projections are defined using the manifold distance but in order to make the computations feasible, orthogonal projections in the tangent space are used as approximations. Geodesic PCA minimizes residual errors and uses the fact that the minimizing geodesics need not pass the means in non-linear spaces. Therefore, the analysis is not centered like PGA. PGA was first applied to medial representations, confer Section 4.

In two papers, we consider algorithms and statistics on Riemannian manifolds: in Paper #4, an algorithm for computing the derivative of the Exponential map on manifolds is developed with the application of computing PGA without approximating projections with orthogonal projections in the tangent space. This procedure is denoted *exact PGA*. Approximating projections with orthogonal projections in the tangent space corresponds to a *linearization* of the manifold, and, from an algorithmic point of view, the exact PGA procedure shows how essential notions in non-linear statistics can be computed without such linearization. From a modeling point of view, it is however an interesting observation that for fairly localized data, the difference between PGA and exact PGA can be negligible. In order to test this further, we compare in Paper #5 the exact and non-exact algorithms on a dataset of vertebrae outlines represented using the bicycle chain shape model of Paper #6 and on a non-linear model of human pose. The results emphasize that, informally, non-linearity equals curvature times spread: for the



(a) Camera output and superimposed tracking result.

(b) Tracked pose with end-effectors (dots).

(c) Eigenmode of PGA; full-body poses.

Figure 1.4: In Paper #5, we compare approximated and exact PGA on a non-linear spatial model of human poses. In addition, in [17, 18], we use different Gaussian-like distributions on the pose manifold to drive a particle-filter based tracking algorithm.

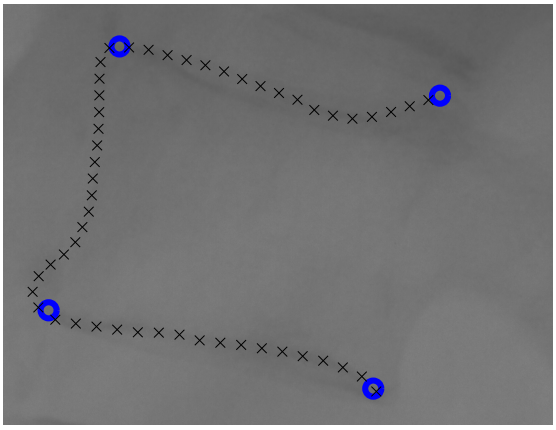
vertebra dataset, a linear model would be sufficient; for the human poses that show great variation, a linear model would result in a poor approximation.

In our papers [17, 18], two of the papers that are not included in the thesis, we continue exploring the non-linearity of the human pose representation by using several of the different non-linear generalizations of Gaussian distributions to estimate human poses from video sequences using particle filtering system. In [17], tangent space and embedding space Gaussians projected to the manifold are used, and, in [18], we develop a numerical scheme for simulating manifold valued Brownian motion and use that in the particle filter.

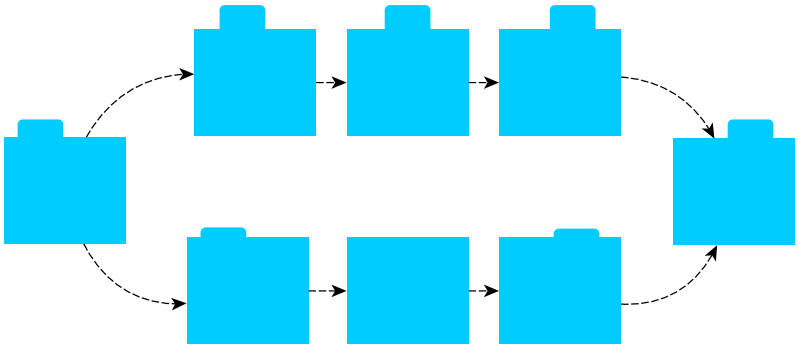
4 2D Shape Modeling

A large class of images of organs are two dimensional and this makes 2D modeling of curves and shapes in the plane important. Outlines of human vertebrae will here serve as the main example of 2D shapes, confer Figure 1.5(a).

2D shapes can be modeled using only the curve surrounding the shape or using the entire interior of the shape. Approaches to the latter include modeling deformation of the domain containing the shape or using set distances such as the Hausdorff distance. In this section, we will focus on curve models; in contrast, the LDDMM framework discussed in Section 2 models domain deformation. Curve models can further be categorized in approaches modeling correspondence and approaches seeking invariance of point correspondences. Point distribution models (PDM) are examples from the former category, and parametrization invariant models are members of the latter category. Besides the fundamental question of how the distance between two shapes should be defined, confer Figure 1.5(b), the debate between proponents of each approach touches aspects such as the problem of actually determining correspondences, the implication of noise in the measurements on invariant models, and analytical issues when removing



(a) Vertebra outline represented using the bicycle chain shape model (Paper #6). Notice the constant pairwise distances between the points resembling the constant distances between the pins in a bicycle chain.



(b) Shape modeling concerns in particular the definition of distances in the shape space. For example, with the 2D shapes in the picture, the distance between the left and right shapes can be interpreted as the work required to (top) *move* the bulb from left to right or to (bottom) *remove* and *recreate* the bulb.

Figure 1.5: 2D shape modeling: the bicycle chain shape model and the definition of distances.

the reparametrization group.[1]

Before discussing properties of the *bicycle chain shape model* which we propose in Paper #6, we discuss four different approaches to 2D shape modeling. The bicycle chain model refers directly to the first two models, the PDM and L^2-models. Approaches to performing statistics in the non-linear shape spaces of the elastic and medial models inspired the algorithms used in the bicycle chain model and the PGA algorithms discussed in Section 3. A thorough review of 2D shape models can be found in the monographs [19, 20, 21].

[1] These issues were primary discussion topics at the "Geometry for Anatomy" workshop in Banff, Alberta, Canada, August 2011.

Point Distribution Models (PDM)

Perhaps the most classical approach to shape modeling is to represent the curve with a number of sample points or *landmarks*. In the Kendall shape space [22, 23], distances between curves represented by landmarks are measured using the Euclidean distance between the landmarks but modulo the effect of rotations, translations, and scaling of the points. The Kendall shape spaces are non-linear but linear models such as the active shape models [24] have been used with great success. Importantly, the linearity allows the use of statistical tools such as PCA for dimensionality reduction. The use of landmarks makes correspondences an integral part of these methods. Correspondences can be found with e.g. MDL approaches [25].

L^2-norm on Immersed Curves

In set $\mathrm{Imm}(S^1, \mathbb{R}^2)$ of closed curves immersed in $\mathbb{R}^2$ [26, 27], a variation v of a curve $c \in \mathrm{Imm}(S^1, \mathbb{R}^2)$ gives a vector at each point of the curve. Elements of the set $C^\infty(S^1, \mathbb{R}^2)$ of such variations can be considered tangent vectors on the manifold of immersed curves, and a natural choice of metric on this space is

$$\langle v_1, v_2 \rangle_c = \int_{S^1} \langle v_1(t), v_2(t) \rangle_{\mathbb{R}^2} \, \|\dot{c}(t)\|_{\mathbb{R}^2} dt$$

for variations v_1, v_2 of the curve c.

The parametrization of curves in $\mathrm{Imm}(S^1, \mathbb{R}^2)$ is often not considered a part of the geometry of the curve. Since the above inner product is invariant of the choice of parametrization, it induces an inner product on the quotient $\mathrm{Imm}(S^1, \mathbb{R}^2) \setminus \mathrm{Diff}(S^1)$.

This metric is less natural than it seems [28]: the distance between any two curves vanishes as a result of the parametrization invariance and infinite dimensionality of the spaces. Various other choices of metrics than the L^2-norm have been proposed to prevent this degeneracy of the metric. These include penalizing the length of the curves [29], penalizing the curvature of the curves [26], and Sobolev-type metrics [30, 27]. For the latter approach, derivatives of the tangent vectors is included in metrics on the form

$$\langle v_1, v_2 \rangle_c = \int_{S^1} \sum_\alpha \langle D_t^\alpha v_1(t), D_t^\alpha v_2(t) \rangle_{\mathbb{R}^2} \, \|\dot{c}(t)\|_{\mathbb{R}^2} dt \ .$$

Elastic Models

Variations of curves can also be formulated in terms of variations in their angle function and parametrization speed. The *elastic* curve metric [31] is defined through an inner product on such variations. Various representations of curves suitable for the elastic metric have been proposed [32, 33] each resulting in different expressions for the metric. They all center around representing the velocity vector $c'(t)$ of the differentiable curve c by a pair of functions (φ, θ) such that

$$c'(t) = \varphi(t) e^{i\theta(t)}$$

in complex coordinates, and φ can for example be given in logarithmic form or as the squared length of the curve derivative. Penalizing variations in φ corresponds

to increasing the tension of the curve, and penalizing variations of θ increases the rigidity. The review paper [27] provides an overview of the relation between the elastic metric, the L^2- and Sobolev metrics.

Medial Representations

A different approach to shape modeling is based on the *medial axis* of Blum [34]. The medial axis of a 2D or 3D shape is the subset of the points enclosed by the shape boundary that have more than one closest point on the boundary. By describing the position of each such point together with the distance to the boundary and vectors pointing to each of the closest points, one obtain the *medial representation* of the shape. The m-rep representation has been very successful in modeling shape variation in many medical applications [35, 36, 13, 14], and the PGA procedure discussed in Section 3 was first applied to m-reps.

In Paper #6, we propose the *bicycle chain shape model*. The analogy with bicycle chains stems from the fact that curves are represented by points having fixed pairwise distances similar to the constant distances between the pins enforced by the links in a bicycle chain. The model can be seen as a PDM with constraints on the point placements or as an L^2-like model with fixed parametrization and finite discretization. The rationale for the first viewpoint arises from problems in establishing correspondences for PDMs. In the paper, the outline of human vertebrae are manually annotated by medical experts but the annotations exhibit variation in the actual placement of the points. Redistributing the points to have constant pairwise distances reduces this variation and may lead to more robust statistics on the shapes. In addition, the pairwise-distance constraint results in the dimension of the shape space being roughly halved, and the model thus provides an example of how compactness can be increased by a non-linear restriction of the modeling space. From the second viewpoint, the fixed distances imply a constant speed parametrization, and distances are measured as length of paths in the non-linear shape space with the induced Euclidean metric on the tangent space measuring the cost of infinitesimal deformations.

In the paper, we use a *shooting* algorithm to compute geodesics and distances in the shape space. This approach is also used in some LDDMM algorithms [37] though path straightening algorithms are commonly used for the elastic models [32] and LDDMM [21]. The algorithms for computing optimal deformations with the kernel bundle and the higher order kernels in Paper #1 and Paper #2 are also shooting methods, and they are inspired by both [37] and the bicycle chain shooting algorithm. We use the same shooting approach for computing distances on the human pose manifold in Paper #5.

2.

Paper #1: *Sparse Multi-Scale Diffeomorphic Registration: the Kernel Bundle Framework*

Paper invited for submission to the Scale-Space and Variational Methods special issue in the Journal on Mathematical Imaging and Vision (JMIV). Submitted to JMIV, December 2011.

The paper is based on and extends the three conference papers [1, 2, 3].

Authors:

Stefan Sommer, François Lauze, Mads Nielsen, and Xavier Pennec

Notes:

We introduce the *kernel bundle* framework, a multi-scale extension of the LDDMM registration framework. The goal is to represent deformation at multiple scales and thus increase the capacity of sparse deformation representations while allowing compact representations. The latter is possible by the ability of the kernel bundle to represent *sparsity across scales*. We derive the KB-EPDiff evolution equations and prove the momentum conservation property. By applying sparse priors to the scale-momentum, we seek to represent deformation at the relevant scales only. This combines the modeling capacity of the kernel bundle with increased the compactness of the representation. The method is evaluated on synthethic and real examples, and, on a dataset of manually annotated lung CT images, we show that the increased capacity of the method does not impact the ablity of the method to generalize to test data; that the method removes the need for classical scale selection; and that the property of sparsity across scales is achieved.

Noname manuscript No.
(will be inserted by the editor)

Sparse Multi-Scale Diffeomorphic Registration: the Kernel Bundle Framework

Stefan Sommer · **François Lauze** · **Mads Nielsen** · **Xavier Pennec**

Received: date / Accepted: date

Abstract In order to detect small-scale deformations during disease propagation while allowing large-scale deformation needed for inter-subject registration, we wish to model deformation at multiple scales and represent the deformation compactly at the relevant scales only. This paper presents the *kernel bundle* extension of the LDDMM framework allowing multiple kernels at multiple scales to be incorporated in the registration while preserving much of the mathematical structure underlying the single-scale method. We combine sparsity priors with the kernel bundle resulting in compact representations across scales, and we present the mathematical foundation of the framework with derivation of the KB-EPDiff evolution equations. Through examples, we illustrate the influence of the kernel scale and show that the method achieves the important property of *sparsity across scales*. In addition, we demonstrate on a dataset of annotated lung CT images how the kernel bundle framework with a compact representation reach the same accuracy as the standard method optimally tuned with respect to scale.

Keywords kernel bundle · LDDKBM · LDDMM · diffeomorphic registration · scale space · computational anatomy

S. Sommer, F. Lauze, M. Nielsen
The Image Group, Department of Computer Science
University of Copenhagen, Denmark
Tel.: +4535321400
E-mail: sommer@diku.dk

M. Nielsen
BiomedIQ A/S
Copenhagen, Denmark

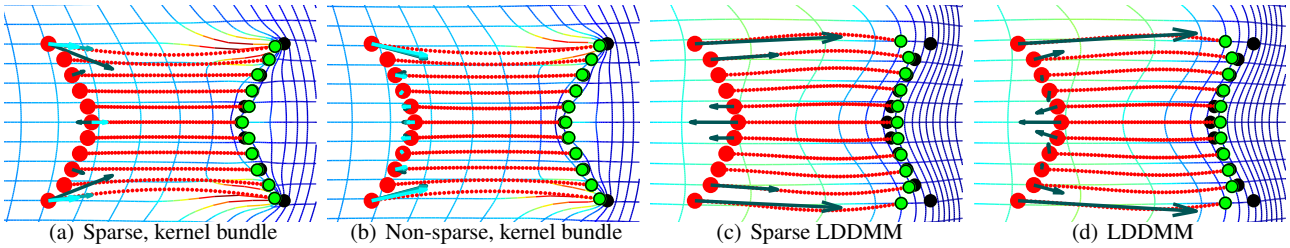
X. Pennec
Asclepios Project-Team
INRIA Sophia-Antipolis, France

1 Introduction

Deformation captured in image registration occur at multiple scales: lungs deform at large scale during the respiratory phases while disease progression may only be detected at small scales. Similarly, large-scale deformation is needed when registering brains from different subjects while e.g. atrophy in the hippocampus occur at small scales during the progression of Alzheimer's disease. Representing deformation at multiple scales is therefore useful when performing statistics on small-scale features over a population requiring large-scale inter-subject registration. In this paper, we develop a method that represents deformation at multiple scales while seeking to represent the deformation at the relevant scales only. The resulting sparse, multi-scale *kernel bundle* registration framework supports *sparsity across scales* while extending the range of deformation expressed by single-scale models. We derive and test the construction to show that the across scale sparsity is indeed achieved; that the extra capacity of the method does not hamper generalization to test data; and that the method removes the need for classical scale selection.

1.1 Background

The LDDMM framework is widely used in the field of computational anatomy to model deformation and perform registration of geometric objects. It provides convenient parametrization of flows of diffeomorphisms and a complete mathematical setting ensuring existence of optimal warps and allowing meaningful statistics to be performed on the registration results. Recent work has shown that the infinite dimensional space of parameters for the registration can be successfully approximated using sparse, finite dimensional representations [8,16]. However, the notion of kernels, which lies at



(a) Sparse, kernel bundle (b) Non-sparse, kernel bundle (c) Sparse LDDMM (d) LDDMM

Fig. 1 Matching eleven landmarks (red) to eleven landmarks (black) and results (green) with four registration methods: (a) the proposed *kernel bundle* multi-scale method with sparse prior; (b) the kernel bundle method without sparse prior; (c) LDDMM (single-scale) with sparse prior; and (d) LDDMM (single-scale) without enforced sparsity. The arrows show the initial momentum with different colors for each of the three scales in (a) and (b). Initially square grids are shown deformed by each diffeomorphism; the grids are colored with the log-trace of Cauchy-Green strain tensor. (d) The inherent single-scale behaviour of LDDMM causes large deviation between the landmarks and results (black and green). (c) This effect is increased when adding a sparse prior; the low number of non-zero momentum vectors indicate the sparsity. (b) With the kernel bundle and multiple scales, the algorithm matches the points well through the increased capacity in the deformation description. (a) Adding a sparse prior to the kernel bundle results in a compact representation (few non-zero momentum vectors) with sparsity across scales, and the sparser representation continues to provide a good match between the landmarks and results (black and green). This happens without sacrificing warp regularity: the deformed grid with sparse prior (a) is similar to the deformed grid without the prior (b).

the heart of the framework, and the kernel shape and scale impose restrictions on the sparse representations, and it limits the range of deformations the model is able to express.

The key to obtain sparse representations without limiting the range of the deformation model is to increase the *capacity* of the deformation description. Locally, the capacity can be increased with higher order kernels [16] but varying the spatial extend of the deformation requires multiple scales. Enabling LDDMM to model deformation occurring at multiple scales has been the subject of several works [2, 12, 13] resulting in improved registration results. Deformation at different scales may however occur at *different spatial locations*, and we wish to represent deformation at different locations at the appropriate scales *only*. This requires a multi-scale framework designed specifically to allow sparse representations. Consider registering two images of fairly uniform objects. The large-scale deformations can then be expected to be located at the center of the object while lower scale deformations occur close to the boundaries. The sum of kernels approach [13] will represent deformation at all scales at all spatial locations. In contrast, we aim for constructing a framework able to represent deformation at the appropriate scales *only*.

In order to achieve this, we introduce the *kernel bundle* framework (LDDKBM) which is designed to represent deformation individually at different scales. This in particular allows the momentum field to vanish at some scales while being non-zero at others at the same spatial locations, a property we denote *sparsity across scales*. By applying sparse priors on the momentum field at the different scales individually, we explicitly force the cross-scale sparsity, and the method therefore allows *sparse deformation description* across space and scales. The resulting framework greatly extends the *range* of deformations expressed by the sparse models while allowing *compact* representations of deformation occurring at multiple scales.

1.2 Deformation at Multiple Scales; An Example

Figure 1 shows a simple example of landmark matching. In order to register the points, movement is needed at both large and small scales, and the single-scale nature of the LDDMM algorithm limits its ability to match the points well. This fact becomes even more expressed when adding a sparsity prior. With multiple-scales, the match improves as seen by the reduced deviation between the landmarks and results. When adding a sparse prior to the kernel bundle, equivalent precision can be obtained with a compact description which exhibits sparsity across scales.

1.3 Related Work

The deformable template model pioneered by Grenander [10] and the flow approach by Christensen et al. [6] together with the theoretical contributions of Dupuis et al. and Trouvé [7, 19] started the development of the LDDMM deformation model. Beg et al. [1] developed algorithms for computing optimal diffeomorphisms in the framework, and the momentum representation has been used for statistics and for momentum based algorithms for the landmark matching problem [20]. The review paper [22] and the book [21] provide excellent overviews of the theory and applications of LDDMM in medical imaging.

Multi-scale extensions of LDDMM have been treated in several recent works. Bruveris et al. developed an extension of LDDMM allowing two scales through the use of semi-direct product groups [2] and Risser et al. [13] included scale

in LDDMM by adding kernels of different scales. The approach of Risser et al. does not divide the deformation description across scales and enforced sparsity will occur at all scales simultaneously. This is in direct contrast to the result we seek to obtain: we search for a representation that can handle both small and large scale features independently to allow different deformation at different scales, and we wish to allow the effect of enforced sparsity to occur at different scales individually. A representation supporting this *sparsity across scales* property is the main contribution of this paper, and the fact that the kernel bundle representation supports this property fundamentally differentiates it from the approach of Risser et al. Outside the registration setting, the effect on the underlying Hilbert spaces when scaling the kernel have been treated in [9]. Increasing the capacity of the deformation description locally can been obtained with higher order kernels [16].

The literature on sparse representations and sparse penalty functions is wide, and we will in this paper limit the discussion to a small set of such priors [4]. A control point formulation of LDDMM template-based image registration has been developed by Durrleman et al. [8]. Sparsity is enforced by a $\log -L^1$ penalty on the initial momenta, and the prior guides a search towards low-dimensional representations of deformation for populations of images. The method was developed for image registration but the sparse prior introduced apply to any finite dimensional LDDMM implementation. The fixed size of the kernels does however limit the expressiveness of the model. The fundamental idea behind the present paper is to remove this limitation by using kernels of multiple scales.

1.4 Content and Outline

This paper combines the conference papers [17, 15, 18] and adds additional new material. We aim at presenting a full account of the kernel bundle framework that in the previous papers is also denoted the LDDKBM method (KB for *k*ernel *b*undle). The new material comprises full derivation of the forwards and backwards gradient transport equations which are fundamental for computing optimal warps with the framework; additional algorithm information; discussion on the relation to other multi-scale approaches; and extended experiments section showing the obtained effect of sparsity across scales and using cross validation to tune the regularization weights for the method comparisons.

We start by discussing the variational formulation of LDDMM and the kernel bundle method before presenting the theoretical construction allowing the multi-scale representation. We relate the method to other multi-scale approaches before deriving the KB-EPDiff evolution equations. Next follows the forwards and backwards transport equations with implementation details and last the extended experiments section. The paper thus contributes by

(1) combining the previous work on the LDDKBM method [17], the evolution equations [15], and sparse and compact representations [18] to one account of the *kernel bundle* framework,
(2) giving a complete derivation of the forwards and backwards gradient transport equations together with algorithm details,
(3) discussing the relation between the kernel bundle and other LDDMM multi-scale approaches,
(4) providing extended experiments section showing in particular the ability to represent sparsity across scales.

2 Registration: the LDDMM and Kernel Bundle Variational Formulation

The kernel bundle framework extends the single-scale LDDMM (Large Deformation Diffeomorphic Metric Mapping) framework by allowing regularization at multiple-scales in the registration. We here provide an overview of the registration problem and the variational formulation used in both frameworks.

In the kernel bundle and LDDMM frameworks, registration is performed through the action of diffeomorphisms on geometric objects. The approach is very general and allows the frameworks to be applied to both landmarks, curves, surfaces, images, and tensors. In the case of landmarks, the action of a diffeomorphism φ takes the form $\varphi.x = \varphi(x)$, and given landmarks $x_1,\dots,x_N$ and $y_1,\dots,y_N$, the registration amounts to a search for φ such that $\varphi.x_i \sim y_i$ for all $i = 1,\dots,N$. In exact matching, we wish $\varphi.x_i$ be exactly equal to y_i but, more frequently, we allow some amount of inexactness to account for noise and give smoother diffeomorphisms. This is done by defining a quality of match measure U and a regularization measure E_1 to give a combined energy

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi)\,. \tag{1}$$

Here λ is a positive real representing the trade-off between regularity and goodness of fit and U is often the L^2-error which in the landmark case takes the form $U(\varphi) = \sum_{i=1}^{N} \|\varphi(x_i) - y_i\|^2$.

2.1 The Regularization Energy

The formulation of the regularization energy E_1 in the kernel bundle framework is an extension of the LDDMM formulation. We here introduce notation which will lead to the LDDMM formulation before describing the extension in the next section. Let the domain Ω be a subset of of $\mathbb{R}^d$ with

$d = 2,3$ in applications, and let V denote a Hilbert space of vector fields $v : \Omega \to \mathbb{R}^d$ such that V with associated norm $\|\cdot\|_V$ is included in $L^2(\Omega,\mathbb{R}^d)$ and admissible as defined in [21, Chap. 9]. Given a time-dependent vector field $t \mapsto v_t$ with

$$E_1(v_t) = \int_0^1 \|v_t\|_V^2 \, dt < \infty \tag{2}$$

the associated differential equation $\partial_t \varphi_t = v_t \circ \varphi_t$ has with initial condition $\varphi_s = \varphi$ a diffeomorphism φ_{st}^v as unique solution. The set G_V of diffeomorphisms built from V by such differential equations is a Lie group, and V is its tangent space at each point. The inner product on V associated to the norm $\|\cdot\|_V$ makes G_V a Riemannian manifold with right-invariant metric. Setting $\varphi_{00}^v = Id_\Omega$, the map $t \mapsto \varphi_{0t}^v$ is a path from Id_Ω to φ with energy given by (2). We will use this notation throughout the paper. A critical path for the energy is a geodesic on G_V, and the LDDMM regularization energy is defined by

$$E_1(\varphi) = \min_{v_t \in V, \varphi_{01}^v = \varphi} E_1(v_t) = \min_{v_t \in V, \varphi_{01}^v = \varphi} \int_0^1 \|v_s\|_V^2 \, ds \,, \tag{3}$$

i.e., it measures the minimal energy necessary to pass from Id_Ω to φ. The energy penalizes highly varying paths and, therefore, a low value of $E_1(\varphi)$ implies that φ is regular.

The regularity is ultimately controlled by the norm on V and this norm is associated to a *reproducing kernel* $K : \Omega \times \Omega \to \mathbb{R}^{d\times d}$. The kernel is often chosen to ensure rotational and translational invariance [21] and the Gaussian kernel $K(x,y) = \exp(\frac{\|x-y\|^2}{\sigma^2})\mathrm{Id}_d$ is a convenient and often used choice. The scaling factor σ is not limited to Gaussian kernels and allows for many kernels to vary the amount of regularization. Larger scales lead in general to higher regularization and smoother diffeomorphisms, whereas smaller kernels penalize higher frequencies less and often gives better matches. This phenomenon is in particular apparent for objects with sparse information and images with e.g. areas of constant intensity.

3 Kernels, Momentum and the Kernel Bundle

The kernel bundle framework extends LDDMM by equipping the diffeomorphism manifold G_V in LDDMM with vector bundles allowing deformation to be described at different scales. We start this section by discussing the relation between kernels and momentum in LDDMM before defining the kernel bundle and discussing the mathematical foundation behind the framework.

3.1 Kernel and Momentum

As a consequence of the assumed admissibility of V, the evaluation functionals $\delta_x : v \mapsto v(x) \in \mathbb{R}^d$ is well-defined and continuous for any $x \in \Omega$. Thus, for any $a \in \mathbb{R}^d$ the map $a \otimes \delta_x : v \mapsto a^T v(x)$ belongs to the topological dual V^* consisting of the continuous linear maps of V. This in turn implies the existence of spatially dependent matrices $K : \Omega \times \Omega \to \mathbb{R}^{d\times d}$, the *kernel*, such that, for any constant vector $a \in \mathbb{R}^d$, the vector field $K(\cdot,x)a \in V$ represents $a \otimes \delta_x$ and $\langle K(\cdot,x)a, v\rangle_V = a \otimes \delta_x(v)$ for any $v \in V$, point $x \in \Omega$ and vector $a \in \mathbb{R}^d$. This latter property is denoted the reproducing property and gives V the structure of a reproducing kernel Hilbert space (RKHS). Tightly connected to the norm and kernels is the notion of *momentum* given by the linear momentum operator $L : V \to V^* \subset L^2(\Omega,\mathbb{R}^d)$ which satisfies

$$\langle Lv, w\rangle_{L^2(\Omega,\mathbb{R}^d)} = \int_\Omega \big(Lv(x)\big)^T w(x) dx = \langle v, w\rangle_V \tag{4}$$

for all $v, w \in V$. The momentum operator connects the inner product on V with the inner product in $L^2(\Omega,\mathbb{R}^d)$, and the image Lv of an element $v \in V$ is denoted the momentum of v. The momentum Lv might be singular and in fact $L\big(K(\cdot,y)a\big)(x)$ is the Dirac measure $\delta_y(x)a$. Considering K as the map $a \mapsto \int_\Omega K(\cdot,x)a(x)dx$, L can be viewed as the inverse of K. Confer [21] for a thorough introduction to reproducing kernels, especially with a view towards the LDDMM framework.

3.2 The Kernel Bundle

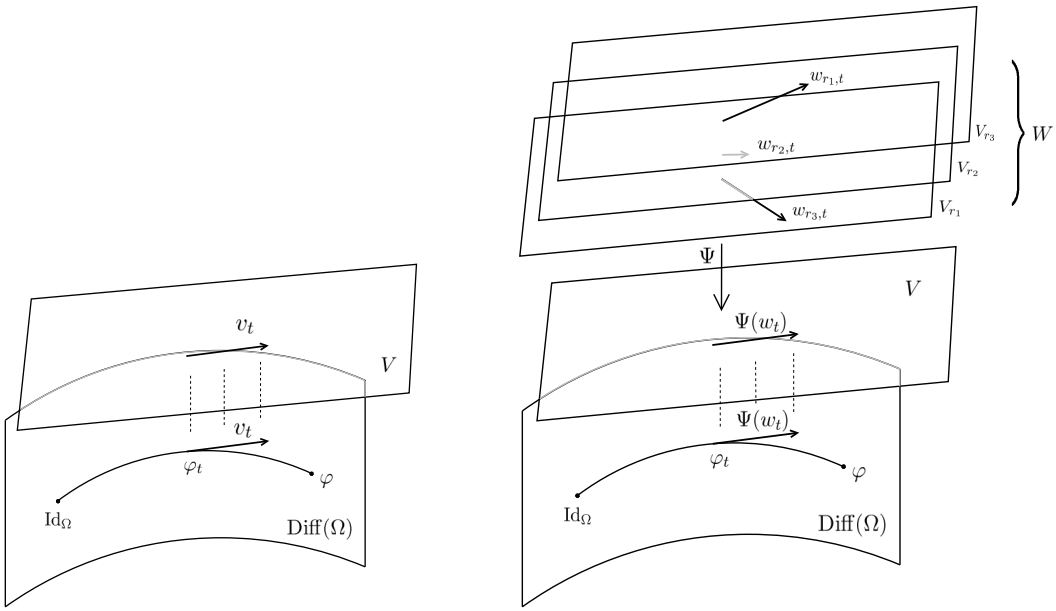
In order to describe deformation at different scales, we extend in the following the tangent vector space V to a family of vector spaces W which will eventually lead to the bundle construction. We consider a parameter set I_W and subspaces V_r, $r \in I_W$ of the tangent space V where each V_r is equipped with a norm $\|\cdot\|_r$, corresponding kernel K_r, and momentum operator L_r. Typically, I_W will be a discrete set or a closed and bounded interval of $\mathbb{R}^+$ representing different scales. We then let W be the space of functions $w : I_W \to V$, $w_r \in V_r$ such that

$$\int_{I_W} \|w_r\|_r^2 \, dr < \infty \quad \text{and} \quad \int_{I_W} \|w_r\|_r \, dr < \infty \,.$$

The vector space structures on V_r induce a vector space structure on W, and it can be shown that under reasonable assumptions, the inner product

$$\langle v, w\rangle_W = \int_{I_W} \langle v_r, w_r\rangle_r \, dr, \; v, w \in W$$

turns W into a Hilbert space. With this construction, we obtain a vector bundle $G_V \times W$, the *kernel bundle*, allowing kernels of different sizes and shapes. A map $\Psi : G_V \times W \to TG_V = G_V \times V$ allows parts w_r of a bundle vector $w \in W$ at each scale r to be combined to one derivative vector in V. Ψ is defined by integration, i.e. $\Psi(w) = \int_{I_W} w_r \, dr$.



(a) In LDDMM, a path on the manifold $G_V \subset \mathrm{Diff}(\Omega)$ is generated by integrating the time varying vector field v_t in the tangent space V.

(b) Wit the kernel bundle, a path w_t in the vector space W, here constructed from V_{r_1}, V_{r_2}, and V_{r_3}, sum through the map Ψ to a vector field $\Psi(w_t)$, which in turn generates a path on $\mathrm{Diff}(\Omega)$.

Fig. 2 The manifold view of LDDMM and the kernel bundle.

We note that the parameter space I_W can be a compact interval or finite set of scalars in which case the integral reduces to just a sum. Often, it will be an interval specifying a scale range, and a practical implementation will discretize the interval into a finite set of scalars.

3.3 Flows in the Bundle Setting

Similarly to the connection between paths in V and paths on the manifold G_V, we get using the map Ψ a relation between paths $w_t = \{w_{r,t}\}_r$ in W and paths in G_V by

$$w_t \mapsto \varphi_{0t}^{\Psi(w)}\,, \tag{5}$$

i.e. $\varphi_{0t}^{\Psi(w)}$ is the path starting at Id_Ω with derivative $\partial_t \varphi_{0t}^{\Psi(w)} = \Psi(w_t) \circ \varphi_{0t}^{\Psi(w)}$. We can measure the energy of a bundle path w_t by

$$E_1(w_t) = \int_0^1 \|w_s\|_W^2 \, ds\,, \tag{6}$$

and, using this energy, we get a new definition of the regularization energy E_1:

$$E_1(\varphi) = \min_{w_t \in W, \varphi_{01}^{\Psi(w)}=\varphi} E_1(w_t) = \min_{w_t \in W, \varphi_{01}^{\Psi(w)}=\varphi} \int_0^1 \|w_s\|_W^2 \, ds \tag{7}$$

Together with a quality of match measure $U(\varphi)$, this defines the registration problem in the kernel bundle framework as the search for diffeomorphisms minimizing

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi) \tag{8}$$

with E_1 given by (7). By design, paths in the kernel bundle generating the diffeomorphisms have components at each scale, and this is precisely the property that will later allow us to enforce sparsity at different scales individually. This will be done by adding priors that affect the individual scale components of bundle vectors to (8).

The above registration energy should be compared with the LDDMM formulation (1) using the regularization (3). It is immediately clear that the kernel bundle formulation is an extension of the LDDMM regularization, since the original regularization is the special case with only one scale and hence $W = V$.

3.4 Structure of W

It is interesting to note that W possesses a structure very similar to a RKHS. On V we have for each $x \in \Omega$ and $a \in \mathbb{R}^d$ the evaluation functionals $a \otimes \delta_x(v) = a^T v(x)$. Using the integral map Ψ defined above, we define the linear maps on W

$$a \otimes \delta_x^\Psi(w) := \int_{I_W} a \otimes \delta_x(w_r) dr = \int_{I_W} a^T w_r(x) dr = a \otimes \delta_x(\Psi(w))\,.$$

As seen from the equation, the maps evaluate w_r at each scale and integrate the results using Ψ. These maps are continuous and hence in the dual W^*. For the elements $K(\cdot,x)a = \{K(\cdot,x)_r a\}_r \in W$, we have

$$\begin{aligned}\langle K(\cdot,x)a, K(\cdot,y)b\rangle_W &= \int_{I_W} \langle K(\cdot,x)_r a, K(\cdot,y)_r b\rangle_r \, dr \\ &= \int_{I_W} a^T K_r(x,y) b \, dr = a^T \int_{I_W} K_r(x,y) b \, dr \\ &= a \otimes \delta_x^\Psi (K(\cdot,y)b) = a^T \Psi(K(x,y)b)\end{aligned}$$

which is similar to the reproducing property in V except for the integration performed by Ψ on the right-hand side of the equation. Also, close to the RKHS situation, we see that

$$\begin{aligned}\langle K(\cdot,x)a, w\rangle_W &= \int_{I_W} \langle K(\cdot,x)_r a, w_r\rangle_r \, dr \\ &= \int_{I_W} a^T w_r(x) \, dr = a \otimes \delta_x^\Psi(w) \ , \ w \in V\end{aligned}$$

again with the integration of w occuring in $a \otimes \delta_x^\Psi(w)$.

3.5 Multi-Scale Representation and Relation to other Approaches

With the kernel bundle, the momentum components can vary over scale, and any combination of small and large scale features at each spatial location can be represented. In particular, the bundle allows sparse priors to force vanishing momentum at one scale while allowing it to be non-zero at other scales at the same position. The effect is to allow representing deformation compactly with non-zero components only at the right scales.

In contrast to this, the simultaneous coarse and fine method developed by Risser et al. in [12, 13] builds a kernel by summing Gaussians of different scale. This effectively changes only the shape of the kernel and does not allow different momentum at different scales. If momenta vanish, they will vanish at all scales simultaneously, and, therefore, the ability to represent sparsity across scales that we search for here is not possible.

When not using sparse priors and when the L^2-norm is combined linearly across scales, Bruveris et al. [3] showed that optimal deformations with the kernel bundle coincide with results obtained with the sum of kernels approach. Thus, though the kernel bundle is able to represent sparsity across scales, the cross-scale sparsity will not occur without adding more scale information to the system. This seems challenging with the approach of Risser et al. but adding such information becomes straightforward with the scale decoupled bundle representation which illustrates the descriptive power offered by the kernel bundle. Imposing sparse priors as we pursue later in this paper constitutes an example of this, and, as we will see in the experiments, optimal deformations with a prior do indeed increase the compactness of the representation and exhibit sparsity across scales. Correspondingly, incorporating scale information from the data term can guide the deformation model further towards the right mixture of scales and allow momentum-based statistics [20] to be performed across scale. This is again possible with the decoupled bundle representation, and we are currently pursuing this path.

4 Evolution Equations: Kernel Bundle EPDiff

In the single scale LDDMM case, the EPDiff equations describes the evolution of optimal paths for the registration problem. They are most often formulated in the following continuous form: let $a_t = Lv_t$ denote the momentum at time t and assume that φ_t is a path minimizing $E_1(\varphi)$ with $\varphi_1 = \varphi$ minimizing $E(\varphi)$ and v_t is the derivative of φ_t. Then v_t satisfies the system

$$\begin{aligned}v_t &= \int_\Omega K(\cdot,x) a_t(x) dx \ , \\ \frac{d}{dt} a_t &= -Da_t v_t - a_t \nabla \cdot v_t - (Dv_t)^T a_t \ .\end{aligned}$$

The first equation connects the momentum a_t with the velocity v_t, and the second describes the evolution of the momentum. The EPDiff equations can be interpreted as geodesic equations on the manifold G_V, and they are important for implementations since they limit the search for optimal paths to paths satisfying the system.

As we will show in this section, there exists similar equations with the kernel bundle: if $\Psi(w_t)$ is the derivative of the path of diffeomorphisms φ_t minimizing (8) with $\varphi = \varphi_1$ minimizing (8) then

$$\begin{aligned}w_{r,t} &= \int_\Omega K_r(\cdot,x) a_{r,t}(x) dx \ , \\ \frac{d}{dt} a_{r,t} &= \int_{I_W} -Da_{r,t} w_{s,t} - a_{r,t} \nabla \cdot w_{s,t} - (Dw_{s,t})^T a_{r,t} \, ds \ .\end{aligned} \tag{9}$$

with $a_{r,t}$ being the momentum for the part $w_{r,t}$ of w_t. In essence, the standard EPDiff equations are integrated over the parameter space I_W to obtain the evolution of the momentum at each scale, and, in particular, the result will imply that the momentum conservation property of LDDMM also holds in kernel bundle case. We will derive the KB-EPDiff equations in a more general form which implies the above formulation, and, for doing this, we will follow the strategy in [21] for the LDDMM case.

4.1 Euler-Lagrange equations

For any time varying path w_t in W, we denote by $\varphi_{t_1 t_2}^{\Psi(w)}$ the diffeomorphism obtained by integrating $\Psi(w_t)$ from time t_1

to time t_2. The end of the integrated path $\varphi_{01}^{\Psi(w)}$ is the diffeomorphism used for the registration. For the energy $E(w_t) = E_1(w_t) + \lambda U(\varphi_{01}^{\Psi(w)})$, we consider a variation $h_t \in W$ and calculate

$$\frac{d}{d\varepsilon}E(w_t+\varepsilon h_t) = 2\int_0^1 \langle w_t, h_t\rangle_W \, dt + \lambda \frac{d}{d\varepsilon}U(\varphi_{01}^{\Psi(w)+\varepsilon\Psi(h)}) \, . \tag{10}$$

Following [21], we define $\mathrm{Ad}_\varphi v(x) = (D\varphi\, v)\circ\varphi^{-1}(x)$ for $v \in V$ and get a functional Ad^*_φ on the dual V^* of V by $(\mathrm{Ad}^*_\varphi \rho | v) = (\rho | \mathrm{Ad}_\varphi(v))$. It is shown in [21] that a variation $\tilde{h}_t$ in V of the match functional satisfies

$$\frac{d}{d\varepsilon}U(\varphi_{01}^{v+\varepsilon\tilde{h}}) = \int_0^1 \left(\mathrm{Ad}^*_{\varphi_{t1}^v}\bar{\partial}U(\varphi_{01}^v)\big|\tilde{h}_t\right) dt$$

with $\bar{\partial}U$ denoting the Eulerian differential of U (see [21, Chap. 10]). Inserting into (10) gives

$$\frac{d}{d\varepsilon}E(w_t+\varepsilon h_t) = 2\int_0^1 \langle w_t, h_t\rangle_W \, dt + \lambda \int_0^1 \left(\mathrm{Ad}^*_{\varphi_{t1}^{\Psi(w)}}\bar{\partial}U(\varphi_{01}^{\Psi(w)})\big|\Psi(h_t)\right) dt \, . \tag{11}$$

For each r, we define the operator $\mathrm{Ad}^{T,r}_\varphi v = K_r(\mathrm{Ad}^*_\varphi(L_r v))$ which then satisfies $\left\langle \mathrm{Ad}^{T,r}_\varphi v, w\right\rangle_r = (\mathrm{Ad}^*_\varphi(L_r v)|w)$, and we can now derive the fundamental results [21, Prop. 11.6/Cor. 11.7] in the bundle case:

Proposition 1 *If w_t is an optimal path for E then for almost every $r \in I_W$,*

$$w_{t,r} = \mathrm{Ad}^{T,r}_{\varphi_{t1}^{\Psi(w)}} w_{1,r}$$

with $w_{1,r} = -\frac{1}{2}\nabla^{V_r}U(\varphi_{01}^{\Psi(w)})$.

Proof Assume instead that there exists a time varying h_t in W and $t \in [0,1]$ such that

$$\begin{aligned} 0 &< \int_{I_W}\left\langle w_{t,r} - \mathrm{Ad}^{T,r}_{\varphi_{t1}^{\Psi(w)}} w_{1,r}, h_{t,r}\right\rangle_r dr \\ &= \int_{I_W}\langle w_{t,r}, h_{t,r}\rangle_r dr - \int_{I_W}\left\langle \mathrm{Ad}^{T,r}_{\varphi_{t1}^{\Psi(w)}} w_{1,r}, h_{t,r}\right\rangle_r dr \\ &= \langle w_t, h_t\rangle + \frac{1}{2}\int_{I_W}(\mathrm{Ad}^*_{\varphi_{t1}^{\Psi(w)}}\bar{\partial}U(\varphi_{01}^{\Psi(w)})\big|h_{t,r})dr \\ &= \langle w_t, h_t\rangle + \frac{1}{2}(\mathrm{Ad}^*_{\varphi_{t1}^{\Psi(w)}}\bar{\partial}U(\varphi_{01}^{\Psi(w)})\big|\Psi(h_t)) \, . \end{aligned}$$

But the right hand side vanishes for all t and all h_t by (11) and the fact that w_t is optimal for E, a contradiction.

Corollary 1 *Under the same conditions, for almost every $r \in I_W$,*

$$w_{t,r} = \mathrm{Ad}^{T,r}_{\varphi_{t0}^{\Psi(w)}} w_{0,r} \, . \tag{12}$$

The proof of the corollary is identical to the proof of [21, Cor. 11.7].

4.2 Scale Conservation and KB-EPDiff

In the kernel bundle, the momentum of a path may differ across scales. For a path w_t in W, we let a_t be the bundle momentum defined by $a_{t,r} = L_r(w_{t,r})$ recalling that L_r is the momentum operator at scale r. For each t, we can consider a_t to be in the dual W^* by $(a_t|\tilde{w}) = \int_{I_W}(a_{t,r}|\tilde{w}_r)dr$ which is continuous since

$$\left|(a_t|\tilde{w})\right| \le \left|\int_{I_W}(a_{t,r}|\tilde{w}_r)dr\right| = \left|\int_{I_W}\langle w_{t,r}, \tilde{w}_r\rangle_r dr\right| \le \|w_t\|\|\tilde{w}\| \, .$$

Suppose now w_t satisfies the transport equation (12) for almost every $r \in I_W$. Then for all $\tilde{w} \in W$,

$$\begin{aligned} (a_t|\tilde{w}) &= \int_{I_W}\langle w_{t,r}, \tilde{w}_r\rangle_r dr = \int_{I_W}\left\langle \mathrm{Ad}^{T,r}_{\varphi_{t0}^{\Psi(w)}} w_0, \tilde{w}_r\right\rangle_r dr \\ &= \int_{I_W}\left\langle w_{0,r}, \mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}\tilde{w}_r\right\rangle_r dr = (a_0|\mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}\tilde{w}) \end{aligned} \tag{13}$$

where $\mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}\tilde{w}$ is the element of W obtained by applying $\mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}$ to each $\tilde{w}_r$. The above equation shows that the momentum at time t is completely specified by the momentum at time 0 and thus reproduces the momentum conservation property for LDDMM. Note that since $\tilde{w}$ can be chosen arbitraly in (13), the momentum is conserved for each scale separately. By differentiating $\mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}\tilde{w}$, the momentum conservation property directly implies the equation

$$\partial_t(a_t|\tilde{w}) = -\big(a_t|D\Psi(w_t)\,\tilde{w} - D\tilde{w}\,\Psi(w_t)\big) \tag{14}$$

or, equivalently,

$$\partial_t a_t + \mathrm{ad}^*_{\Psi(w_t)}a_t = 0$$

with $\left(\mathrm{ad}^*_{\Psi(w_t)}a_t|\tilde{w}\right) = \big(a_t|D\Psi(w_t)\,\tilde{w} - D\tilde{w}\,\Psi(w_t)\big)$. Both equations imply the system (9) and extend the EPDiff equations for LDDMM. We denote them KB-EPDiff.

An important difference from the single-scale framework relates to the energy along optimal paths. The relation to geodesics in LDDMM suggests that the norm $\|v_t\|_V$ is constant in t when v_t is optimal for $E_1(\varphi)$. This is in fact the case for LDDMM. With the kernel bundle, momentum is conserved along optimal paths of $E_1(\varphi)$ though $\|w_t\|_W$ is not constant. This occurs because the new energy is not directly related to a metric in the Riemannian sense.

4.3 KB-EPDiff for Landmarks: An Example

To give a concrete application of the KB-EPDiff equations, we redo the calculation for LDDMM landmark matching with scalars kernels to arrive at the corresponding system with the bundle. The initial momentum $a_{0,r}$ will in this case be supported at the N landmarks x_i, $i = 1\ldots,N$, i.e. $a_{0,r} = \sum_{i=1}^{N} a_{0,r,i} \otimes \delta_{x_i}$ with vectors $a_{0,r,i} \in \mathbb{R}^d$. We let $x_{t,i}$ denote the trajectory of the ith landmark so that $x_{t,i} = \varphi_{0t}^{\Psi(w)}(x_{0,i})$.

Letting $a_{t,r,i} = (D\varphi_{t0}^{\Psi(w)})^T a_{0,r,i}$, we get from (13)

$$\begin{aligned}
\big(a_{t,r}|\tilde{w}\big) &= \left(\mathrm{Ad}^*_{\varphi_{t0}^{\Psi(w)}}\left(\sum_{i=1}^{N} a_{0,r,i}\otimes\delta_{x_{0,i}}\right)\Big|\tilde{w}\right)\\
&= \left(\sum_{i=1}^{N} a_{0,r,i}\otimes\delta_{x_{0,i}}\Big|\mathrm{Ad}_{\varphi_{t0}^{\Psi(w)}}(\tilde{w})\right)\\
&= \sum_{i=1}^{N} a_{0,r,i}^T (D\varphi_{t0}^{\Psi(w)}\,\tilde{w})\circ\varphi_{0t}^{\Psi(w)}(x_{0,i})\\
&= \left(\sum_{i=1}^{N} a_{t,r,i}\otimes\delta_{x_{t,i}}\Big|\tilde{w}\right)\ .
\end{aligned}$$

Since $\frac{d}{dt}(D_{x_{t,i}}\varphi_{t0}^{\Psi(w)})^T = -D_{x_{t,i}}\Psi(w_t)^T(D_{x_{0,i}}\varphi_{t0}^{\Psi(w)})^T$, the derivative of the momentum satisfies

$$\frac{d}{dt}a_{t,r,i} = \frac{d}{dt}\big((D\varphi_{t0}^{\Psi(w)})^T a_{0,r,i}\big) = -D_{x_{t,i}}\Psi(w_t)^T a_{t,r,i}\ .$$

The trajectories of the landmarks and momentum evolution is therefore completely described by the system

$$\begin{aligned}
\Psi(w_t) &= \int_{I_W}\textstyle\sum_{l=1}^{N} K_r(\cdot,x_{t,l})a_{t,r,l}\,dr\\
\tfrac{d}{dt}a_{t,r,i} &= -\left(\int_{I_W}\textstyle\sum_{l=1}^{N} D_1\big(K_s(x_{t,i},x_{t,l})a_{t,s,l}\big)^T ds\right)a_{t,r,i}\\
x_{t,i} &= \varphi_{0t}^{\Psi(w)}(x_{0,i})\ .
\end{aligned}\tag{15}$$

Note that the system is finite if I_W is finite.

5 Sparse Kernel Bundle Representation

In a variety of applications, it is useful to obtain compact representations in the form of *sparse* solutions [4]. The standard method of obtaining sparsity is to add a penalty function to a variational formulation of the problem. The penalty function is also denoted a *sparse prior*.

Combining sparsity and multi-scale representations promises enhancements for both pairwise and group-wise registration: For statistics following pairwise registration with the aim of retrieving scale information, it is paramount to represent the deformation at the right scale *only*. Low-scale deformation may be represented by high-scale momenta but will require a higher number of non-zero parameters than if represented at the correct low-scale. Enforcing sparsity makes the low-scale representation more likely. This property is possible with *sparsity across scales* as discussed below.

For group-wise registration, each pair of images may be registered with a sparsely parameterized deformation. However, the non-zero momenta may have different spatial localization for the different pairs of images. Sparsity should therefore in this case be applied on a group level. Inter-subject registration may however emphasize the need for multi-scale representation: if modeling inter-subject differences using only a single large-scale, small scale features may be lost. If using only small-scale deformation, the representation will not be sparse.

Durrleman et al. [8] showed that the number of points in a finite control point formulation of LDDMM can be controlled by a $\log - L^1$ like penalty term: a weight λ_{sp} and truncated log function

$$f_{\log^c}(x) = \max(\log(x),\log(c)) - \log(c)$$

is applied to the norms of the set of N single-scale momenta resulting in the extension of (8) to the energy

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi) + \lambda_{sp}\sum_{l=1}^{N} f_{\log^c}(\|a_{0,l}\|)\ . \tag{16}$$

The prior is added to all elements of a population of images, and it is shown that a fairly large reduction in the number of non-zero momenta does not affect the registration results much.

In the multi-scale case, the connection (4) between the momentum space and the kernel bundle can also be exploited in order to define penalty functions. Sparsity is generally formulated via the L_0-norm which on the bundle momentum take the form

$$\|w\|_{L^0} = \int_{I_W}\mathrm{Area}\{L_r w_r \neq 0\}dr\ .$$

This reduces to the number of non-zero momentum vectors $\|w\|_{L^0} = \int_{I_W}|\{Lw_r \neq 0\}|dr$ in the finite-dimensional case. For sparse problems in general, optimization based on L_0 penalty functions is a combinatorial problem and thus computationally prohibitive. Instead, the L_0-norm is approximated by the L_1-norm or similar functions.

In the multi-scale, finite dimensional setting, we parametrize the bundle momentum in the same way as the momentum is represented in the single-scale case: for N landmarks and R scales or, equivalently, for N control points and R scales in image registration, $N\cdot R$ vectors $a_{0,l,r}$ will specify the initial momentum. We then formulate a multi-scale sparse registration functional extending (16) by

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi) + \sum_{r=1}^{R}\lambda_{sp,r}\sum_{l=1}^{N} f(a_{0,l,r}) \tag{17}$$

and we require the evolution of $a_{t,l,r}$ to follow the KB-EPDiff equations. Here $\lambda_{sp,r}$ denote scale-dependent weights on the sparse prior $f : \mathbb{R}^d \to \mathbb{R}$. As in the single-scale case, the idea is to push small momentum vectors towards zero without affecting large momenta much. We denote registration governed by (17) *sparse kernel bundle* registration.

5.1 Choice of Prior

Approximations of the L_0-norm aiming to ease the complexity of the combinatorial optimization has been considered in many applications [4]. Though a full discussion of this subject out of scope of this paper, we will provide a brief rationale for our choice of penalty function. We note that ensuring convexity is not a major concern in this setting because the non-linearity of the connection between initial momenta and the match functional U makes the energy (8) non-convex even before adding the prior.

The most widely used approximation is probably the L^1-norm which provide sparse solutions but has the downside of penalizing large momenta relatively hard, and it therefore provides poor approximation of the L_0-norm in such cases. The L^1-norm has been applied to LDDMM in addition to $f_{\log^c}$ [11]. Candès et al. [4] proposes several penalty functions including the function

$$f_{\log,\varepsilon}(x) = \log(1 + x/\varepsilon) \ .$$

Figure 3 illustrates the approximation of the L_0-norm provided by the L_1-norm, $f_{\log^c}$, and $f_{\log,\varepsilon}$. Both $f_{\log^c}$ and $f_{\log,\varepsilon}$ suffer less from the poor approximation for large momenta. Both necessitates a choice of parameter, c or ε. Though $f_{\log,\varepsilon}$ may seem more natural than $f_{\log^c}$ which is zero for small values, the gradient of $f_{\log,\varepsilon}$ may cause numerical issues close to zero. In the experiments section, we use $f_{\log^c}$ to get results comparable with the single-scale algorithm in [8].

5.2 Sparsity Across Scales

An important quality of the sparse, multi-scale construction is that a momentum vector $a_{0,l,r}$ at scale r may be zero while a momentum vector $a_{0,l,r'}$ at scale r' for the *same point* may be non-zero. Hence, a purely low-scale deformation may be represented with momenta being non-zero at that particular scale only. The kernel bundle construction is made explicitly to allow independent velocity at the different scales, and the behaviour of sparsity across scales is allowed by this fact. As we will see in the experiments, optimal deformations computed with a sparse prior do indeed exhibit sparsity across scales.

The weights $\lambda_{sp,r}$ should ideally be chosen by cross-validation in same way the weight λ in (8) and the weighting between scales in the bundle are determined. At this point, we heuristically choose $\lambda_{sp,r}$ either constant in r or $\lambda_{sp,r} = \lambda_{sp}/r^\alpha$ for a fixed scalar λ_{sp} and exponent $\alpha \geq 0$ in order to compensate for the often larger momenta at small scales.

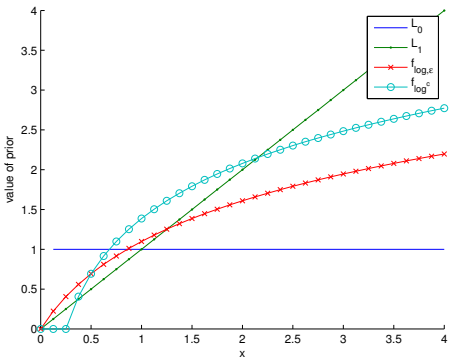


Fig. 3 The L_0-norm for real valued data, and the approximations L_1, $f_{\log^c}$, and $f_{\log,\varepsilon}$ with $\varepsilon = 0.5$ and $c = 0.25$. The L_1-norm provides poor approximation for large values. The truncated log $f_{\log^c}$ is not non-zero for small values.

6 Implementation

We here describe how optimal registrations with the kernel bundle can be computed in the case of landmark matching. Extending the method to images using a control point formulation similar to [8] and [16] poses no conceptual problem.

The running time will primarily be dominated by the backwards gradient transport described below. The system parallelizes well and can be implemented on GPU hardware [14]. The cost of adding a sparsity prior and computing its gradient is insignificant compared to the cost of integrating the flow equations described below. We do not experience any substantial differences in the number of iterations of the optimization procedure with and without sparse priors. The computation time is primarily a function of the number of landmarks and the number of included scales.

6.1 Algorithm

Since the evolution of the bundle momentum and velocity are required to follow the KB-EPDiff equations, we can optimize (17) using gradient based optimization strategies. A simple gradient descent scheme will given an guess for the initial momentum a_0 calculate the gradient $\nabla E(w_0) = \nabla E_1(w_0) + \lambda \nabla U(w_0)$ using $w_0 = Ka_0$, add the gradient from the sparsity term if using sparse prior, and update a_0 by

adding a vector proportional to the gradient. In practice, we use limited-memory BFGS updates[1].

The gradient $\nabla E(w_0)$ can be computed using a two step algorithm: the initial bundle velocity w_0 is transported forward in time to obtain the diffeomorphism φ before flowing the gradient at time $t = 1$ backwards to obtain the gradient $\nabla E(w_0)$ at $t = 0$. The gradient $\nabla U(w_1)$ at $t = 1$ is provided by the similarity measure; if U measure the L^2-error, the gradient is just the vector with the ith component being $2(x_{1,i} - y_i)$ where y_i are the target points.

The KB-EPDiff equations governing the forward integration and the backwards gradient transport constitute nonlinear ODEs which are finite if the set of scales I_W is finite. In practice, I_W is a discretization $\{s_1, \ldots, s_R\}$ of an interval $[s_1, s_R]$ using R scalars. The ODEs can be integrated using standard Runge-Kutta integrators such as MATLAB's `ode45` solver. The systems are described in detail below.

The sparse penalty functions considered here have gradients

$$\nabla f_{\log^c}(a_{0,l,r}) = \lambda_{sp,r} a_{0,l,r} / \|a_{0,l,r}\|^2 ,$$
$$\nabla f_{\log,\varepsilon}(a_{0,l,r}) = \lambda_{sp,r} a_{0,l,r} / ((e + \|a_{0,l,r}\|)\|a_{0,l,r}\|) .$$

If applying $\nabla f_{\log^c}$, $\|a_{0,l,r}\|$ is considered zero if it is less than c in which case we do not add the gradient to $\nabla E(w_0)$. Pruning of small values $a_{0,l,r}$ may be done during the optimization process but does not seem to effect stability of the algorithm much.

6.2 Forwards and Backwards Transport

The diffeomorphism φ is determined by w_0 by the KB-EPDiff equations, and the forward transports integrates the KB-EPDiff system (15) to generate φ. The system is a non-linear ODE with w_0 and the point positions $x_1, \ldots, x_N$ as initial values.

Because w_0 through the evolution of w_t is uniquely linked to w_1, $U(\varphi)$ can in addition be considered a function of w_1. The gradient $\nabla U(w_0)$ can be obtained by differentiating (15) and solving the transpose system backwards with $\nabla U(w_1)$ as initial condition. This approach is described in the single-scale case in [21]. With multiple scales, the gradient $\nabla E_1(w_0)$ can be computed simultaneously with $\nabla U(w_1)$ by adding it to the backwards ODE. Combined, the gradient $\nabla E(w_0)$ can be found as the solution at $t = 0$ of an affine, non-autonomous ODE

$$\dot{y}_t = v_t + M_t y_t \tag{18}$$

integrated from $t = 1$ to $t = 0$. The linear component transports $\nabla U(w_t)$ while the affine component transport $\nabla E_1(w_t)$. We provide explicit form of this system below.

[1] See e.g. `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

As in Section 4.3, we let $x_{t,i}$ denote the point positions at time t and the set of time-dependent vectors $a_{t,r,i}$ is the momentum of the flow. These components are computed from the forward integration of the KB-EPDiff equations (15). Note that the momenta have components at each scale r. By differentiating the KB-EPDiff equations we obtain the linear ODE

$$\dot{y}_t = B_t y_t . \tag{19}$$

The matrix M_t in the backwards equations (18) arises as the transpose of the matrix B_t. Both systems (19) and (18) have components coding the variation in point positions and momentum, respectively. We denote these components $b^x_{t,k}$ and $b^a_{t,k,r}$ for (19) and $m^x_{t,k}$ and $m^a_{t,k,r}$ for (18). Here $k = 1, \ldots, N$ and we consider the case of a finite set of scales R so that $r = 1, \ldots, R$. We assume the kernel K is scalar $K(x,y) = \gamma(\|x-y\|^2)\mathrm{Id}_d$ with a real function γ and write $\gamma_{t,kl}$ for $\gamma(\|x_{t,k} - x_{t,l}\|^2)$. Differentiating (15) then provides the components of (19):

$$\begin{aligned}
b^x_k &= \sum_{r=1}^{R}\sum_{l=1}^{N} \gamma^r_{kl} b^a_{l,r} + 2\sum_{r=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^r_{kl}(x_k - x_l)^T(m^x_k - m^x_l)a_{l,r} \\
b^a_{k,s} &= -2\sum_{r=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^r_{kl}(a^T_{k,r} m^a_{l,r} + a^T_{l,r} m^a_{k,s})(x_k - x_l) \\
&\quad - 2\sum_{r=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^r_{kl} a^T_{k,s} a_{l,r}(m^x_k - m^x_l) \\
&\quad - 4\sum_{r=1}^{R}\sum_{l=1}^{N} \ddot{\gamma}^r_{kl} a^T_{k,s} a_{l,r}(x_k - x_l)^T(m^x_k - m^x_l)(x_k - x_l)
\end{aligned}$$

where we omitted the time dependence of all terms to keep the notation compact. By transposing B_t, we get M_t and hence the linear parts of (18). This is in components

$$\begin{aligned}
m^x_k &= -2\sum_{r=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^r_{kl}(a^T_{k,r} m^x_l + a^T_{l,r} m^x_k)(x_k - x_l) \\
&\quad + 2\sum_{r,r'=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^{r'}_{kl}(a^T_{k,r} a_{l,r'} m^a_{k,r} - a^T_{k,r'} a_{l,r} m^a_{l,r}) \\
&\quad + 4\sum_{r,r'=1}^{R}\sum_{l=1}^{N} \ddot{\gamma}^{r'}_{kl}(x_k - x_l)^T(a^T_{k,r} a_{l,r'} m^a_{k,r} - a^T_{k,r'} a_{l,r} m^a_{l,r})(x_k - x_l) \\
m^a_{k,s} &= -\sum_{l=1}^{N} \gamma^s_{kl} m^x_l + 2\sum_{r=1}^{R}\sum_{l=1}^{N} (x_k - x_l)^T(\dot{\gamma}^r_{kl} m^a_{k,r} - \dot{\gamma}^s_{kl} m^a_{l,r})_{l,r} .
\end{aligned}$$

The simpler affine term has components

$$\begin{aligned}
m^x_k &= -4\sum_{r=1}^{R}\sum_{l=1}^{N} \dot{\gamma}^r_{kl} a^T_{k,r} a_{l,r}(x_k - x_l) \\
m^a_{k,s} &= -\sum_{l=1}^{N} 2\gamma^s_{kl} m^a_{l,s} .
\end{aligned}$$

Letting $m^x_{1,k}$ equal the kth component of $U(w_1)$ and setting $m^a_{1,k,r}$ to zero provides the initial conditions for the system.

After backwards integration, the components $m^a_{0,k,r}$ contain $\nabla E(a_0)$ providing $\nabla E(w_0)$ using $w_0 = Ka_0$.

7 Experiments

We perform four sets of experiments to illustrate and test the behaviour of the kernel bundle method and its ability to support sparsity across scales. We start with a simple synthetic example to visually illustrate the differences between the single and multi-scale, sparse and non-sparse methods. In particular, will see that sparsity is achieved at the different scales individually. We then present landmark based examples of matching hand outlines to test the methods ability to represent both small and large scale features, and we illustrate the differences in the evolution in the diffeomorphism manifold when matching with LDDMM and the kernel bundle with sparse prior. Finally, we apply the method to register annotated lung CT scans, and we show that the extra capacity of the method does not affect its ability to generalize to test data; that manual scale selection is not necessary with the multi-scale method; and that we can control the sparsity across scales by varying the weight of the sparse prior.

7.1 Synthetic Example

Figure 1 presents a simple example which illustrates the effect of fusing sparsity and multiple scales. In the figure, we show the results of matching two sets of 11 points using the sparse kernel bundle method and the kernel bundle without sparse prior together with results when using sparse and non-sparse LDDMM algorithms. In all cases, we search for a diffeomorphism transporting the moving points (red) to the fixed points (black). The green points show the results of the matchings, and the red dotted lines indicate the trajectory of the moving points along the diffeomorphism path. The initial momenta $a_{0,l,r}$, $l = 1,\ldots,11$ are shown with arrows. The Gaussian kernels have scale $\sigma = 6$ for the single-scale LDDMM case and $\sigma = 12, 6, 0.8$ for the multi-scale methods in grid units as indicated by the deformed grids,

The sparse prior on LDDMM forces vanishing momentum for 4 of the 11 points. However, the fixed kernel scale has serious effect on the registration quality: the points are not quite well matched as seen by the large deviation between the landmarks and result points. The match is closer with the kernel bundle algorithm without sparse prior but all momenta at all scales are non-zero as shown by the non-zero momentum vectors and the representation is far from compact. The kernel bundle method with sparse prior obtains the best of both worlds: even with vanishing momenta for 6 of the 11 points, the match quality is comparable with non-sparse LDDMM. Of the $3 \cdot 11$ momenta, 23 vanishes. The result shows that sparsity does indeed occur across scales: point 3 and 9 from above has non-vanishing momenta at only the smallest scale, and the central point (point 6 from above) has vanishing momentum only at the midmost scale.

7.2 Hand Outlines

We consider the hand outlines shown in Figure 4. Using the landmarks (red dots) on the moving hand image, we wish to compute the kernel bundle match against the landmarks on the fixed image (black dots). The match is computed with three scales of 8, 4, and 2 units of the grid overlayed the figures. Figure 4 shows the results of computing the match with the kernel bundle together with results obtained with single-scale LDDMM with each of the three scales separately. For LDDMM with the largest scale, the match is poor and the sharp bend of the thumb is especially badly modelled. The situation improves for the middle scale though the bend of the thumb is still not sufficiently sharp and the match is bad for the middle fingers. For the smallest scales, the thumb is correctly matched but now the smaller scale is not able to model the even movement of the index finger. The kernel bundle method is by including all scales able to correctly register all the critical areas, and, at the same time, it gives the best match of the landmarks.

7.3 KB-EPDiff Across Scales

To illustrate the difference in the evolution of critical paths with the kernel bundle and LDDMM, we match in Figure 5 again eleven points (red) against eleven points (black) with results (green) using both LDDMM and kernel bundle method with two scales and enforced sparsity. In the figure, the results of the two registrations are visible in row 1 and 2 rightmost, and the evolution of the critical paths generated by the EPDiff and KB-EPDiff equations are shown with time increasing across columns. The lower rows display the deformation obtained with the kernel bundle separated for each scale. We see how most of the transport occurs at the largest scale while the lowest scale perform almost no horizontal movement but takes care of the fine adjustment allowing the kernel bundle method to obtain a good match. The sparse prior forces compactness in the representation and spatial locality of the fine scale movement.

7.4 Annotated Lung CT Registration

We now test the sparse kernel bundle on the publicly available DIR [5] dataset of lung CT images and manually annotated landmarks. We aim to show that the extra capacity of the method does not affect its ability to generalize to

[2] distance after match/distance before match.

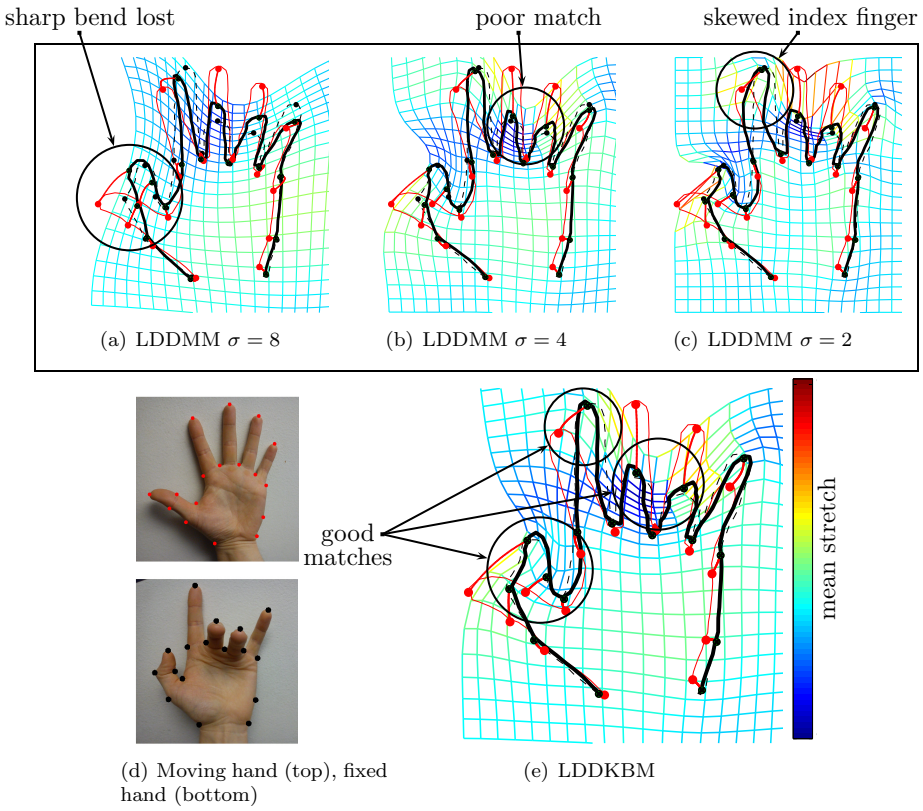


Fig. 4 Matching hands with single- and multiple scales. (d) The moving and fixed hands; (e) result of matching the hands with the kernel bundle method; (a)-(c) results of matching the hands with the single-scale LDDMM method with three different scales separately. The red landmarks of the moving hand are matched against the black landmarks of the fixed hand. The outline of the moving hand (red line) is transported to the black outline and should be compared with the outline of the fixed image (black dashed). The kernel bundle method is by incorporating movement at the multiple scales able to correctly match the critical areas on which LDDMM fails.

test data; that manual scale selection is not necessary with the multi-scale method; and that we can control the sparsity across scales by varying the weight of the sparse prior. We note important differences between the experiments performed in the conference papers [17,18]: we test with sparse priors, and we use the fast algorithm with the backwards gradient transport developed in this paper to allow cross-validation tuning of the regularization term λ in the energies (1) and (8). Thus, we are able to remove the influence of λ on the experiments. In addition, we use isotropic kernels and include more points in the experiments resulting in markedly lower test errors and more robust evaluation.

The dataset consists of five cases of CT images for different stages of the inhale and exhale phases and annotated landmarks for the maximum inhale and exhale phases, respectively. The images and landmarks are available on grids with voxel size varying slightly between the cases but close to $0.6 \times 0.6 \times 2.5$ mm. Further details can be found in the reference. For each case, the 300 publicly available landmarks, $x_1^I, \ldots, x_{300}^I$ for the maximum inhale phases and $x_1^E, \ldots, x_{300}^E$ for the maximum exhale phase, correspond pairwise. We will drive the registration using random subsets of these landmarks, and evaluate the computed match using the target registration error (TRE) measured on the landmarks not used to drive the registration.

To compare LDDMM and the sparse kernel bundle method, we choose random subsets of 200 landmarks to drive the registration, and for each such choice of subset S and each of the five patient cases, we compute the TRE $(\sum_{j \notin S} \|\varphi(x_j^I) - x_j^E\|^2)^{1/2}$. We find the relative size of the TRE against the value before the match, and average over the patients and different choices of subsets. This setup is performed for LDDMM with Gaussian kernels with scale ranging between 10mm and 170mm and with the kernel bundle method with five scales in the same range.

For each choice of random subset S, we tune the regularization term λ used in the energies (1) and (8) using cross-validation on further subsets of S. This ensures that possible variation in the effect of λ on the single- and multi-scale methods does not influence the presented results. For the kernel bundle, we let each scale contribute with equal weight. For this experiment, we let the sparsity weight be $\lambda_{sp} = 0.02$, and we let the prior vary over scale by $\lambda_{sp,r} = \lambda_{sp}/r$. The result of the experiment is not affected when λ_{sp}

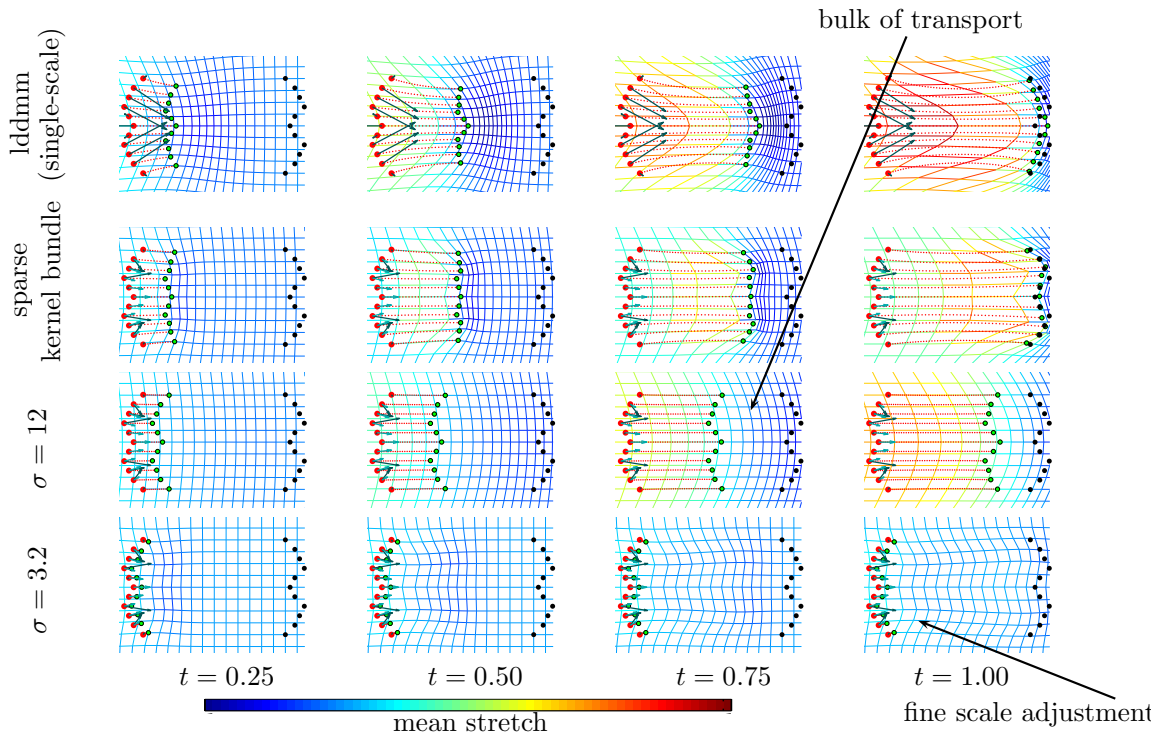


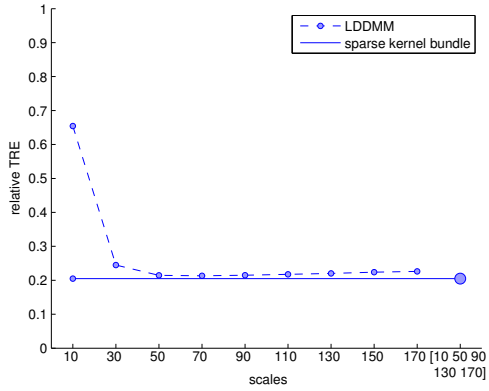
Fig. 5 LDDMM and sparse kernel bundle match of landmarks (red) to landmarks (black) with results (green). The kernels are Gaussian with the kernel bundle applying two scales. Time points of the critical paths are shown along the horizontal axis with the rightmost subfigures displaying the final deformation. (top row) Critical path determined by EPDiff equations with LDDMM (single scale); (row 2) critical path determined by KB-EPDiff equations with the sparse kernel bundle method; (row 3-4) individual contribution of each of the bundle scales (scale σ in grid units). Initially square grids are shown deformed by the diffeomorphism, and the grids are colored with the trace of Cauchy-Green strain tensor indicative of the mean stretch (log-scale for each row individually). With the sparse kernel bundle method, the largest scale contribute to most of the transport movement with smooth deformation while the smallest scale performs fine adjustment of the trajectories to obtain a good match. The sparse prior forces compactness in the representation and spatial locality of the fine scale movement.

varies within a reasonable range of the chosen value, and we further explore the choices of λ_{sp} and $\lambda_{sp,r}$ below.

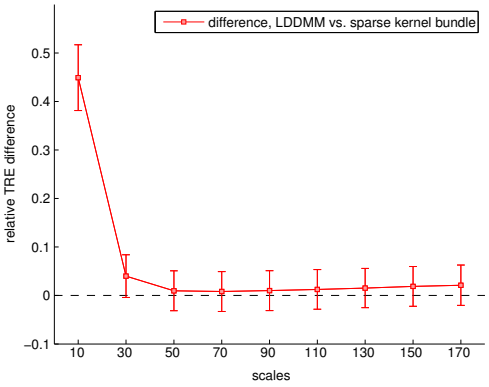
In Figure 6(a), we see that with single-scale LDDMM the TRE decreases with increasing scale up to a scale of 70mm after which it starts increasing. This indicates that a kernel scale of 70mm will be appropriate for LDDMM. As displayed in Figure 6(b), the sparse kernel bundle method attains an error lower than but within one standard deviation of the best LDDMM result. Without tuning for scale, the sparse kernel bundle method is thus as good as LDDMM, and classical scale selection by cross validation is not needed with the multi-scale method. Furthermore, the results indicate that the same quality of match can be reached with less data since we potentially could use the entire dataset to drive the registration with the kernel bundle. Manual scale selection will allow only a part of the data as input for the registration as the rest is needed to select the kernel scale. The experiment shows in addition that the extra capacity and additional degrees of freedom of the kernel bundle do not reduce the ability of the method to generalize to the test data.

To evaluate the effect of applying sparse priors with both single and multiple scales, we compare LDDMM and the kernel bundle both with sparse priors. We fix the regularization term to $\lambda = 8$ and average over all 5 patients and several randomly selected subsets of points to drive the match. We provide LDDMM with the advantage by selecting an already tuned best possible scale of $\sigma = 50$mm with these parameters, and we test against the kernel bundle method using three scales, $\sigma = 10, 50, 90$. Again, we let the scale parameters for the sparse prior vary by $\lambda_{sp,r} = \lambda_{sp}/r$. The value of $\lambda_{sp,50} = \lambda_{sp}/50$ is used for LDDMM in accordance with the choice scale.

With this setup, Figure 7 shows the connection between relative TRE^2, the sparsity weight λ_{sp}, and the fraction of momenta being non-zero after the match. As seen from the top figure, a reduction in the number of non-zero momenta of a factor 5 to 10 can be obtained for kernel bundle with only slightly increasing TRE. The sparse kernel bundle method obtains the largest reduction of non-zero parameters for a given increase in relative TRE. Sparse LDDMM still provides the smallest number of total parameters but the gap narrows as TRE increases. This fact should be viewed in the

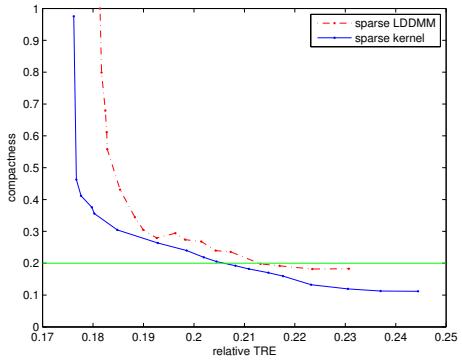


(a) LDDMM and sparse kernel bundle relative errors

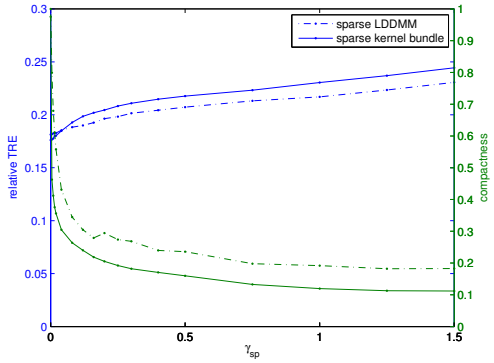


(b) Error differences

Fig. 6 LDDMM and the sparse kernel bundle method: (a) average relative TRE^2 for different kernel scales (LDDMM) and the sparse kernel bundle method (horizontal line and rightmost). Zero relative error indicates perfect match and a relative error of one indicates no error reduction. Labels on the horizontal axis are kernel scale in mm (9 different scales for LDDMM and the interval $[10\ 170]$ discretized in five scales for the sparse kernel bundle). (b) relative TRE for LDDMM (scale in mm on horizontal axis) subtracted the respective relative errors with the sparse kernel bundle again matching with the scale interval $[10\ 170]$ discretized in five scales. Positive values show superior performance of the kernel bundle method. Error bars show standard deviation of the results.



(a) Compactness as a function of relative TRE



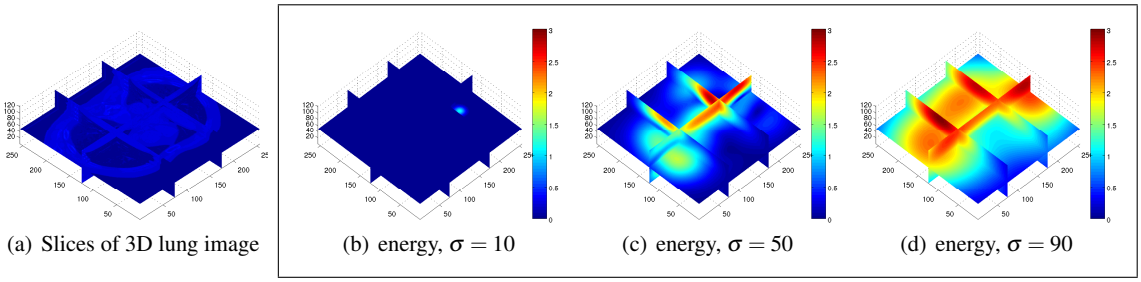
(b) Relative TRE and compactness as a function of the sparsity weight γ_{sp}

Fig. 7 Sparse LDDMM and the sparse kernel bundle method: (a) relative TRE^2 (horizontal axis) versus relative compactness (vertical axis) when the varying sparsity weight λ_{sp}; (b) relative TRE increase (left axis, blue) and relative decrease in non-zero momenta (right axis, green) as function of λ_{sp}. Results averaged over 5 patients. With a factor 5 reduction in non-zero parameters (horizontal line, top), relative TRE for sparse kernel bundle registration is 0.205 in contrast to 0.213 for sparse LDDMM.

light that the sparse LDDMM method is already tuned to the best scale, and that the kernel bundle has more degrees of freedom than LDDMM. The bottom figure shows the reduction in non-zero momenta leveling out while the relative TRE increases, though at a relatively slow pace. The absence of a sharp increase in relative TRE makes the method fairly robust the actual choice of λ_{sp}.

The weighting of the sparsity parameter across scales can be controlled by letting $\lambda_{sp,r} = \lambda_{sp}/r^{\alpha}$ and varying α. To explore this and the resulting cross-scale effects, we select $\lambda_{sp} = 0.05$, and plot the relative TRE against α in Figure 8. In addition, the figure shows how the distribution of non-zero parameters at the different scales varies with α. The increased penalty at small scales for $\alpha > 1$ and corresponding increased penalty for large scales for $\alpha < 1$ is clearly visible. Indeed, the difference in the number of non-zero parameters at the different scales shows that sparsity across scales is achieved.

To illustrate the result of one lung registration with the sparse kernel bundle method, Figure 9 shows the energy of the initial velocity field for the three bundle scales separately. The uniform spread of the velocity provided by the large scale kernels results in a smooth deformation even with only 20% percent non-zero momenta at that scale. The localized deformation field provided by the sparsity of the smaller momenta is in addition clearly visible.



(a) Slices of 3D lung image (b) energy, $\sigma = 10$ (c) energy, $\sigma = 50$ (d) energy, $\sigma = 90$

Fig. 9 Slices of 3D lung image and the multi-scale initial vector field at three scales that when combined generate the sparse kernel bundle registration. Left to right: (a) slices of CT image, (b)-(d) squared L^2-norm of the components at each of the three scales $\sigma = 10, 50, 90$ which in combination make up the multi-scale bundle vector w_0 generating φ at $t = 0$. The uniform spread of the velocity provided by the large scale kernels results in a smooth flow even with only 20% percent non-zero momenta for that scale. The localized deformation field provided by the sparsity of the smaller momenta is in addition clearly visible.

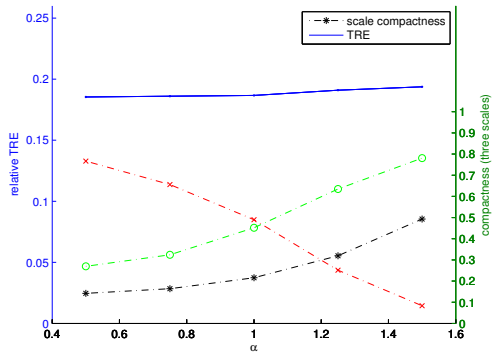


Fig. 8 Sparsity across scales: relative TRE^2 (left axis, blue) and relative decrease in non-zero momenta for each scale (right axis, red (x)/green (o)/black (*)) as function of the scale weighting α averaged over the 5 patients. The red (x) marked curve shows compactness for the smallest scale ($\sigma = 10$), the green (o) marked curves for the midmost scale ($\sigma = 50$) and the black (*) marked curves for the largest scale ($\sigma = 90$). The TRE and total number of non-zero parameters stay relatively constant though the distribution of non-zero parameters over scale varies. In particular, the figure shows that sparsity across scales is achieved.

8 Conclusion and Outlook

The multi-scale kernel bundle framework extends the LDDMM framework by incorporating multiple kernels at multiple scales in the registration. The method allows representing deformation at multiple scales at different spatial locations and thereby increases the capacity of the deformation description while supporting application of sparse priors that ensures compact representation. Since the priors are applied independently across the parts of the bundle, the algorithm allows sparsity across scales, and the multiple scales extend the range of deformation the algorithm is able to model significantly. The method may as well be applied to images in the finite dimensional setting promising similar results and to group-wise registration extending the pairwise experiments presented here.

We visually illustrate the method on synthetic data and show the obtained sparsity across scales. We show the multi-scale effects and cross-scale evolution on additional examples. In addition, when applying the method to a dataset of annotated lung CT images, we demonstrate that the method removes the need for classical scale selection; that sparsity across scales is achieved; that the sparsity may be achieved with only minor increase in registration error; and that the extra capacity of the algorithm does not affect generalization ability.

In addition to the applications demonstrated in this paper, we expect the sparse kernel bundle method to be particularly powerful when applied to population analysis of e.g. atrophy during Alzheimer's disease. From both a theoretical and a practical point of view, the sparse kernel bundle framework provides a compact representation of deformation across scales.

References

1. Beg, M.F., Miller, M.I., Trouvé, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. IJCV **61**(2), 139–157 (2005)
2. Bruveris, M., Gay-Balmaz, F., Holm, D.D., Ratiu, T.S.: The momentum map representation of images. 0912.2990 (2009). URL `http://arxiv.org/abs/0912.2990`
3. Bruveris, M., Risser, L., Vialard, F.: Mixture of kernels and iterated Semi-Direct product of diffeomorphism groups. arXiv:1108.2472 (2011). URL `http://arxiv.org/abs/1108.2472`
4. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted l1 minimization. Journal of Fourier Analysis and Applications **14**, 877–905 (2008). DOI 10.1007/s00041-008-9045-x
5. Castillo, R., Castillo, E., Guerra, R., Johnson, V.E., McPhail, T., Garg, A.K., Guerrero, T.: A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. Physics in Medicine and Biology **54**(7), 1849–1870 (2009). DOI 10.1088/0031-9155/54/7/001
6. Christensen, G., Rabbitt, R., Miller, M.: Deformable templates using large deformation kinematics. Image Processing, IEEE Transactions on **5**(10) (2002)
7. Dupuis, P., Grenander, U., Miller, M.I.: Variational problems on flows of diffeomorphisms for image matching (1998)

8. Durrleman, S., Prastawa, M., Gerig, G., Joshi, S.: Optimal data-driven sparse parameterization of diffeomorphisms for population analysis. Information Processing in Medical Imaging: Proceedings of the ... Conference **22**, 123–134 (2011). PMID: 21761651
9. Fasshauer, G.E., Ye, Q.: Reproducing kernels of generalized sobolev spaces via a green function approach with distributional operators. Numerische Mathematik (2011). DOI 10.1007/s00211-011-0391-2
10. Grenander, U.: General Pattern Theory: A Mathematical Study of Regular Structures. Oxford University Press, USA (1994)
11. Joshi, S.: Optimal data-driven sparse parameterization of diffeomorphisms for population analysis. Presentation at the Geometry for Anatomy workshop, Banff, Canada (2011)
12. Risser, L., Vialard, F., Wolz, R., Holm, D.D., Rueckert, D.: Simultaneous fine and coarse diffeomorphic registration: application to atrophy measurement in alzheimer's disease. MICCAI 2010 **13**(Pt 2), 610–617 (2010). PMID: 20879366
13. Risser, L., Vialard, F.X., Wolz, R., Murgasova, M., Holm, D.D., Rueckert, D.: Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping. IEEE Transactions on Medical Imaging **30**(10), 1746–1759 (2011). DOI 10.1109/TMI.2011.2146787
14. Sommer, S.: Accelerating Multi-Scale flows for LDDKBM diffeomorphic registration. In: GPUCV workshop at ICCV 2011. IEEE (2011)
15. Sommer, S., Lauze, F., Nielsen, M., Pennec, X.: Kernel bundle EPDiff: evolution equations for Multi-Scale diffeomorphic image registration. In: SSVM 2011. Springer (2011)
16. Sommer, S., Nielsen, M., Darkner, S., Pennec, X.: Higher order kernels and locally affine LDDMM registration. Submitted. (2011). URL `http://arxiv.org/abs/1112.3166`
17. Sommer, S., Nielsen, M., Lauze, F., Pennec, X.: A Multi-Scale kernel bundle for LDDMM: towards sparse deformation description across space and scales. In: IPMI 2011. Springer (2011)
18. Sommer, S., Nielsen, M., Pennec, X.: Sparsity and scale: Compact representations of deformation for diffeomorphic registration. In: MMBIA at WACV 2012 (2012)
19. Trouvé, A.: An infinite dimensional group approach for physics based models in patterns recognition (1995)
20. Vaillant, M., Miller, M., Younes, L., Trouvé, A.: Statistics on diffeomorphisms via tangent space representations. NeuroImage **23**(Supplement 1), S161–S169 (2004). DOI 10.1016/j.neuroimage.2004.07.023
21. Younes, L.: Shapes and Diffeomorphisms. Springer (2010)
22. Younes, L., Arrate, F., Miller, M.I.: Evolutions equations in computational anatomy. NeuroImage **45**(1, Supplement 1), S40–S50 (2009). DOI 10.1016/j.neuroimage.2008.10.050

3.

Paper #2: *Higher Order Kernels and Locally Affine LDDMM Registration*

Paper submitted to SIAM Journal on Imaging Sciences, December 2011.

Authors:
Stefan Sommer, Mads Nielsen, Sune Darkner, and Xavier Pennec

Notes:

The kernel bundle framework (Paper #1) paves some of the way towards sparse representations of deformation by introducing scale in the framework. However, sparse representations with LDDMM and the kernel bundle framework will only represent translational movement at each *deformation atom*, and the modeling capacity is therefore still limited. In this paper, we introduce *higher order kernels* in the LDDMM framework. The new kernels allow modeling of non-translational movement and compact description of locally affine deformations. We show how the singular momenta of the higher order kernels make them fit nicely into the LDDMM framework, and we derive the EPDiff equations with higher order kernels. The kernels and their representation power is evaluated on several examples including MR scans of patients suffering from Alzheimer's disease.

HIGHER ORDER KERNELS AND LOCALLY AFFINE LDDMM REGISTRATION

STEFAN SOMMER*, MADS NIELSEN*,†, SUNE DARKNER*, AND XAVIER PENNEC‡

Abstract. To achieve sparse description that allows intuitive analysis, we aim to represent deformation with a basis containing *interpretable* elements, and we wish to use elements that have the description capacity to represent the deformation *compactly*. We accomplish this by introducing *higher order kernels* in the LDDMM registration framework. The kernels allow local description of affine transformations and subsequent compact description of non-translational movement and of the entire non-rigid deformation. This is obtained with a representation that contains directly interpretable information from both mathematical and modeling perspectives. We develop the mathematical construction behind the higher order kernels, we show the implications for *sparse* image registration and deformation description, and we provide examples of how the capacity of the kernels enables registration with a very low number of parameters. The capacity and interpretability of the kernels lead to natural modeling of articulated movement, and the kernels promise to be useful for quantifying ventricle expansion and progressing atrophy during Alzheimer's disease.

Key words. LDDMM, diffeomorphic registration, RHKS, kernels, momentum, computational anatomy

AMS subject classifications. 65D18, 65K10, 41A15

1. Introduction.

Atrophy occurs in the human brain among patients suffering from Alzheimer's disease, and the progressing atrophy can be detected by the expansion of the ventricles [16, 13]. We wish to describe the deformation of the brain caused by the progressing disease using as few parameters as possible and with a representation which allows intuitive analysis: we search for *sparse* representations with basis elements that have the *capacity* to describe deformation with few parameters while being directly *interpretable*.

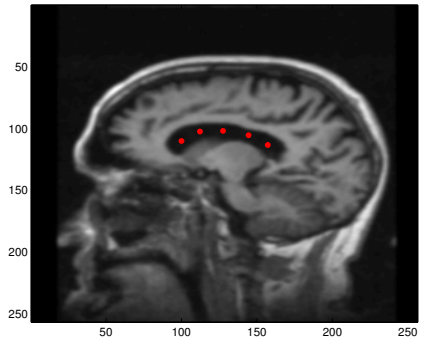
Image registration algorithms often represent translational movement in a dense sampling of the image domain. Such approaches fail to satisfy the above goals: low dimensional deformations such as expansion of the ventricles will not be represented sparsely; the registration algorithm must optimize a large number of parameters; and the expansion cannot easily be interpreted from the registration result.

In this paper, we introduce *higher order kernels* in the LDDMM registration framework to obtain a deformation representation promising *sparsity*, increased *capacity*, end *interpretability*. We show how higher order kernels allow local representation of affine transformations and that they increase the capacity of the representation at each point. We use the compact deformation description to register points and images using very few parameters, and we illustrate how the deformation coded by the kernels can be directly interpreted and that it represents information directly useful in applications: with low numbers of control points, we can detect the expanding ventricles of the patient shown in in Figure 1.1.

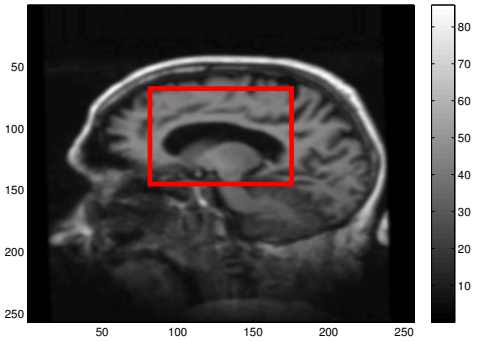
1.1. Background.

Among the many methods for non-rigid registration in medical imaging, the majority model the displacement of each spatial position by either a combination of suitable basis functions for the displacement itself or for the velocity of the voxels. The number of control points vary between one for each voxel [2, 15, 7] and

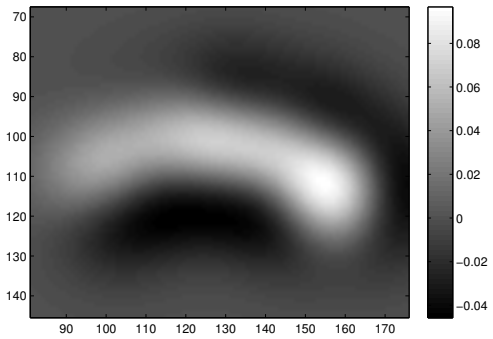
*Dept. of Computer Science, Univ. of Copenhagen, Denmark (sommer@diku.dk)
†BiomedIQ, Copenhagen, Denmark
‡Asclepios Project-Team, INRIA Sophia-Antipolis, France



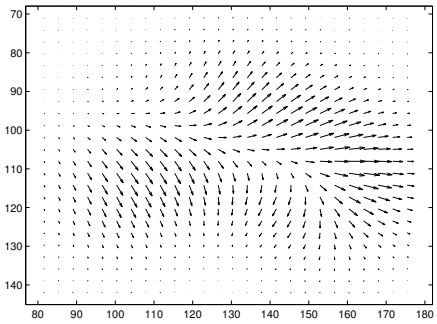
(a) Baseline with control points.



(b) Follow up (box marking zoom area, figure (c) and (d)).



(c) log-Jacobian in ventricle area.



(d) Initial deformation field in ventricle area.

FIG. 1.1. *Progressing Alzheimer's disease cause atrophy and expansion of the ventricles. By placing five* deformation atoms *in the 2D MRI slices [17] and by using* higher order kernels*, we can register the expansion. (a) The position of the deformation atoms shown in the baseline scan; (b) the follow up scan; (c) the* log*-Jacobian determinant of the generated deformation in the ventricle area (red box in (b)); (d) the vector field at* $t = 0$ *of the generated deformation. The logarithm of the Jacobian determinant and the divergence at the deformation atoms are positive which is in line with the expected ventricle expansion, confer also Figure 5.5.*

fewer with larger basis functions [22, 5, 11]. For all methods, the infinite-dimensional space of deformations is approximated by the finite- but high-dimensional subspace spanned by the parametrization of the individual method. The approximation will be good if the underlying deformation is close to this subspace, and the representation will be compact, if few basis functions describe the deformation well. The choice of basis functions play a crucial role, and we will in the rest of the paper denote them *deformation atoms*. Two main observations constitute the motivation for the work presented in this paper:

Observation 1: Order of the Deformation Model. In the majority of registration methods, the deformation atoms model the local *translation* of each point. We wish a richer representation which is in particular able to model locally linear components in addition to local translations. The Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks pursue this by representing the velocity of a path of deformations locally by matrix logarithms. Ideas from the Polyaffine methods have recently been incorporated in e.g. the Demons algorithm [29] but, to the best of our knowledge, not

in the LDDMM registration framework. We wish to extend the set of deformation atoms used in LDDMM to allow representation of *first* and *higher order* structure and hence incorporate the benefits of the Polyaffine methods in the LDDMM framework.

Observation 2: Order of the Similarity Measure. When registering DT images, the reorientation is a function of the derivative of the warp; curve normals also contain directional information which is dependent on the warp derivative and airway trees contain directional information in the three structure which can be used for measuring similarity. These are examples of similarity measures containing *higher order information*. For the case of image registration, the warp derivative may also enter the equation either directly in the similarity measure [21, 19] or to allow use of more image information than provided by a sampling of the warp. Consider an image similarity measure on the form $U(\varphi) = \int_\Omega F(I_m(\varphi^{-1}(x)), I_f(x))dx$. A finite sampling of the domain Ω can approximate this with

$$\tilde{U}^0(\varphi) = \frac{1}{N}\sum_{i=1}^{N} F(I_m(\varphi^{-1}(x_i)), I_f(x_i)) \ .$$

Letting $\{p_1, \ldots, p_P\}$ be uniformly distributed points around 0, we can increase the amount of image information used in $\tilde{U}^0(\varphi)$ *without* additional sampling of the warp by using a first order approximation of φ^{-1}:

$$\tilde{U}^1(\varphi) = \frac{1}{NP}\sum_{i=1}^{N}\sum_{j=1}^{N} F(I_m(D\varphi^{-1}p_j + \varphi^{-1}(x_i)), I_f(p_j + x_i)) \ .$$

This can be considered an increase from *zero* to *first order* in the approximation of U. Besides including more image information than provided by the initial sampling of the warp, the increase in order allows capture of non-translational information - e.g. rotation and dilation - in the similarity measure. The approach can be seen as a specific case of similarity smoothing and more examples of smoothing in intensity based image registration can be found in [9][1].

We focus on deformation modeling with the Large Deformation Diffeomorphic Metric Mapping (LDDMM) registration framework which has the benefit of both providing good registrations and drawing strong theoretical links with Lie group theory and evolution equations in physical modeling [8, 32]. Most often, high-dimensional voxel-wise representations are used for LDDMM although recent interest in *compact* representations [11, 25] show that the number of parameters can be much reduced. These methods use interpolation of the velocity field by deformation atoms to represent translational movement but deformation by other parts of the affine group cannot be compactly represented.

The deformation atoms are in LDDMM called *kernels*. The kernels are centered at different spatial positions and parameters determine the contribution of each kernel. In this paper, we use the partial derivative reproducing property [33] to show that partial derivatives of kernels - *higher order kernels* - fit naturally in the LDDMM framework and constitute deformation atoms along with the original kernels. In particular, the higher order kernels have a singular momentum and the momentum stay singular when transported by the EPDiff evolution equations. We show how the

[1] An updated version of [9] is available at `http://diku.dk/english/staff/?id=383640&f=3&vis=medarbejder`.

higher order kernels allow modeling of locally affine deformations and hence extend the capacity of sparsely discretized LDDMM methods. In addition, they comprise the natural vehicle for incorporating first order similarity measures in the framework.

1.2. Related Work. A number of methods for non-rigid registration have been developed during the last decades including non-linear elastic methods [18], parametrizations using static velocity fields [2, 15], the demons algorithm [26, 29], and spline-based methods [22, 5]. For the particular case of LDDMM, the groundbreaking work appeared with the deformable template model by Grenander [14] and the flow approach by Christensen et al. [7] together with the theoretical contributions of Dupuis et al. and Trouvé [10, 27]. Algorithms for computing optimal diffeomorphisms have been developed in [4], and [28] uses the momentum representation for statistics and develops a momentum based algorithm for the landmark matching problem.

Locally affine deformations can be modeled using the Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks. The velocity of a path of deformations is here computed using matrix logarithms, and the resulting diffeomorphism flowed forward by integrating the velocity. Ideas from the Polyaffine methods have recently been incorporated in e.g. the Demons algorithm [29, 23]. In LDDMM, the deformation atoms, the kernels, represent translational movement and the non-translational part of affine transformations cannot directly be represented. We will show how higher order kernels constitute deformation atoms which allow representing the linear parts of affine transformations. From a mathematical points of view, this is possible due to the partial derivative reproducing property (Zhou [33]). The partial derivative reproducing property has been used in [6] to derive variations of flow equations for LDDMM DTI registration but higher order kernels are not used in the parametrization. Confer the monograph [32] for information on RKHSs and their role in LDDMM.

In order to reduce the dimensionality of the parametrization used in LDDMM, Durrleman et al. [11] introduced a control point formulation of the registration problem by choosing a finite set of control points and constraining the momentum to be concentrated as Dirac measures at the point trajectories. As we will see, higher order kernels make a finite control point formulation possible which is different in important aspects. Younes [31] in addition considers evolution in constrained subspaces.

Higher order kernels increase the capacity of the deformation parametrization, a goal which is also treated in sparse multi-scale methods such as the kernel bundle framework [25]. This method concerns the size of the kernel in contrast to the order which we deal with here. As we will discuss in the experiments section, the size of the kernel is important for higher order kernels as well, and higher order kernels and the kernel bundle method will likely complement each other nicely if applied together.

1.3. Content and Outline. We start the paper with an overview of LDDMM registration and the mathematical constructs behind the method. In the following section, we motivate the introduction of higher order kernels using zero- and first-order similarity measure approximations. We describe the derivative reproducing property, and show how it implies singular momentum for the kernels. The evolution of the momentum and velocity fields governed by the EPDiff evolution equations are then determined. To make actual registration possible, the next section describes the effect of varying the initial conditions and the backwards gradient transport before developing the actual matching algorithm. We give examples in the second last section, and we show how the higher order kernels are particularly useful when registering human brains with progressing atrophy. The paper ends with concluding remarks and outlook. The paper thus contributes by

(1) introducing higher order kernels in the LDDMM framework as the deformation atoms enabling locally affine transformations,
(2) showing how the order of the similarity measure approximation relates to higher order kernels,
(3) relating the derivative reproducing property to LDDMM and showing how it implies a singular momentum for the higher order kernels,
(4) deriving the EPDiff transport equations for higher order kernels,
(5) computing the forward variational equations and describing the backwards gradient transport,
(6) developing an algorithm allowing matching with the higher order kernels,
(7) and demonstrating the application of the kernels with registration examples.

2. LDDMM Registration, Kernels, and Evolution Equations. We here give a brief introduction to LDDMM registration. For further information, confer the monograph [32] with extensive information on the method.

In the LDDMM framework, registration is performed through the action of diffeomorphisms on geometric objects. This approach is very general and allows the framework to be applied to both landmarks, curves, surfaces, images, and tensors. In the case of images, the action of a diffeomorphism φ on the image $I : \Omega \to \mathbb{R}$ takes the form $\varphi.I = I \circ \varphi^{-1}$, and given a fixed image I_f and moving image I_m, the registration amounts to a search for φ such that $\varphi.I_m \sim I_f$. In exact matching, we wish $\varphi.I_m$ be exactly equal to I_f but, more frequently, we allow some amount of inexactness to account for noise in the images and allow for smoother diffeomorphisms. This is done by defining a similarity measure $U(\varphi) = U(\varphi.I_m, I_f)$ on images and a regularization measure E_1 to give a combined energy

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi.I_m, I_f) \ . \tag{2.1}$$

Here λ is a positive real representing the trade-off between regularity and goodness of fit. The similarity measure U is in the simplest form the L^2-error $\int_\Omega |\varphi.I_m(x) - I_f(x)|dx$ but more advanced measures can be used (e.g. [20, 30, 9]).

In order to define the regularization term E_1, we introduce some notations in the following: Let the domain Ω be a subset of $\mathbb{R}^d$ with $d = 2, 3$, and let V denote a Hilbert space of vector fields $v : \Omega \to \mathbb{R}^d$ such that V with associated norm $\|\cdot\|_V$ is included in $L^2(\Omega, \mathbb{R}^d)$ and admissible [32, Chap. 9], i.e. sufficiently smooth. Given a time-dependent vector field $t \mapsto v_t$ with

$$\int_0^1 \|v_t\|_V^2 \, dt < \infty \tag{2.2}$$

the associated differential equation $\partial_t \varphi_t = v_t \circ \varphi_t$ has with initial condition φ_s a diffeomorphism φ_{st}^v as unique solution at time t. The set G_V of diffeomorphisms built from V by such differential equations is a Lie group, and V is its tangent space at each point. The inner product on V associated to a norm $\|\cdot\|_V$ makes G_V a Riemannian manifold with right-invariant metric. Setting $\varphi_{00}^v = Id_\Omega$, the map $t \mapsto \varphi_{0t}^v$ is a path from Id_Ω to φ with energy given by (2.2) and generated by v_t. We will use this notation extensively in the following. A critical path for the energy (2.2) is a geodesic on G_V, and the regularization term E_1 is defined using the energy by

$$E_1(\varphi) = \min_{v_t \in V, \varphi_{01}^v = \varphi} \int_0^1 \|v_s\|_V^2 \, ds \ , \tag{2.3}$$

i.e. it measures the minimal energy of diffeomorphism paths from Id_Ω to φ. Since the energy is high for paths with great variation, the term penalizes highly varying paths, and a low value of $E_1(\varphi)$ thus implies that φ is regular.

2.1. Kernel and Momentum. As a consequence of the assumed admissibility of V, the evaluation functionals $\delta_x : v \mapsto v(x) \in \mathbb{R}^d$ is well-defined and continuous for any $x \in \Omega$. Thus, for any $a \in \mathbb{R}^d$ the map $a \otimes \delta_x : v \mapsto a^T v(x)$ belongs to the topological dual V^* consisting of the continuous linear maps of V. This in turn implies the existence of spatially dependent matrices $K : \Omega \times \Omega \to \mathbb{R}^{d\times d}$, the *kernel*, such that, for any constant vector $a \in \mathbb{R}^d$, the vector field $K(\cdot, x)a \in V$ represents $a \otimes \delta_x$ and $\langle K(\cdot, x)a, v\rangle_V = a \otimes \delta_x(v)$ for any $v \in V$, point $x \in \Omega$ and vector $a \in \mathbb{R}^d$. This latter property is denoted the reproducing property and gives V the structure of a reproducing kernel Hilbert space (RKHS). Tightly connected to the norm and kernels is the notion of *momentum* given by the linear momentum operator $L : V \to V^* \subset L^2(\Omega, \mathbb{R}^d)$ which satisfies

$$\langle Lv, w\rangle_{L^2(\Omega,\mathbb{R}^d)} = \int_\Omega \big(Lv(x)\big)^T w(x)dx = \langle v, w\rangle_V$$

for all $v, w \in V$. The momentum operator connects the inner product on V with the inner product in $L^2(\Omega, \mathbb{R}^d)$, and the image Lv of an element $v \in V$ is denoted the momentum of v. The momentum Lv might be singular and in fact $L\big(K(\cdot, y)a\big)(x)$ is the Dirac measure $\delta_y(x)a$. Considering K as the map $a \mapsto \int_\Omega K(\cdot, x)a(x)dx$, L can be viewed as the inverse of K. Confer [32] for a thorough introduction to reproducing kernels, especially with a view towards the LDDMM framework.

Instead of deriving the kernel from V, the opposite approach can be used: build V from a kernel, and hence impose the regularization in the framework from the kernel. With this approach, the kernel is often chosen to ensure rotational and translational invariance [32] and the scalar Gaussian kernel $K(x, y) = \exp(\frac{\|x-y\|^2}{\sigma^2})\mathrm{Id}_d$ is an often used choice. Confer [12] for details on the construction of V from Gaussian kernels.

2.2. Optimal Paths: The EPDiff Evolution Equations. The relation between norm and momentum lead to convenient equations for minimizers of the energy (2.1). In particular, the EPDiff equations for the evolution of the momentum a_t for optimal paths assert that if φ_t is a path minimizing $E_1(\varphi)$ with $\varphi_1 = \varphi$ minimizing $E(\varphi)$ and v_t is the derivative of φ_t then v_t satisfies the system

$$\begin{aligned} v_t &= \int_\Omega K(\cdot, x)a_t(x)dx \ , \\ \frac{d}{dt}a_t &= -Da_t v_t - a_t\nabla \cdot v_t - (Dv_t)^T a_t \end{aligned}$$

with Da_t and Dv_t denoting spatial differentiation of the momentum and velocity fields, respectively. The first equation connects the momentum a_t with the velocity v_t, and the second equation describes the time evolution of the momentum. In the most general form, the EPDiff equations describe the evolution of the momentum using the adjoint map. Following [32], we define $\mathrm{Ad}_\varphi v(x) = (D\varphi\, v) \circ \varphi^{-1}(x)$ for $v \in V$ and get a functional Ad^*_φ on the dual V^* of V by $(\mathrm{Ad}^*_\varphi \rho|v) = (\rho|\mathrm{Ad}_\varphi(v))$.[2] Define in addition $\mathrm{Ad}^T_\varphi v = K(\mathrm{Ad}^*_\varphi(Lv))$ which then satisfies $\left\langle \mathrm{Ad}^T_\varphi v, w \right\rangle = (\mathrm{Ad}^*_\varphi(Lv)|w)$, and let $\nabla_\varphi U$

[2]Here and in the following, we will use the notation $(p|v) := p(v)$ for evaluation of the functional $p \in V^*$ on the vector field $v \in V$.

denote the gradient of the similarity measure U with respect to the inner product on V so that $\langle \nabla_\varphi U, v\rangle_V = \partial_\epsilon U(\psi^v_{0\epsilon} \circ \varphi)$ for any variation $v \in V$ and diffeomorphism path $\psi^v_{0\epsilon}$ with derivative v. For optimal paths v_t, the EPDiff equations assert that $v_t = \mathrm{Ad}^T_{\varphi^v_{t1}} v_1$ with $v_1 = -\frac{1}{2}\nabla_{\varphi^v_{01}} U$ which leads to the conservation of momentum property for optimal paths. Conversely, the EPDiff equations reduce to simpler forms for certain objects. For landmarks $x_1, \ldots, x_N$, the momentum will be concentrated at point trajectories $x_{t,i} := \varphi_t(x_i)$ as Dirac measures $a_{t,i}\delta_{x_{t,i}}$ leading to the finite dimensional system of ODE's

$$\begin{aligned} v_t &= \sum_{l=1}^{N} K(\cdot, x_{t,l}))a_{t,l} \ , \quad \frac{d}{dt}\varphi_t(x_i) = v_t(x_{t,i}) \ , \\ \frac{d}{dt}a_{t,i} &= -\sum_{l=1}^{N} D_1 K(x_{t,i}, x_{t,l})a^T_{t,i}a_{t,l} \ . \end{aligned} \tag{2.4}$$

3. Higher Order Kernels. We here introduce higher order kernels in the LDDMM registration framework. We start by motivating the construction by considering the approximation used when computing the similarity measure. We then link the kernels to the momentum using the derivative reproducing property, and derive the path energy. We consider locally affine transformations before deriving the EPDiff evolution equations for paths incorporating higher order kernels.

We will motivate the introduction of higher order kernels by considering a specific case of image registration: we take on the goal of using a control point formulation [11] when solving the registration problem (2.1) and hence aim for using a relatively sparse sampling of the velocity or momentum field. To achieve this, we will consider the coupling between the transported control points $\{\varphi^{-1}(x_1), \ldots, \varphi^{-1}(x_N)\}$ and the similarity measure in order to ensure the momentum stays singular and localized at the point trajectories while removing the need for warping the entire image at every iteration of the optimization process.

Considering a similarity measure $U(\varphi) = \int_\Omega F(I_m(\varphi^{-1}(x)), I_f(x))dx$ as discussed in the introduction, and a finite discretization $\tilde{U}^0(\varphi) = 1/N \sum_{i=1}^N F(\varphi.I_m(x_i), I_f(x_i))$ with a sparse set of control points $\{x_i\}$. While using $\tilde{U}^0(\varphi)$ to drive registration of the images will be very efficient in evaluating the warp in few points, it will suffer correspondingly from only using image information present in those points. Apart from not being robust under the presence of noise in the images, the discretization implies that local dilation or rotation around the points $\varphi^{-1}(x_i)$ cannot be detected: any variation $v \in V$ of φ keeping $\varphi^{-1}(x_i)$ constant for all $i = 1, \ldots, N$ will not change $\tilde{U}^0(\varphi)$. Formally, if $\psi_{0\epsilon}$ is a diffeomorphism path that is equal to φ at $t = 0$ and has derivative v at $t = 0$, i.e. $\partial_\epsilon \psi_{0\epsilon} = v$ and $\psi_{00} = \varphi$, then

$$\partial_\epsilon F(\psi_{0\epsilon}.I_m(x_i), I_f(x_i)) = \partial_1 F(\varphi.I_m(x_i), I_f(x_i)) \cdot \left(\nabla_{\varphi^{-1}(x_i)} I_m\right)^T v(\varphi^{-1}(x_i))$$

which vanishes if $v(\varphi^{-1}(x_i)) = 0$. Here $\partial_1 F$ denotes the derivative of $F : \mathbb{R}^2 \to \mathbb{R}$ with respect to the first variable.

A simple way to include more image in formation in the similarity measure is to convolve with a kernel K_s, and thus extend $\tilde{U}^0$ to

$$U^1(\varphi) = \frac{1}{N}\sum_{i=1}^{N} c_{K_s} \int_\Omega K_s(p + x_i, x_i) U(\varphi.I_m(p + x_i), I_f(p + x_i))dp$$

with c_{K_s} a normalization constant. If K_s is a box kernel, this amounts to a finer sampling of both the image and warp, and hence a finer discretization of the Riemann integral. The kernel K_s should not be confused with the RKHS kernel connected to the norm on V that is used when generating the V-gradient. A Gaussian kernel may be used for K_s [9], and more information on using smoothing kernels for intensity based image registration can be found in [9, 34].

The measure $U^1(\varphi)$ is problematic since a variation of φ would affect not only the point $\varphi^{-1}(x_i)$ but also $\varphi.I_m(p + x_i)$, and $U^1(\varphi)$ will therefore be dependent on $\varphi.I_m(p+x_i)$ for any p where $K_s(p, x_i)$ is non-zero. In this situation, the momentum is no longer concentrated in Dirac measures located at $\varphi_t^{-1}(x_i)$, and it will be necessary to increase the sampling of the warp. However, a first order expansion of φ^{-1} yields the approximation

$$\tilde{U}^1(\varphi) = \frac{1}{N}\sum_{i=1}^{N} c_{K_s} \int_\Omega K_s(p+x_i, x_i) U(I_m(D_{x_i}\varphi^{-1}p+\varphi^{-1}(x_i)), I_f(p+x_i))dp \ . \quad (3.1)$$

The measure $\tilde{U}^1(\varphi)$ is now again local depending only on $\varphi^{-1}(x_i)$ and the first order derivatives $D_{x_i}\varphi^{-1}$. It offers the stability provided by the convolution with K_s, and, importantly, variations v of φ keeping $\varphi^{-1}(x_i)$ constant but changing $D_{x_i}\varphi^{-1}$ do indeed affect the similarity measure. This implies that $\tilde{U}^1(\varphi)$ is able to catch rotations and dilations and drive the search for optimal φ accordingly. Please note the differences with the approach of Durrleman et al. [11]: when using $\tilde{U}^1(\varphi)$ as outlined here, the need for flowing the entire moving image forward is removed and the momentum field will stay singular *directly* thus removing the need for constraining the form of the velocity field.

This raises the question of how to represent variations of $D\varphi$ in the LDDMM framework. As we will see, higher order kernels appear as the natural choice of deformation atoms allowing singular momentum for variations of $D\varphi$ and hence keeping the benefits of the finite control point formulation.

3.1. Derivative Reproducing Property. Recall the reproducing property of the RKHS structure, i.e. $\langle K(\cdot, x)a, v\rangle_V = a \otimes \delta_x(v)$ for $v \in V$, $x \in \Omega$ and $a \in \mathbb{R}^d$. Zhou [33] shows that this property holds not only for the kernel but also for its partial derivatives. Letting $D_x^\alpha v$ denote the derivative of $v \in V$ at $x \in \Omega$ with respect to the multi-index α,

$$D_x^\alpha v = \frac{\partial^{|\alpha|}}{\partial_{x^1}^{\alpha_1} \dots \partial_{x^q}^{\alpha_q}} v(x)$$

and defining $(D_x^\alpha Ka)(y) = D_x^\alpha(K(\cdot, y)a)$ for $a \in \mathbb{R}^d$, Zhou proves that $D_x^\alpha Ka \in V$ and that the *partial derivative reproducing property*

$$\langle D_x^\alpha Ka, v\rangle_V = a^T D_x^\alpha(v) \quad (3.2)$$

holds when the maps in V are sufficiently smooth for the derivatives to exist. In the following, we denote the matrices $D_x^\alpha K$ *higher order kernels*. Similarly, we denote the maps $a \otimes D_x^\alpha : V \to \mathbb{R}$ defined by $a \otimes D_x^\alpha(v) := a^T D_x^\alpha v$ *higher order Diracs*. It follows that

$$a \otimes D_x^\alpha = \left(v \mapsto \langle D_x^\alpha Ka, v\rangle_V\right) \in V^* \ .$$

As a consequence of Zhous result, we can derive the momentum for the higher order kernels. Recall that the momentum map $L : V \to V^*$ satisfies $\langle Lv, w\rangle_{L^2} = \langle v, w\rangle_V$. With the higher order kernels,

$$\langle LD_x^\alpha Ka, v\rangle_{L_2} = \langle D_x^\alpha Ka, v\rangle_V = a \otimes D_x^\alpha(v) = \langle a \otimes D_x^\alpha, v\rangle_{L^2} \ .$$

Thus $LD_x^\alpha Ka = a \otimes D_x^\alpha$ or, shorter, $LD_x^\alpha K = D_x^\alpha$. That is, the higher order kernels and higher order Diracs corresponds just as the kernels and Diracs in the usual RKHS sense.

Consider a map on diffeomorphisms $U : G_V \to \mathbb{R}$ e.g. an image similarity measure dependent on φ. In a finite dimensional setting with N evaluation points x_i, U would decompose as $U(\varphi) = P \circ Q(\varphi)$ with $Q(\varphi) = (\varphi(x_1), \ldots, \varphi(x_N))$ and $P : \mathbb{R}^{dN} \to \mathbb{R}$. Introducing higher order kernels, we let $Q(\varphi) = (D_{x_1}^{\alpha_1}(\varphi), \ldots, D_{x_N}^{\alpha_J}(\varphi))$ with J multi-indices α_j, and decompose U as $U(\varphi) = P \circ Q(\varphi)$ with $P : \mathbb{R}^{dNJ} \to \mathbb{R}$. We allow α_j to be empty and hence incorporate the standard zero-order case. The partial derivative reproducing property now allows to compute the V-gradient of U as a sum of higher order kernels.

PROPOSITION 3.1. *Let $\nabla^{ij} P$ denote the gradient with respect to the variable indexed by $D_{x_i}^{\alpha_j}(\varphi)$ in the expression for Q. Then the gradient $\nabla_\varphi U \in V$ of U with respect to the inner product in V is given by $\nabla_\varphi U = \sum_{i=1}^N \sum_{j=1}^J D_{x_i}^{\alpha_j} K \nabla_{Q(\varphi)}^{ij} P$.*

Proof. The gradient $\nabla_\varphi U$ at φ is defined by $\langle \nabla_\varphi U, v\rangle = \partial_\epsilon U(\epsilon v + \varphi)$ for all variations $v \in V$. For such v, we get using (3.2) that

$$\begin{aligned}\partial_\epsilon U(\epsilon v + \varphi) &= \partial_\epsilon P \circ Q(\epsilon v + \varphi) = \partial_\epsilon P(D_{x_i}^{\alpha^j}(\epsilon v + \varphi)) = \partial_\epsilon P(\epsilon D_{x_i}^{\alpha^j} v + D_{x_i}^{\alpha^j}\varphi) \\ &= \sum_{i=1}^N \sum_{j=1}^J (\nabla_{Q(\varphi)}^{ij} P)^T D_{x_i}^{\alpha^j} v = \left\langle \sum_{i=1}^N \sum_{j=1}^J D_{x_i}^{\alpha^j} \nabla_{Q(\varphi)}^{ij} P, v \right\rangle_V \ .\end{aligned}$$

□

3.2. Momentum and Energy. As a result of Proposition 3.1, the momentum of the gradient of U is $L\nabla_\varphi U = \sum_{i=1}^N \sum_{j=1}^J \nabla_{Q(\varphi)}^{ij} P \otimes D_{x_i}^{\alpha_j}$. In general, if $v \in V$ is a sum of higher order kernels, the energy $\|v\|_V^2$ can be computed using (3.2) as a sum of the different order kernels evaluated at the points x_i. To keep the notation brief, we restrict to sums of zero- and first order kernels in the following. If $v(\cdot) = \sum_{i=1}^N \left(K(x_i, \cdot)a_i + \sum_{j=1}^d D^j K(x_i, \cdot)a_i^j\right)$, we get the energy

$$\begin{aligned}\|v\|_V^2 &= \left\langle \sum_{i=1}^N \left(K(x_i,\cdot)a_i + \sum_{j=1}^d D^j K(x_i,\cdot)a_i^j\right), \sum_{i=1}^N \left(K(x_i,\cdot)a_i + \sum_{j=1}^d D^j K(x_i,\cdot)a_i^j\right) \right\rangle_V \\ &= \sum_{i,l=1}^N \langle K(x_l,\cdot)a_l, K(x_i,\cdot)a_i\rangle_V + \sum_{i,l=1}^N \sum_{j,j'}^d \left\langle D^j K(x_l,\cdot)a_l^j, D^{j'} K(x_i,\cdot)a_i^{j'} \right\rangle_V \\ &\quad + 2\sum_{i,l=1}^N \sum_{j=1}^d \left\langle D^j K(x_l,\cdot)a_l^j, K(x_i,\cdot)a_i \right\rangle_V \\ &= \sum_{i,l=1}^N a_l^T K(x_l,x_i)a_i + \sum_{i,l=1}^N \sum_{j,j'}^d a_i^{j',T} D_2^{j'} D_1^j K(x_l,x_i)a_l^j + 2\sum_{i,l=1}^N \sum_{j=1}^d a_i^T D_1^j K(x_l,x_i)a_l^j\end{aligned} \tag{3.3}$$

with $D_i^j K(\cdot,\cdot)$ denoting differentiation with the respect to the ith variable, $i = 1, 2$, and jth coordinate, $j = 1, \ldots, d$. For scalar symmetric kernels such as Gaussians, this expression reduces to

$$\|v\|_V^2 = \sum_{i,l=1}^{N} a_l^T K(x_l, x_i) a_i + \sum_{i,l=1}^{N} \sum_{j,j'}^{d} \left(D_2 \nabla_1 K(x_l, x_i)\right)_j^{j'} a_i^{j',T} a_l^j + 2 \sum_{i,l=1}^{N} \sum_{j=1}^{d} (\nabla_1 K(x_l, x_i))^j a_i^T a_l^j \ .$$

3.3. Locally Affine Transformations. The Polyaffine and Log-Euclidean Polyaffine [3, 1] frameworks model locally affine transformations using matrix logarithms which has limited range. Though the higher order kernels can be seen as the LDDMM sibling of the Polyaffine methods, the methods differ in that diffeomorphism paths generated by higher order kernels, in particular, kernels of zero- and first order, can locally approximate all affine transformation with linear component having positive determinant. The approximation will depend only on how fast the kernel approaches zero towards infinity. The manifold structure of G_V provides this result immediately. Indeed, let $\varphi(x) = Ax + b$ be an affine transformation with $\det(A) > 0$. We define a path φ_t of finite energy such that $\varphi_1 \approx \varphi$ which shows that $\varphi_1 \in G_V$ and can be reached in the framework. The matrices of positive determinant is path connected so we can let ψ_t be a path from Id_d to A and define $\tilde{\psi}_t(x) = \psi_t x + bt$. Then with $\tilde{v}_t(x) = (\partial_t \psi_t)\tilde{\psi}_t^{-1}(x) + b$, we have $\partial_t \tilde{\psi}_t(x) = (\partial_t \psi_t)x + b = \tilde{v}_t \circ \tilde{\psi}_t(x)$ and

$$x + \int_0^1 \tilde{v}_t \circ \tilde{\psi}_t(x) dt = x + \int_0^1 (\partial_t \psi_t)x + b dt = \varphi(x) \ .$$

Now use that $(\partial_t \psi_t)\tilde{\psi}_t^{-1}(x) = (\partial_t \psi_t)(\psi_t)^{-1}(x - bt)$ and let the $M_t = (m_{1,t} \ldots m_{d,t})$ be the t-dependent matrix $(\partial_t \psi_t)(\psi_t)^{-1}$ so that the first term of $\tilde{v}_t(x)$ equals $M_t(x - bt)$. Then choose a radial kernel, e.g. a Gaussian K_σ, and define the approximation v_t of $\tilde{v}_t$ by

$$v_t(x) = \sum_{j=1}^{d} D^j_{\tilde{\psi}_t(0)} K_\sigma(x) m_{j,t} + K_\sigma(\tilde{\psi}_t(0), x) b \ . \tag{3.4}$$

The path φ_{01}^v generated by v_t then has finite energy, and

$$\varphi_{01}^v(x) = x + \int_0^1 v_t \circ \varphi_{0t}^v(x) dt \approx \varphi(x)$$

with the approximation depending only on the kernel scale σ. Note that the affine transformations with linear components having negative determinant can in a similar way be reached by starting the integrating at a diffeomorphism with negative Jacobian determinant.

In the experiments section, we will illustrate the locally affine transformations encoded by zero and first order kernels, and, therefore, it will be useful to introduce a notation for these kernels. We encode the translational part of either the momentum or velocity using the notation

$$\mathrm{Tsl}_x(b) = K_\sigma(x, \cdot)b$$

and the linear part by

$$\mathrm{Lin}_x(M) = \sum_{j=1}^{d} D_x^j K_\sigma(\cdot) m_j$$

with m_1, m_j being the columns of the matrix M. Equation (3.4) can then be written

$$v_t(x) = \mathrm{Lin}_{\tilde{\psi}_t(0)}(M_t) + \mathrm{Tsl}_{\tilde{\psi}_t(0)}(b) \ . \tag{3.5}$$

We emphasize that though we mainly focus on zero and first order kernels, the mathematical construction allows any order kernel permitted by the smoothness of the kernel at order zero.

3.4. EPDiff Equations. It is important to note that the higher order kernels offer a convenient representation for the gradients of maps U incorporating derivative information but since the kernels are members of V and their momentum in the dual V^*, the analytical of structure of LDDMM is not changed. In particular, the adjoint form of the EPDiff equations, i.e. that optimal paths v_t satisfy $v_t = \mathrm{Ad}^T_{\varphi^v_{t1}} v_1$ with $v_1 = -\frac{1}{2}\nabla_{\varphi^v_{01}} U$, is still valid. The momentum $\rho_1 = Lv_1$ is transported to the momentum ρ_t by $\mathrm{Ad}^*_{\varphi^v_{t1}} p_1$. Because

$$(\rho_t|w) = (\rho_1|\mathrm{Ad}_{\varphi^v_{t1}}(w)) = (\rho_1|(D\varphi^v_{t1}\, w) \circ (\varphi^v_{t1})^{-1}) \ ,$$

if ρ_1 is a sum of higher order kernels, ρ_t will be sum of higher order kernels for all t. However, since the time evolution of $(\rho_t|w)$ with the above relation involves derivatives of $D\varphi^v_{t1}$, this form is inconvenient for computing ρ_t. Instead, we make use of the Hamiltonian form of the EPDiff equations [32, P. 265]. Here, the momentum ρ_t is pulled back to ρ_0 but with a coordinate change of the evaluation vector field: the Hamiltonian form μ_t is defined by $\big(\mu_t\big|w\big) := \big(\rho_0\big|(D\varphi^v_{0t})^{-1}(y)w(y)\big)_y$ where the subscript stresses that $(D\varphi^v_{0t})^{-1}(y)w(y)$ is evaluated as a y-dependent vector field. Using this notation, the evolution equations become

$$\begin{aligned} \partial_t \varphi^v_{0t}(y) &= \sum_{k=1}^{d} \big(\mu_t\big|K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y))\big)_x e_k \\ \big(\partial_t \mu_t\big|w\big) &= -\sum_{k=1}^{d} \big(\mu_t\big|\big(\mu_t\big|D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y))w(y)\big)_x e_k\big)_y \ . \end{aligned} \tag{3.6}$$

For the case when $(\rho_0|w)$ does not involve derivatives of w, these equations form an ordinary differential equation describing the evolution of the path and momentum [32]. For the higher order case, we will need to incorporate additional information in the system.

Again we restrict to the zero- and first order case, and we hence work with initial momenta on the form

$$\rho_0 = \sum_{i=1}^{N} a_{0,i} \otimes \delta_{x_{0,i}} + \sum_{i=1}^{N}\sum_{j=1}^{d} a^j_{0,i} \otimes D^j \delta_{x_{0,i}} \tag{3.7}$$

with $x_{t,i}$ as usual denoting the point positions $\varphi^v_{0t}(x_i)$ at time t. Then

$$\begin{aligned}
\big(\mu_t\big|w\big) &= \big(\rho_0\big|D\varphi^v_{0t}(y)^{-1}w(y)\big)_y = \int_\Omega \Big(\sum_{i=1}^N a_{0,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\Big)D\varphi^v_{0t}(y)^{-1}w(y)dy \\
&= \sum_{i=1}^N \big(a_{0,i}\otimes\delta_{x_{0,i}}\big|D\varphi^v_{0t}(y)^{-1}w(y)\big)_y + \sum_{j=1}^d \big(a^j_{0,i}\otimes\delta_{x_{0,i}}\big|\big(D^jD\varphi^v_{0t}(y)^{-1}\big)w(y)\big)_y \\
&\quad + \sum_{i=1}^N\sum_{j=1}^d \big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\big|w\big) \\
&= \sum_{i=1}^N \Big(\big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a_{0,i} + \sum_{j=1}^d \big(D^jD\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i}\big)\otimes\delta_{x_{0,i}}\big|w\Big) \\
&\quad + \sum_{i=1}^N\sum_{j=1}^d \big(D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\otimes D^j\delta_{x_{0,i}}\big|w\big)
\end{aligned}$$

showing that $\mu_t = \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}$ with

$$\begin{aligned}
\mu_{t,i} &= D\varphi^v_{0t}(x_{0,i})^{-1,T}a_{0,i} + \sum_{j=1}^d \big(D^jD\varphi^v_{0t}(x_{0,i})^{-1}\big)^T a^j_{0,i} \\
\mu^j_{t,i} &= D\varphi^v_{0t}(x_{0,i})^{-1,T}a^j_{0,i}\ .
\end{aligned} \tag{3.8}$$

The momentum ρ_t can the be recovered as

$$\begin{aligned}
\big(\rho_t\big|w\big) &= \big(\mu_t\big|w\circ\varphi^v_{0t}\big) = \big(\sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{0,i}} + \sum_{i=1}^N\sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}\big)w\circ\varphi^v_{0t} \\
&= \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{t,i}}w + \sum_{i=1}^N\sum_{j=1}^d \mu^{j,T}_{t,i}Dw(D^j\varphi^v_{0t})(x_{0,i}) \\
&= \sum_{i=1}^N \mu_{t,i}\otimes\delta_{x_{t,i}}w + \sum_{i=1}^N\sum_{j=1}^d \big(\sum_{k=1}^d (D^k\varphi^v_{0t})(x_{0,i})^j\mu^k_{t,i}\big)\otimes D^j\delta_{x_{t,i}}w
\end{aligned}$$

and hence the coefficients of the momentum $a_{t,i}$ and $a^j_{t,i}$ (confer (3.7)) are given by $a_{t,i} = \mu_{t,i}$ and $a^j_{t,i} = \sum_{k=1}^d (D^k\varphi^v_{0t})(x_{0,i})^j\mu^k_{t,i}$.

3.5. Time Evolution of the EPDiff Equations. Even though $\mu_{t,i}$ in (3.8) depend on the second order derivative of φ, we will show that the complete evolution in the zero- and first order case can be determined by solving for $\varphi^v_{0t}(x_{i,0})$, $D\varphi^v_{0t}(x_{i,0})$, and $\mu_{t,i}$. This will provide the computational representation we will use when implementing the systems. In order to simplify the notation, we will work mainly with scalar kernels so that $K^k_l(x,y) = K(x,y)$ if and only if $k = l$ and 0 otherwise.

Using (3.6), φ^v_{0t} evolves according to

$$\begin{aligned}
\partial_t\varphi^v_{0t}(y) &= \sum_{k=1}^d \int_\Omega \sum_{i=1}^N \big(\mu^T_{t,i}\otimes\delta_{x_{0,i}} + \sum_{j=1}^d \mu^j_{t,i}\otimes D^j\delta_{x_{0,i}}\big)K^k(\varphi^v_{0t}(x),\varphi^v_{0t}(y))dxe_k \\
&= \sum_{k=1}^d\sum_{i=1}^N \big(\mu^T_{t,i}K^k(\varphi^v_{0t}(x_{0,i}),\varphi^v_{0t}(y)) + \sum_{j=1}^d \mu^{j,T}_{t,i}D_1K^k(\varphi^v_{0t}(x_{0,i}),\varphi^v_{0t}(y))D^j\varphi^v_{0t}(x_{0,i})\big)e_k\ .
\end{aligned}$$

With scalar kernels, the trajectories $x_{t,i}$ are given by

$$\partial_t \varphi^v_{0t}(x_{0,n}) = \sum_{i=1}^{N} \big(K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))\mu_{t,i} + \sum_{j=1}^{d} \nabla_1 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))^T D^j \varphi^v_{0t}(x_{0,i})\mu^j_{t,i}\big) \ .$$

It is shown in [32] that the evolution of $D\varphi^v_{0t}(x_{i,0})$ is given by

$$\partial_t D\varphi^v_{0t}(y)a = \sum_{k=1}^{d} \big(\mu_t \big| D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)a\big)_x e_k \ .$$

Inserting the Hamiltonian form of the higher order momentum, each component (l,k) of the matrix $D\varphi^v_{0t}(y)$ thus evolves according to

$$\begin{aligned}
\partial_t D\varphi^v_{0t}(y)^l_k &= \big(\mu_t \big| D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)e_l\big)_x \\
&= \int_\Omega \sum_{i=1}^{N} \big(\mu_{t,i} \otimes \delta_{x_{0,i}} + \sum_{j=1}^{d} \mu^j_{t,i} \otimes D^j \delta_{x_{0,i}}\big) D_2 K^k(\varphi^v_{0t}(x), \varphi^v_{0t}(y)) D\varphi^v_{0t}(y)e_l dx \\
&= \sum_{i=1}^{N} \mu^T_{t,i} D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D\varphi^v_{0t}(x_{0,n})e_l \\
&\quad + \sum_{i=1}^{N}\sum_{j=1}^{d} \mu^{j,T}_{t,i} \big(\sum_{m=1}^{d} \big(D^m_1 D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))\big)\big(D^j \varphi^v_{0t}(x_{0,i})\big)^m\big) D\varphi^v_{0t}(x_{0,n})e_l \ .
\end{aligned}$$

With scalar kernels, the evolution at the trajectories is then

$$\begin{aligned}
\partial_t D\varphi^v_{0t}(x_{0,n})^l = \sum_{i=1}^{N} \Big(&\nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))^T D^l \varphi^v_{0t}(x_{0,n})\mu_{t,i} \\
&+ \sum_{j=1}^{d} \big(D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i})\big)^T D^l \varphi^v_{0t}(x_{0,n})\mu^j_{t,i}\Big) \ .
\end{aligned}$$

The complete derivation of the evolution of μ_t is notationally heavy and can be found in Appendix A. Combining this derivation with the expressions above, we arrive at the following result:

PROPOSITION 3.2. *The EPDiff equations in the scalar case with zero- and first order kernels are given in Hamiltonian form by the system*

$$\begin{aligned}
\partial_t \varphi^v_{0t}(x_{0,n}) &= \sum_{i=1}^{N} \big(K(x_{t,i}, x_{t,n})\mu_{t,i} + \sum_{j=1}^{d} \nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi^v_{0t}(x_{0,i})\mu^j_{t,i}\big) \\
\partial_t D\varphi^v_{0t}(x_{0,n})^l &= \sum_{i=1}^{N} \Big(\nabla_2 K(x_{t,i}, x_{t,n})^T D^l \varphi^v_{0t}(x_{0,n})\mu_{t,i} \\
&\qquad + \sum_{j=1}^{d} \big(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi^v_{0t}(x_{0,i})\big)^T D^l \varphi^v_{0t}(x_{0,n})\mu^j_{t,i}\Big)
\end{aligned} \tag{3.9}$$

$$
\begin{aligned}
\partial_t \mu_{t,n} = &-\sum_{i=1}^{N} \Big(\big(\mu_{t,n}^T \mu_{t,i}\big) \nabla_2 K(x_{t,i}, x_{t,n}) \\
&\quad + \sum_{j=1}^{d} \big(\mu_{t,n}^{j,T} \mu_{t,i} - \mu_{t,n}^T \mu_{t,i}^j\big) D_2 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,n}) \\
&\quad + \sum_{j,j'=1}^{d} \big(\mu_{t,n}^{j',T} \mu_{t,i}^j\big) D_2 \big(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,i})\big) D^{j'} \varphi_{0t}^v(x_{0,n}) \Big) \\
\mu_{t,i}^j = &\, D\varphi_{0t}^v(x_{0,i})^{-1,T} a_{0,i}^j \ .
\end{aligned}
$$

Note that both $x_{1,i} = \varphi_{01}^v(x_{0,i})$ and $D\varphi_{01}^v(x_{0,i})$ are provided by the equation, and hence can be used to evaluate a similarity measure such as $\tilde{U}^1$ which depend on these entities. As in the zero-order case, the entire evolution can be recovered by the initial conditions for the momentum.

4. Variations of the Initial Conditions. There exists various choices of optimization algorithms for LDDMM registration. Roughly, they can be divided into two groups based on whether they represent the initial momentum/velocity or the entire path φ_t. Here, we take the approach of incorporating higher order kernels with the shooting method of e.g. Vaillant et al. [28]. The algorithm will take a guess for initial momentum, integrate the EPDiff equations forward, compute the similarity measure gradient ∇U, and flow the gradient backwards to provide an improved guess. For this to work, we will need the variation of the EPDiff equations when varying the initial conditions. Following this, we discuss the backwards gradient transport and arrive at a full matching algorithm.

A variation $\delta\rho_0$ of the initial momentum will induce a variation of the system (3.9). By differentiating the system, we get the time evolution of the variation. To ease notation, we assume the scalar kernel has the form $K(x,y) = \gamma(|x-y|^2)$ and write $\gamma_{t,in} = K(x_{t,i}, x_{t,n})$. Variations of the kernel and kernel derivatives such as the entity $\delta\nabla_1 K(x_{t,i}, x_{t,n})$ below depend only on the variation of point trajectories $\delta x_{t,i}$. The full expressions for these parts are provided in Appendix B. The evolution of the derived system then takes the following form:

$$
\begin{aligned}
\partial_t \delta\varphi_{0t}^v(x_{0,n}) = &\sum_{i=1}^{N} \big(\delta K(x_{t,i}, x_{t,n}) \mu_{t,i} + \gamma_{t,in} \delta\mu_{t,i}\big) \\
&+ \sum_{i=1}^{N} \sum_{j=1}^{d} \big(\delta\nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i}) \mu_{t,i}^j + \nabla_1 K(x_{t,i}, x_{t,n})^T \delta D^j \varphi_{0t}^v(x_{0,i}) \mu_{t,i}^j \\
&\quad + \nabla_1 K(x_{t,i}, x_{t,n})^T D^j \varphi_{0t}^v(x_{0,i}) \delta\mu_{t,i}^j\big)
\end{aligned}
\tag{4.1}
$$

$$
\begin{aligned}
\partial_t \delta D\varphi_{0t}^v(x_{0,n})^l = &\sum_{i=1}^{N} \big(\delta\nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi_{0t}^v(x_{0,n})\mu_{t,i} + \nabla_2 K(x_{t,i},x_{t,n})^T \delta D^l\varphi_{0t}^v(x_{0,n})\mu_{t,i} \\
&+ \nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi_{0t}^v(x_{0,n})\delta\mu_{t,i}\big) \\
&+ \sum_{i=1}^{N}\sum_{j=1}^{d} \Big(\big(\delta D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)^T D^l\varphi_{0t}^v(x_{0,n})\mu_{t,i}^j \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})\delta D^j\varphi_{0t}^v(x_{0,i})\big)^T D^l\varphi_{0t}^v(x_{0,n})\mu_{t,i}^j \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)^T \delta D^l\varphi_{0t}^v(x_{0,n})\mu_{t,i}^j \\
&\quad + \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)^T D^l\varphi_{0t}^v(x_{0,n})\delta\mu_{t,i}^j\Big) \\
\partial_t \delta\mu_{t,n} = &-\sum_{i=1}^{N}\big(\big(\delta\mu_{t,n}^T\mu_{t,i} + \mu_{t,n}^T\delta\mu_{t,i}\big)\nabla_2 K(x_{t,i},x_{t,n}) + \big(\mu_{t,n}^T\mu_{t,i}\big)\delta\nabla_2 K(x_{t,i},x_{t,n})\big) \\
&- \sum_{i=1}^{N}\sum_{j=1}^{d}\big(\big(\delta\mu_{t,n}^{j,T}\mu_{t,i} + \mu_{t,n}^{j,T}\delta\mu_{t,i} - \delta\mu_{t,n}^T\mu_{t,i}^j - \mu_{t,n}^T\delta\mu_{t,i}^j\big)D_2\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,n}) \\
&\quad + \big(\mu_{t,n}^{j,T}\mu_{t,i} - \mu_{t,n}^T\mu_{t,i}^j\big)\delta D_2\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,n}) \\
&\quad + \big(\mu_{t,n}^{j,T}\mu_{t,i} - \mu_{t,n}^T\mu_{t,i}^j\big)D_2\nabla_2 K(x_{t,i},x_{t,n})\delta D^j\varphi_{0t}^v(x_{0,n})\big) \\
&- \sum_{i=1}^{N}\sum_{j,j'=1}^{d}\big(\big(\delta\mu_{t,n}^{j',T}\mu_{t,i}^j + \mu_{t,n}^{j',T}\delta\mu_{t,i}^j\big)D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)D^{j'}\varphi_{0t}^v(x_{0,n}) \\
&\quad + \big(\mu_{t,n}^{j',T}\mu_{t,i}^j\big)\delta D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)D^{j'}\varphi_{0t}^v(x_{0,n}) \\
&\quad + \big(\mu_{t,n}^{j',T}\mu_{t,i}^j\big)D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)\delta D^{j'}\varphi_{0t}^v(x_{0,n})\big)\ .
\end{aligned}
$$

The variation of $\mu_{t,i}^j$ is available as

$$
\delta\mu_{t,i}^j = -\big(D\varphi_{0t}^v(x_{0,i})^{-1}\delta D\varphi_{0t}^v(x_{0,i})D\varphi_{0t}^v(x_{0,i})^{-1}\big)^T a_{0,i}^j + D\varphi_{0t}^v(x_{0,i})^{-1,T}\delta a_{0,i}^j\ .
$$

However, when computing the backwards transport, we will need to remove the dependency on $\delta a_{0,i}^j$ which is only available for forward integration. Instead, by writing the evolution of $\mu_{t,i}^j$ in the form

$$
\begin{aligned}
\partial_t\mu_{t,i}^j &= \partial_t D\varphi_{0t}^v(x_{0,i})^{-1,T}a_{0,i}^j = -\big(D\varphi_{0t}^v(x_{0,i})^{-1}\partial_t D\varphi_{0t}^v(x_{0,i})D\varphi_{0t}^v(x_{0,i})^{-1}\big)^T a_{0,i}^j \\
&= -D\varphi_{0t}^v(x_{0,i})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,i})^T\mu_{t,i}^j\ ,
\end{aligned}
$$

we get the variation

$$
\begin{aligned}
\partial_t\delta\mu_{t,n}^j = &-\delta D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\mu_{t,n}^j - D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t\delta D\varphi_{0t}^v(x_{0,n})^T\mu_{t,n}^j \\
&- D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\delta\mu_{t,n}^j\ .
\end{aligned}
$$

4.1. Backwards Transport. The correspondence between initial momentum ρ_0 and end diffeomorphism φ_{01}^v asserted by the EPDiff equations allows us to view the similarity measure $U(\varphi_{01}^v)$ as a function of ρ_0. Let A denote the result of integrating the system for the variation of the initial conditions from $t = 0$ to $t = 1$ such that $w = A\delta\rho_0 \in V$ for a variation $\delta\rho_0$. We then get a corresponding variation δU in the

similarity measure. To compute the gradient of U as a function of ρ_0, we have

$$\delta U(\varphi_{01}^v) = \left\langle \nabla_{\varphi_{01}^v} U, w \right\rangle_V = \left\langle \nabla_{\varphi_{01}^v} U, A\delta\rho_0 \right\rangle_V = \left\langle A^T \nabla_{\varphi_{01}^v} U, \delta\rho_0 \right\rangle_{V^*} \ .$$

Thus, the V^*-gradient of $\nabla_{\rho_0} U$ is given by $A^T \nabla_{\varphi_{01}^v} U$. The gradient can equivalently be computed in momentum space at both endpoints of the diffeomorphism path using the map P defined in Proposition 3.1.

The complete system for the variation of the initial conditions is a linear ODE, and, therefore, there exists a time-dependent matrix M_t such that the ODE

$$\partial_t y_t = M_t y_t$$

has the variation as a solution y_t. It is shown in [32] that, in such cases, solving the backwards transpose system

$$\partial_t w_t = -M_t^T w_t \tag{4.2}$$

from $t = 1$ to $t = 0$ provides the value of $A^T w$. Therefore, we can obtain $\nabla_{\rho_0} U$ by solving the transpose system backwards. The components of M_t can be identified by writing the evolution equations for the variation in matrix form. This provides M_t^T and allows the backwards integration of the system 4.2. The components of the transpose matrix M_t are provided in Appendix C.

4.2. Algorithm. The registration problem (2.1) consists of both the similarity measure U and the minimal path energy E_1. For e.g. landmark based registration, $U(\varphi)$ is most often expressed in terms of φ directly whether as U is usually dependent on the inverse φ^{-1} for image registration. In the first case, the gradient $\nabla_\varphi U$ is known, and, given the initial momentum ρ_0, we can obtain the gradient $\nabla_{\rho_0} U$ for a gradient descent based optimisation procedure from the backwards transport equations (4.2) discussed above. For the energy part, it is a fundamental result property of critical paths in the LDDMM framework that the energy stays constant along the path. Thus, $\int_0^1 \|v_t\|_V^2 dt = \|v_0\|_V^2 = \big(\rho_0 \big| K(\rho_0)\big)^2$ and we can easily compute the gradient from (3.3). Given this, the zero-order matching algorithm in the initial momentum is generalized to zero- and first order kernels in Algorithm 1.

Algorithm 1 Matching with Higher Order Kernels.

$\rho_0 \leftarrow$ initial guess
repeat
  Solve EPDiff equations forward
  Compute U and ∇P
  Solve backwards the transpose equations
  Compute the energy gradient $\nabla \|v_0\|^2$
  Update ρ_0 from $\nabla \|v_0\|^2 + \nabla_{\rho_0} U$
until convergence

Traditionally, the similarity measure $U(\varphi)$ is in image matching formulated using the inverse of φ, and this approach was taken when formulating the approximation (3.1). For this reason, at finite control point formulation is naturally expressed using a sampling $\{x_1, \ldots, x_N\}$ in the *target* image with the algorithm optimizing for the momentum ρ_1 at time $t = 1$. The evaluation points $\varphi^{-1}(x_i)$ are then generated by flowing *backwards* from $t = 1$ to $t = 0$, the gradient of $U(\varphi)$ can then be computed

in $\varphi^{-1}(x_i)$ and flowed *forwards* to update ρ_1. This corresponds to switching the role of the moving and target image combined with backwards integration of the flow equations. Algorithm 1 will accommodate this situation by just reversing the integration directions. The control points can be chosen either at e.g. anatomically important locations, at random, or on a regular grid. In the experiments, we will register expanding ventricles using control points placed in the ventricles.

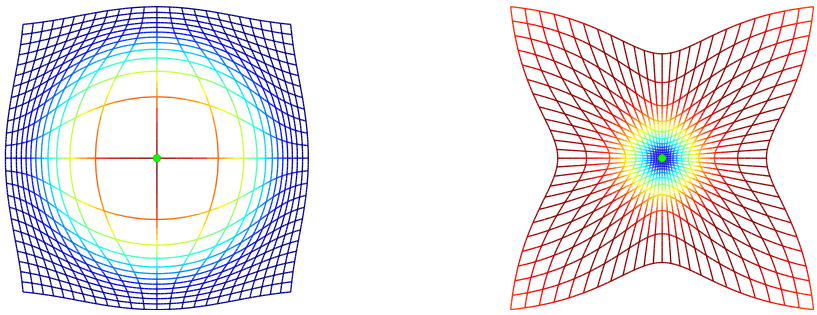
The integration of the ODEs can be performed with standard Runge-Kutta integrators such as Matlabs `ode45` procedure. With zero order kernels only and N points, the forward and backwards system consist of $2dN$ equations. With zero- and first order kernels, the forward system is extended to $N(2d + d^2)$ and the backwards system to $2N(d+d^2)$. For $d = 3$, this implies an 2.5 time increase in the size of the system. In addition to this should be considered the extra floating point operations necessary for computing the somewhat more complicated evolution equations. This increase should, however, be viewed against the fact that the finite dimensional system contain orders of magnitude fewer control points, and the added capacity of deformation description included in the derivative information. In addition and in contrast to previous approaches, we transport the similarity gradient *only* at the control point trajectories, again an order of magnitude reduction of transported information. As we will see in the following section, the inclusion of higher order kernels provides information with very few control points.

5. Experiments. In order to demonstrate the efficiency and sparsity of representations using higher order kernels, we perform four sets of experiments. First, we provide four examples illustrating the type of deformations produced by zero- and first order kernels and the relation to the Polyaffine framework. We then use point based matching using first order information to show how complicated warps that would require many parameters with zero order deformation atoms can be generated with very compact representations using higher order kernels. When then underline the point that higher order kernels allow low-dimensional transformations to be registered using correspondingly low-dimensional representations: we show how synthetic test images generated by a low-dimensional transformation can be registered using only one deformation atom when representing using first order kernels and using the first order similarity measure approximation (3.1). We further emphasize this point by registering articulated movement using only one deformation atom per rigid part, and thus exemplify a natural representation that reduces the number of deformation atoms and the ambiguity in the placement of the atoms while also reducing the degrees of freedom in the representation. Finally, we illustrate how higher order kernels in a natural way allow registration of human brains with progressing atrophy. We describe the deformation field throughout the ventricles using few deformation atoms, and we thereby suggest a method for detecting anatomical change using few degrees of freedom. We start by briefly describing the similarity measures used throughout the experiments.

For the point examples below, we register moving points $x_1, \ldots, x_N$ against fixed points $y_1, \ldots, y_N$. In addition, we match first order information by specifying values of $D^j_{x_i}\varphi$. This is done compactly by providing matrices Y_i so that we seek $D_{x_i}\varphi = Y_i$ for all $i = 1, \ldots, N$. The similarity measure is simple sum of squares, i.e.

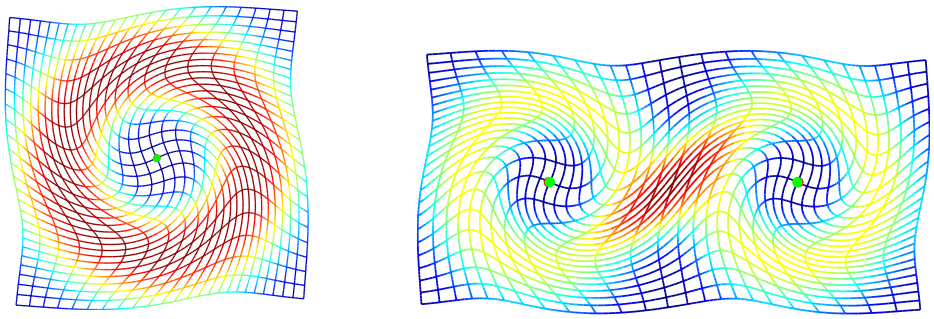
$$U(\varphi) = \sum_{i=1}^{N} \|\varphi(x_i) - y_i\|^2 + \|D_{x_i}\varphi - Y_i\|^2$$

using the matrix 2-norm. This amounts to fitting φ against a locally affine map with



(a) Expansion

(b) Contraction



(c) Rotation $(-\pi/2)$

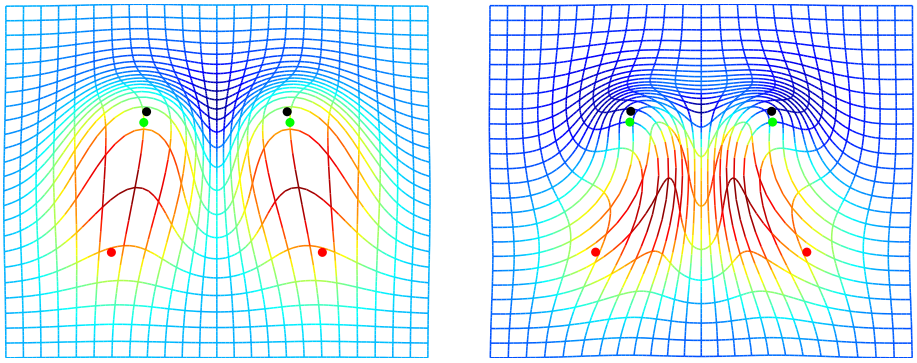
(d) Two rotations $(\pi/2)$

FIG. 5.1. *The effect of the generated deformation on an initially square grid for several initial first order momenta: Using the notation of Section 3.3, (a) expansion* $\rho_0 = \mathrm{Lin}_0(\mathrm{Id}_2)$*; (b) contraction* $\rho_0 = \mathrm{Lin}_0(-\mathrm{Id}_2)$*; (c) rotation* $\rho_0 = \mathrm{Lin}_0(\mathrm{Rot}(v))$*,* $v = -\pi/2$*; (d) two rotations* $v = \pi/2$*. The kernel is Gaussian with* $\sigma = 8$ *in grid units, and the grids are colored with the trace of Cauchy-Green strain tensor (log-scale). Notice the locality of the deformation caused by the finite scale of the kernel, and that the deformation stays diffeomorphic even when two rotations force conflicting movements.*

translational components y_i and linear components Y_i. For the image cases, we use L^1-similarity to build the first order approximation (3.1) with the smoothing kernel K_s being Gaussian of the same scale as the LDDMM kernel.

5.1. First Order Illustrations. To visually illustrate the deformation generated by higher order kernels, we show in Figure 5.1 the generated deformations on an initially square grid with four different first-order initial momenta. The deformation locally model the linear part of affine transformations and the the locality is determined by the Gaussian kernel that in the examples has scale $\sigma = 8$ in grid units. Notice for the rotations that the deformation stays diffeomorphic in the presence of conflicting forces. The similarity between the examples and the deformations generated in the Polyaffine framework [1] underlines the viewpoint that the registration using higher order kernels constitutes the LDDMM sibling of the Polyaffine framework.

5.2. First Order Point Registration. Figure 5.2 presents simple point based matching results with first order information. The lower points (red) are matched against the upper points (black) with match against expansion $D_\varphi(x_i) = 2\mathrm{Id}_2$ and



(a) Match with dilations (expansion) (b) Match with rotations ($-\pi/2$ and $\pi/2$)

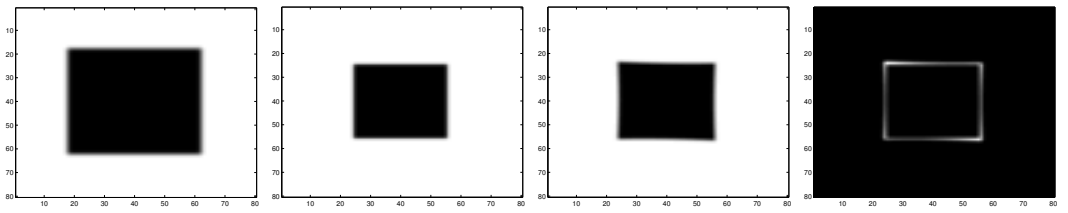
FIG. 5.2. *Two moving points (red) are matched against two fixed points (black) with results (green) and with match against (a) expansion* $D_\varphi(x_i) = 2\mathrm{Id}_2$, $i = 1, 2$*; and (b) rotation* $D_\varphi(x_i) = \mathrm{Rot}(v)$, $v = \mp\pi/2$, $i = 1, 2$. *The kernel is Gaussian with* $\sigma = 8$ *in grid units, and the grids are colored with the trace of Cauchy-Green strain tensor (log-scale).*

rotation $D_\varphi(x_i) = \mathrm{Rot}(v) = \begin{pmatrix} \cos(v), \sin(v) \\ -\sin(v), \cos(v) \end{pmatrix}$ for $v = \mp\pi/2$. The optimal diffeomorphisms exhibit the expected expanding and turning effect, respectively. We stress that the deformations are generated using only two deformations atoms with combined 12 parameters. Representing equivalent deformation using zero order kernels would require a significantly increased number of atoms and a correspond increase in the number of parameters.

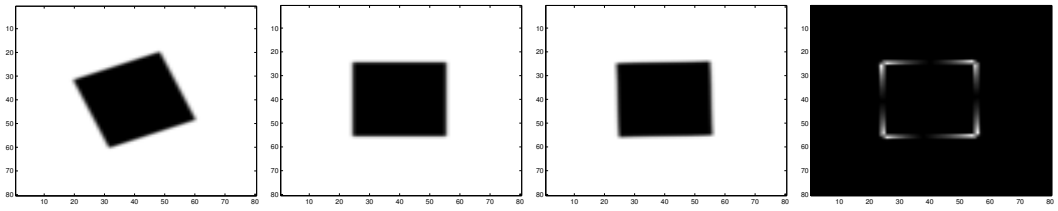
5.3. Low Dimensional Image Registration. We now exemplify how higher order kernels allow low-dimensional transformations to be registered using correspondingly low-dimensional representations. We generate two test images by applying two linear transformations, an dilation and a rotation, to a binary image of a square, confer the moving images (a) and (e) in Figure 5.3. By placing one deformation atom in the center of each fixed image and by using the similarity measure approximation (3.1), we can successfully register the moving and fixed images. The result and difference plots are shown in Figure 5.3. The dimensionality of the linear transformations generating the moving images is equal to the number of parameters for the deformation atom. A registration using zero order kernels would need more than one deformation atom which would result in a number of parameters larger than the dimensionality. The scale of the Gaussian kernel used for the registration is 50 pixels.

5.4. Articulated Motion. The articulated motion of the finger[3] in Figure 5.4 (a) and (b) can be described by three locally linear transformations. With higher order kernels, we can place deformation atoms at the center of the bones in the moving and fixed images, and use the point positions together with the direction of the bones to drive a registration. This natural and low dimensional representation allows a fairly good match of the images resembling the use of the Polyaffine affine framework for articulated registration [23]. A similar registration using zero order kernels would

[3]X-ray frames from `http://www.archive.org/details/X-raystudiesofthejointmovements-wellcome`

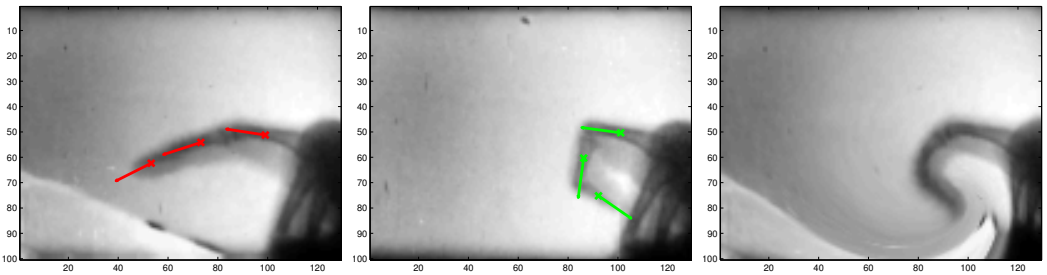


(a) Moving image (b) Fixed image (c) Registration result (d) Difference



(e) Moving image (f) Fixed image (g) Registration result (h) Difference

FIG. 5.3. *With linear transformations, the dimensionality of the higher order representation matches the dimensionality of the transformation. A dilation (e) and rotation (d) is applied to the fixed binary images (b) and (f), respectively. The registration results (c) and (g) subtracted from the fixed images are shown in the difference pictures (d) and (h). The registration is performed with a single first order kernel in the center of the pictures, and the number of parameters for the registration thus matches the dimensionality of the linear representations. The slight differences between results and fixed images are caused by the first order approximation in* (3.1)*. Increasing the kernel size, adding more control points, or using second order kernels would imply less difference.*



(a) Moving (b) Fixed (c) Result, first order

FIG. 5.4. *Registering articulated movement using directional information of the bones: the landmarks and bone orientations (red points and arrows) in the moving image (a) are matched against the landmarks and bone orientations (green points and arrows) in the fixed image (b). The result using first order kernels (c) can be obtained with a low number of deformation atoms that can be consistently placed at the center of the bones. A corresponding zero order representation would use a higher number of atoms with a corresponding increase in the number of parameters.*

need two deformation atoms per bone and lacking a natural way to place such atoms, the positions would need to be optimized. With higher order kernels, the deformation atoms can be placed in a natural and consistent way, and the total number of free parameters is lower than a zero order representation using two atoms per bone.

5.5. Registering Atrophy. Atrophy occurs in the human brain among patients suffering from Alzheimer's disease, and the progressing atrophy can be detected by the expansion of the ventricles [16, 13]. Since first order kernels offer compact description

of expansion, this makes a parametrization of the registration based on higher order kernels suited for describing the expansion of the ventricles, and, in addition, the deformation represented by the kernels will be easily interpretable. In this experiment, we therefore suggest a registration method that using few degrees of freedom describes the expansion of the ventricles, and does so in a way that can be interpreted when doing further analysis of e.g. the volume change.

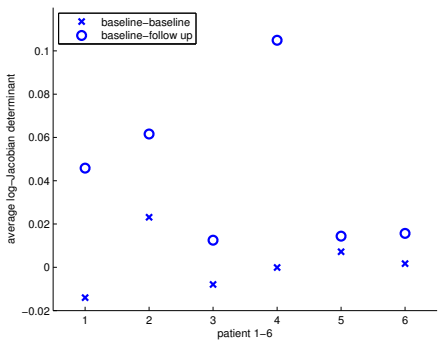
We will provide examples of 2D registration with the purpose of *illustrating* the use of the higher order kernels and suggest a method which can be applied in 3D. We do not aim at a quantitative evaluation but we plan to follow up on the experiment in future work with 3D registration of more subjects and explore the connection between first order initial momentum and actual ventricle expansion in greater detail.

We use the publicly available Oasis dataset[4] [17], and we select a small number of patients from which two baseline scans are acquired at the same day together with a later follow up scan. The patients are in various stages of dementia. We perform rigid registration [9] before selecting vertical 2D slices where the ventricles are clearly visible. The slice plane is the same for all three scans of each patient.

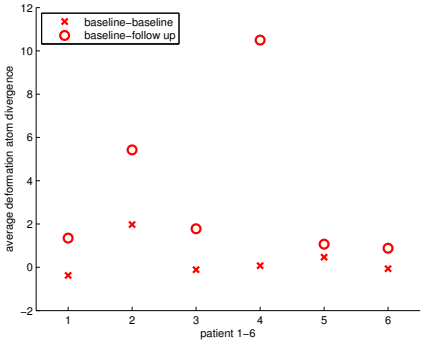
The expanding ventricles can be registered by placing deformation atoms in the form of higher order kernels in the center of the ventricles of the fixed image as shown in Figure 1.1. We manually place five deformation atoms in the ventricle area of each patient. It is important to note that though we localize the description of the deformation to the deformation atoms, the atoms control the deformation field throughout the ventricle area. Based on the size of the ventricles, we use Gaussian kernels with a scale of 15 voxels for the kernels, and we let the regularization weight in (2.1) be $\lambda = 16$. The effect of these choices is discussed below. Each deformation atom consists of a zero- and first order kernel, and, for each patient, we perform two registrations: we register the two baseline scans acquired at the same day, and we register one baseline scan against the follow up scan. Thus, the baseline-baseline registration should indicate no ventricle expansion, and we expect the baseline-follow up registration to indicate ventricle expansion. Figure 1.1 shows for one patient the placement of the control points in the baseline image, the follow up image, the log-Jacobian determinant in the ventricle area of the generated deformation, and the initial vector field driving the registration.

The use of first order kernels allows us to interpret the result of the registrations and to relate the results to possible expansion of the ventricles. The volume change is indicated by the Jacobian determinant of the generated deformation at the deformation atoms as well as by the divergence of the first order kernels. The latter is available directly from the registration parameters. We plot in Figure 5.5 the logarithm of the Jacobian determinant and the divergence for both the same day baseline-baseline registrations and for the baseline-follow up registrations. Patient $1-4$ are classified as demented, patient 5 and 6 as non-demented, and all patient have constant clinical dementia rating through the experiment. The time-span between baseline and follow up scan is 1.5-2 years with the exception of 3 years for patient four. As expected, the log-Jacobian is close to zero for the same day baseline-baseline scans but it increases with the baseline-follow up registrations of the demented patients. In addition, the correlation between the log-Jacobian and the divergence shows how the indicated volume change is available directly from the registration parameters. This result suggests the usefulness of the approach and points to future experiments to validate the method.

[4] `http://www.oasis-brains.org`



(a) The average log-Jacobian of the final deformation at the 5 deformation atoms for the baseline-baseline and baseline-follow up registrations



(b) The average divergence at the deformation atoms for the baseline-baseline and baseline-follow up registrations

FIG. 5.5. *Indicated volume change: (a) The average* log-*Jacobian determinant of the generated deformation at the 5 deformation atoms for six patients (1-4 demented, 5-6 non-demented); (b) divergence of the 5 higher order kernels representing the deformation. The divergence can be extracted directly from the parameters of the higher order kernels, and the correlation between the* log-*Jacobian and the divergence as seen by the similarity between (a) and (b) therefore shows the interpretability of the deformation atoms.*

We chose two important parameters above: the kernel scale and the regularization term. The choice of one scale for all patients works well if the ventricles to be registered are of approximately the same size at the baseline scans. If the ventricles vary in size, the scale can be chosen individually for each patient. Alternatively, a multi-scale approach could do this automatically which suggests combining the method with e.g. the kernel bundle framework [24]. Depending on the image forces, the regularization term in (2.1) will affect the amount of expansion captured in the registration. Because of the low number of control points, we can in practice set the contribution of the regularization term to zero without experiencing non-diffeomorphic results. It will be interesting in the future to estimate the actual volume expansion directly using the parameters of the deformation atoms with this less biased model.

6. Conclusion and Outlook. We have introduced higher order kernels in the LDDMM registration framework. The kernels allow *compact* representation of locally affine transformations by increasing the *capacity* of the deformation description. Coupled with similarity measures incorporating first order information, the higher order kernels improve the range of deformations reached by sparsely discretized LDDMM methods, and they allow direct capture of first order information such as expansion and contraction. In addition, the constitute deformation atoms for which the generated deformation is directly interpretable.

In the paper, we have shown how the partial derivative reproducing property implies singular momentum for the higher order kernels, and we used this to derive the EPDiff evolution equations. By computing the forward and backward variational equations, we are able to transport gradient information and derive a matching algorithm. We provide examples showing typical deformation coded by first order kernels and how images can be registered using a very few parameters, and we have applied the method to register human brains with progressing atrophy.

The experiments included here show only a first step in the application of higher

order kernels: the kernels may be applied to register entire images; merging the method with multi-scale approaches will increase the description capacity and may lead to further reduction in the dimensionality of the representation. Combined with efficient implementations, higher order kernels promise to provide a step forward in compact deformation description for image registration.

Appendix A. Time Evolution of μ_t. Inserting the Hamiltonian form of the momentum, we have

$$
\begin{aligned}
\big(\partial_t\mu_t\big|w\big) &= -\sum_{k=1}^{d}\big(\mu_t\big|\big(\mu_t\big|D_2K^k(\varphi_{0t}^v(x),\varphi_{0t}^v(y))w(y)\big)_x e_k\big)_y \\
&= -\sum_{k=1}^{d}\big(\mu_t\big|\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big)D_2K^k(\varphi_{0t}^v(x),\varphi_{0t}^v(y))w(y)dxe_k\big) \\
&= -\sum_{k=1}^{d}\big(\mu_t\big|\sum_{i=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(y))w(y)e_k\big) \\
&\quad -\sum_{k=1}^{d}\big(\mu_t\big|\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(y))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(y)e_k\big) \\
&= -\sum_{k=1}^{d}\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big)\sum_{i=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(y))w(y)e_kdy \\
&\quad -\sum_{k=1}^{d}\int_\Omega\big(\sum_{i=1}^{N}\mu_{t,i}\otimes\delta_{x_{0,i}}+\sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^j\otimes D^j\delta_{x_{0,i}}\big) \\
&\qquad\qquad \sum_{i=1}^{N}\sum_{j=1}^{d}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(y))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(y)e_kdy \\
&= -\sum_{k=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(x_{0,n}))w(x_{0,n})\mu_{t,n}^T e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T\big(\sum_{m=1}^{d}D_2^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,n}))^m\big)w(x_{0,n})\mu_{t,n}^{j,T}e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^T D_2K^k(\varphi_{0t}^v(x_{0,i})),\varphi_{0t}^v(x_{0,n}))D^jw(x_{0,n})\mu_{t,n}^{j,T}e_k \\
&\quad -\sum_{k,j=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)w(x_{0,n})\mu_{t,n}^T e_k \\
&\quad -\sum_{k,j,j'=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m,m'=1}^{d}D_2^{m'}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m(D^{j'}\varphi_{0t}^v(x_{0,n}))^{m'}\big) \\
&\qquad\qquad w(x_{0,n})\mu_{t,n}^{j',T}e_k \\
&\quad -\sum_{k,j,j'=1}^{d}\sum_{i,n=1}^{N}\mu_{t,i}^{j,T}\big(\sum_{m=1}^{d}D_1^mD_2K^k(\varphi_{0t}^v(x_{0,i}),\varphi_{0t}^v(x_{0,n}))(D^j\varphi_{0t}^v(x_{0,i}))^m\big)D^{j'}w(x_{0,n})\mu_{t,n}^{j',T}e_k\ .
\end{aligned}
$$

Thus

$$\begin{aligned}
\partial_t \mu^T_{t,n} = &- \sum_{k=1}^{d} (\mu_{t,n})^k \sum_{i=1}^{N} \mu^T_{t,i} D_2 K^k(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&- \sum_{k,j=1}^{d} (\mu^j_{t,n})^k \sum_{i=1}^{N} \mu^T_{t,i} \big(\sum_{m=1}^{d} D_2^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,n}))^m \big) \\
&- \sum_{k,j=1}^{d} (\mu_{t,n})^k \sum_{i=1}^{N} \mu^{j,T}_{t,i} \big(\sum_{m=1}^{d} D_1^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,i}))^m \big) \\
&- \sum_{k,j,j'=1}^{d} (\mu^{j'}_{t,n})^k \sum_{i=1}^{N} \mu^{j,T}_{t,i} \\
&\qquad \big(\sum_{m,m'=1}^{d} D_2^{m'} D_1^m D_2 K^k(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n}))(D^j \varphi^v_{0t}(x_{0,i}))^m (D^{j'} \varphi^v_{0t}(x_{0,n}))^{m'} \big) \ .
\end{aligned}$$

We write K^k_l (column-row) and get

$$\begin{aligned}
\partial_t \mu_{t,n} = &- \sum_{k,l=1}^{d} \sum_{i=1}^{N} \Big((\mu_{t,n})^k (\mu_{t,i})^l \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&+ \sum_{j=1}^{d} (\mu^j_{t,n})^k (\mu_{t,i})^l D_2 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,n}) \\
&+ \sum_{j=1}^{d} (\mu_{t,n})^k (\mu^j_{t,i})^l D_1 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \\
&+ \sum_{j,j'=1}^{d} (\mu^{j'}_{t,n})^k (\mu^j_{t,i})^l D_2 \big(D_1 \nabla_2 K^k_l(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \big) D^{j'} \varphi^v_{0t}(x_{0,n}) \Big)
\end{aligned}$$

.

For scalar kernels $K^k_l(x,y) = K(x,y)$ iff $k = l$, and hence

$$\begin{aligned}
\partial_t \mu_{t,n} = &- \sum_{i=1}^{N} \Big(\big(\mu^T_{t,n} \mu_{t,i} \big) \nabla_2 K(\varphi^v_{0t}(x_{0,i})), \varphi^v_{0t}(x_{0,n})) \\
&+ \sum_{j=1}^{d} \big(\mu^{j,T}_{t,n} \mu_{t,i} \big) D_2 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,n}) \\
&+ \sum_{j=1}^{d} \big(\mu^T_{t,n} \mu^j_{t,i} \big) D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \\
&+ \sum_{j,j'=1}^{d} \big(\mu^{j',T}_{t,n} \mu^j_{t,i} \big) D_2 \big(D_1 \nabla_2 K(\varphi^v_{0t}(x_{0,i}), \varphi^v_{0t}(x_{0,n})) D^j \varphi^v_{0t}(x_{0,i}) \big) D^{j'} \varphi^v_{0t}(x_{0,n}) \Big) \ .
\end{aligned}$$

Using symmetry and rewriting,

$$\begin{aligned}\partial_t \mu_{t,n} = -\sum_{i=1}^{N} \Big(& \big(\mu_{t,n}^T \mu_{t,i}\big) \nabla_2 K(x_{t,i}, x_{t,n}) \\ & + \sum_{j=1}^{d} \big(\mu_{t,n}^{j,T} \mu_{t,i} - \mu_{t,n}^T \mu_{t,i}^j\big) D_2 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,n}) \\ & + \sum_{j,j'=1}^{d} \big(\mu_{t,n}^{j',T} \mu_{t,i}^j\big) D_2 \big(D_1 \nabla_2 K(x_{t,i}, x_{t,n}) D^j \varphi_{0t}^v(x_{0,i}) \big) D^{j'} \varphi_{0t}^v(x_{0,n}) \Big) \ .\end{aligned}$$

Appendix B. Variation of the Kernel and Derivatives. With $K(x,y) = \gamma(|x-y|^2)$, we have the following expressions for the derivatives of the kernel:

$$\begin{aligned}
\nabla_1 K &= 2\dot\gamma(|x-y|^2)(x-y) \\
\nabla_2 K &= -2\dot\gamma(|x-y|^2)(x-y) \\
D_1\nabla_1 K &= 2\dot\gamma(|x-y|^2)\mathrm{Id}_d + 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
D_2\nabla_1 K &= -2\dot\gamma(|x-y|^2)\mathrm{Id}_d - 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
D_1\nabla_2 K &= -2\dot\gamma(|x-y|^2)\mathrm{Id}_d - 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
D_2\nabla_2 K &= 2\dot\gamma(|x-y|^2)\mathrm{Id}_d + 4\ddot\gamma(|x-y|^2)(x-y)(x-y)^T \\
D_2(D_1\nabla_2 K a) &= -D_2(2\dot\gamma(|x-y|^2)a + 4\ddot\gamma(|x-y|^2)(x-y)^T a(x-y)) \\
&= +4\ddot\gamma(|x-y|^2)a(x-y)^T + 8\dddot\gamma(|x-y|^2)(x-y)^T a(x-y)(x-y)^T \\
&\quad + 4\ddot\gamma(|x-y|^2)(x-y)a^T + 4\ddot\gamma(|x-y|^2)(x-y)^T a\mathrm{Id}_d \\
&= 4\Big(\ddot\gamma(|x-y|^2)a(x-y)^T + \ddot\gamma(|x-y|^2)(x-y)a^T \\
&\quad + \ddot\gamma(|x-y|^2)(x-y)^T a\mathrm{Id}_d + 2\dddot\gamma(|x-y|^2)(x-y)^T a(x-y)(x-y)^T\Big) \ .
\end{aligned}$$

Variations of these expressions the take the form

$$\begin{aligned}
\delta K(x_{t,i}, x_{t,n}) &= 2\dot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n}) \\
\delta\nabla_1 K(x_{t,i}, x_{t,n}) &= 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n}) + 2\dot\gamma_{t,in}(\delta x_{t,i} - \delta x_{t,n}) \\
\delta\nabla_2 K(x_{t,i}, x_{t,n}) &= -\delta\nabla_1 K(x_{t,i}, x_{t,n}) \\
\delta D_1\nabla_2 K(x_{t,i}, x_{t,n}) &= -4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})\mathrm{Id}_d \\
&\quad - 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T \\
&\quad - 4\ddot\gamma_{t,in}(\delta x_{t,i} - \delta x_{t,n})(x_{t,i}-x_{t,n})^T \\
&\quad - 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(\delta x_{t,i} - \delta x_{t,n})^T \\
\delta D_2\nabla_2 K(x_{t,i}, x_{t,n}) &= -\delta D_1\nabla_2 K(x_{t,i}, x_{t,n})
\end{aligned}$$

$$
\begin{aligned}
\delta D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})a\big) &= D_2\big(D_1\nabla_2 K(x_{t,i},x_{t,n})\delta a\big)\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})a(x_{t,i}-x_{t,n})^T + 4\ddot{\gamma}_{t,in}a(\delta x_{t,i}-\delta x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})a^T + 4\ddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})a^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T a\mathrm{Id}_d + 4\ddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})^T a\mathrm{Id}_d\\
&+ 16\ddddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T a(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(\delta x_{t,i}-\delta x_{t,n})^T a(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T a(\delta x_{t,i}-\delta x_{t,n})(x_{t,i}-x_{t,n})^T\\
&+ 8\dddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})^T a(x_{t,i}-x_{t,n})(\delta x_{t,i}-\delta x_{t,n})^T\ .
\end{aligned}
$$

Appendix C. The Transpose Derivative System. We let M denote the time-dependent matrix governing the linear ODE (4.1) for the evolution of the variation of the initial conditions of the EPDiff equations, and we write M as a block matrix

$$
M = \begin{pmatrix} M^{\varphi\varphi} & M^{\varphi D\varphi} & M^{\varphi\mu} & M^{\varphi\mu^{j'}} \\ M^{D\varphi\varphi} & M^{D\varphi D\varphi} & M^{D\varphi\mu} & M^{\varphi\mu^{j'}} \\ M^{\mu\varphi} & M^{\mu D\varphi} & M^{\mu\mu} & M^{\varphi\mu^{j'}} \\ M^{\mu^j\varphi} & M^{\mu^j D\varphi} & M^{\mu^j\mu} & M^{\varphi^j\mu^{j'}} \end{pmatrix} = \begin{pmatrix} \left(a_{ni}^{\varphi\varphi}\right) & \left(a_{ni}^{\varphi D^l\varphi}\right) & \left(a_{ni}^{\varphi\mu}\right) & \left(a_{ni}^{\varphi\mu^{j'}}\right) \\ \left(a_{ni}^{D\varphi^l\varphi}\right) & \left(a_{ni}^{D\varphi^l D\varphi^k}\right) & \left(a_{ni}^{D\varphi^l\mu}\right) & \left(a_{ni}^{D\varphi^l\mu^{j'}}\right) \\ \left(a_{ni}^{\mu\varphi}\right) & \left(a_{ni}^{\mu D\varphi^k}\right) & \left(a_{ni}^{\mu\mu}\right) & \left(a_{ni}^{\mu\mu^{j'}}\right) \\ \left(a_{ni}^{\mu^j\varphi}\right) & \left(a_{ni}^{\mu^j D\varphi^k}\right) & \left(a_{ni}^{\mu^j\mu}\right) & \left(a_{ni}^{\mu^j\mu^{j'}}\right) \end{pmatrix}.
$$

In order to determine the transpose M^T, we isolate the components of the submatrices $M_{ni}^{\cdot\cdot}$ from the right-hand side of system (4.1). All components not listed below will be zero.

$$
\begin{aligned}
m_{ni}^{\varphi\varphi} &= 2\dot{\gamma}_{t,in}\mu_{t,i}(x_{t,i}-x_{t,n})^T\mathrm{Id}_d\\
&\quad + \sum_{j'=1}^{d}\mu_{t,i}^{j'}D^{j'}\varphi_{0t}^v(x_{0,i})^T\big(4\ddot{\gamma}_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot{\gamma}_{t,in}\mathrm{Id}_d\big)\\
&\quad \text{iff } i=n:\\
&\quad \sum_{i'=1}^{n}-2\dot{\gamma}_{t,i'n}\mu_{t,i'}(x_{t,i'}-x_{t,n})^T\mathrm{Id}_d\\
&\quad - \sum_{i'=1}^{n}\sum_{j'=1}^{d}\mu_{t,i'}^{j'}D^{j'}\varphi_{0t}^v(x_{0,i'})^T\big(4\ddot{\gamma}_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T + 2\dot{\gamma}_{t,i'n}\mathrm{Id}_d\big)\\
m_{ni}^{\varphi D\varphi^l} &= \mu_{t,i}^l\nabla_1 K(x_{t,i},x_{t,n})^T\\
m_{ni}^{\varphi\mu} &= \gamma_{t,in}\mathrm{Id}_d\\
m_{ni}^{\varphi\mu^j} &= \nabla_1 K(x_{t,i},x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})\mathrm{Id}_d
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{D\varphi^l\varphi} &= -\mu_{t,i} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot\gamma_{t,in}\mathrm{Id}_d\big)\\
&\quad - \sum_{j'=1}^{N}\mu_{t,i}^{j'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot\gamma_{t,in}D^{j'}\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i})\mathrm{Id}_d + 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,i}))^T\big)\\
&\quad \text{iff } i=n:\\
&\quad \sum_{i'=1}^{n}\mu_{t,i'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot\gamma_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T + 2\dot\gamma_{t,i'n}\mathrm{Id}_d\big)\\
&\quad + \sum_{i'=1}^{n}\sum_{j'=1}^{N}\mu_{t,i'}^{j'} D^l\varphi_{0t}^v(x_{0,n})^T\big(4\ddot\gamma_{t,i'n}D^{j'}\varphi_{0t}^v(x_{0,i'})(x_{t,i'}-x_{t,n})^T\\
&\qquad + 8\dddot\gamma_{t,i'n}(x_{t,i'}-x_{t,n})(x_{t,i'}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i'})(x_{t,i'}-x_{t,n})^T\\
&\qquad + 4\ddot\gamma_{t,i'n}(x_{t,i'}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,i'})\mathrm{Id}_d + 4\ddot\gamma_{t,i'n}(x_{t,i'}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,i'}))^T\big)\\
m_{ni}^{D\varphi^l D\varphi^k} &= \mu_{t,i}^k D^l\varphi_{0t}^v(x_{0,n})^T D_1\nabla_2 K(x_{t,i},x_{t,n})\\
&\quad \text{iff } i=n,\ \text{iff } l=k:\\
&\quad \sum_{i'=1}^{n}\mu_{t,i'}\nabla_2 K(x_{t,i'},x_{t,n})^T + \sum_{i'=1}^{n}\sum_{j'=1}^{d}\mu_{t,i'}^{j'}\big(D_1\nabla_2 K(x_{t,i'},x_{t,n})D^{j'}\varphi_{0t}^v(x_{0,i'})\big)^T\mathrm{Id}_d\\
m_{ni}^{D\varphi^l\mu} &= \nabla_2 K(x_{t,i},x_{t,n})^T D^l\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d\\
m_{ni}^{D\varphi^l\mu^j} &= \big(D_1\nabla_2 K(x_{t,i},x_{t,n})D^j\varphi_{0t}^v(x_{0,i})\big)^T D^l\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu\varphi} &= (\mu_{t,n}^T\mu_{t,i})\big(4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T + 2\dot\gamma_{t,in}\mathrm{Id}_d\big)\\
&\quad - \sum_{j'=1}^{d}\big(\mu_{t,n}^{j',T}\mu_{t,i} - \mu_{t,n}^T\mu_{t,i}^{j'}\big)\big(4\ddot\gamma_{t,in}D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})\mathrm{Id}_d + 4\ddot\gamma_{t,in}(x_{t,i}-x_{t,n})(D^{j'}\varphi_{0t}^v(x_{0,n}))^T\big)\\
&\quad - \sum_{j,j'=1}^{d}\big(\mu_{t,n}^{j',T}\mu_{t,i}^{j}\big)\big(8\dddot\gamma_{t,in}D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})D^j\varphi_{0t}^v(x_{0,i})^T D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot\gamma_{t,in}D^j\varphi_{0t}^v(x_{0,i})D^{j'}\varphi_{0t}^v(x_{0,n})^T + 4\ddot\gamma_{t,in}D^j\varphi_{0t}^v(x_{0,i})^T D^{j'}\varphi_{0t}^v(x_{0,n})\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 4\ddot\gamma_{t,in}D^{j'}\varphi_{0t}^v(x_{0,n})D^j\varphi_{0t}^v(x_{0,i})^T\\
&\qquad + 16\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T\\
&\qquad\qquad D^{j'}\varphi_{0t}^v(x_{0,n})(x_{t,i}-x_{t,n})^T\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})D^j\varphi_{0t}^v(x_{0,i})^T\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})^T D^{j'}\varphi_{0t}^v(x_{0,n})\\
&\qquad + 8\dddot\gamma_{t,in}(x_{t,i}-x_{t,n})^T D^j\varphi_{0t}^v(x_{0,i})(x_{t,i}-x_{t,n})D^{j'}\varphi_{0t}^v(x_{0,n})^T\\
&\qquad \big)
\end{aligned}
$$

iff $i = n$:

$$
\begin{aligned}
&- \sum_{i'=1}^{N} (\mu_{t,n}^{T}\mu_{t,i'})\big(4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^{T} + 2\dot{\gamma}_{t,i'n}\mathrm{Id}_d\big) \\
&+ \sum_{i'=1}^{N}\sum_{j'=1}^{d} \big(\mu_{t,n}^{j',T}\mu_{t,i'} - \mu_{t,n}^{T}\mu_{t,i'}^{j'}\big)\big(4\ddot{\gamma}_{t,i'n}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})\mathrm{Id}_d \\
&\quad + 4\ddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(D^{j'}\varphi_{0t}^{v}(x_{0,n}))^{T}\big) \\
&+ \sum_{i'=1}^{N}\sum_{j,j'=1}^{d} \big(\mu_{t,n}^{j',T}\mu_{t,i'}^{j}\big)\big(\\
&\quad 8\dddot{\gamma}_{t,i'n}D^{j}\varphi_{0t}^{v}(x_{0,i'})(x_{t,i'} - x_{t,n})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 4\ddot{\gamma}_{t,i'n}D^{j}\varphi_{0t}^{v}(x_{0,i'})D^{j'}\varphi_{0t}^{v}(x_{0,n})^{T} \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})D^{j}\varphi_{0t}^{v}(x_{0,i'})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 4\ddot{\gamma}_{t,i'n}D^{j}\varphi_{0t}^{v}(x_{0,i'})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n}) \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^{T}D^{j}\varphi_{0t}^{v}(x_{0,i'})D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 4\ddot{\gamma}_{t,i'n}D^{j'}\varphi_{0t}^{v}(x_{0,n})D^{j}\varphi_{0t}^{v}(x_{0,i'})^{T} \\
&\quad + 16\ddddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^{T}D^{j}\varphi_{0t}^{v}(x_{0,i'})(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^{T} \\
&\qquad\qquad D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{t,i'} - x_{t,n})^{T} \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})(x_{t,i'} - x_{t,n})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})D^{j}\varphi_{0t}^{v}(x_{0,i'})^{T} \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^{T}D^{j}\varphi_{0t}^{v}(x_{0,i'})(x_{t,i'} - x_{t,n})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n}) \\
&\quad + 8\dddot{\gamma}_{t,i'n}(x_{t,i'} - x_{t,n})^{T}D^{j}\varphi_{0t}^{v}(x_{0,i'})(x_{t,i'} - x_{t,n})D^{j'}\varphi_{0t}^{v}(x_{0,n})^{T} \\
&\quad)
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu D\varphi^{l}} = &- \sum_{j'=1}^{d} 4\big(\mu_{t,n}^{j',T}\mu_{t,i}^{l}\big)\big(\ddot{\gamma}_{t,in}(x_{i,t} - x_{n,t})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})\mathrm{Id}_d \\
&\quad + \ddot{\gamma}_{t,in}(x_{i,t} - x_{n,t})D^{j'}\varphi_{0t}^{v}(x_{0,n})^{T} \\
&\quad + \ddot{\gamma}_{t,in}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{i,t} - x_{n,t})^{T} \\
&\quad + 2\dddot{\gamma}_{t,in}(x_{i,t} - x_{n,t})(x_{i,t} - x_{n,t})^{T}D^{j'}\varphi_{0t}^{v}(x_{0,n})(x_{i,t} - x_{n,t})^{T}\big)
\end{aligned}
$$

iff $i = n$:

$$
\begin{aligned}
&- \sum_{i'=1}^{N} \big(\mu_{t,n}^{l,T}\mu_{t,i'} - \mu_{t,n}^{T}\mu_{t,i'}^{l}\big)D_2\nabla_2 K(x_{t,i'}, x_{t,n}) \\
&- \sum_{i'=1}^{N}\sum_{j'=1}^{d} \big(\mu_{t,n}^{l,T}\mu_{t,i'}^{j'}\big)D_2\big(D_1\nabla_2 K(x_{t,i'}, x_{t,n})D^{j'}\varphi_{0t}^{v}(x_{0,i'})\big)
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu\mu} &= -\nabla_2 K(x_{t,i}, x_{t,n})\mu_{t,n}^T - \sum_{j'=1}^{d} D_2\nabla_2 K(x_{t,i}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,n}^{j',T} \\
&\quad \text{iff } i = n: \\
&\quad - \sum_{i'=1}^{N} \nabla_2 K(x_{t,i'}, x_{t,n})\mu_{t,i'}^T + \sum_{i'=1}^{N}\sum_{j'=1}^{d} D_2\nabla_2 K(x_{t,i'}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^{j',T} \\
m_{ni}^{\mu\mu^j} &= +D_2\nabla_2 K(x_{t,i}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,n}^T \\
&\quad - \sum_{j'=1}^{d} D_2\big(D_1\nabla_2 K(x_{t,i}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,i})\big) D^{j'}\varphi_{0t}^v(x_{0,n})\mu_{t,n}^{j',T} \\
&\quad \text{iff } i = n: \\
&\quad - \sum_{i'=1}^{N} D_2\nabla_2 K(x_{t,i'}, x_{t,n}) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^T \\
&\quad - \sum_{i'=1}^{N}\sum_{j'=1}^{d} D_2\big(D_1\nabla_2 K(x_{t,i'}, x_{t,n}) D^{j'}\varphi_{0t}^v(x_{0,i'})\big) D^j\varphi_{0t}^v(x_{0,n})\mu_{t,i'}^{j',T}
\end{aligned}
$$

$$
\begin{aligned}
m_{ni}^{\mu^j\varphi} &= -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'}\varphi} \\
m_{ni}^{\mu^j D\varphi^l} &= -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'} D\varphi^l} \\
&\quad \text{iff } i = n: \\
&\quad D\varphi_{0t}^v(x_{0,n})^{-1,T} e_l\big(D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\mu_{t,n}^j\big)^T \\
m_{ni}^{\mu^j\mu} &= -\sum_{j'=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j'}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j'}\mu} \\
m_{ni}^{\mu^j\mu^{j'}} &= -\sum_{j''=1}^{d} D\varphi_{0t}^v(x_{0,i})^{-1,T} e_{j''}\mu_{t,n}^{j,T} m_{ni}^{D\varphi^{j''}\mu^{j'}} \\
&\quad \text{iff } i = n,\ j = j': \\
&\quad - D\varphi_{0t}^v(x_{0,n})^{-1,T}\partial_t D\varphi_{0t}^v(x_{0,n})^T\ .
\end{aligned}
$$

As described in Section 4, the gradient at $t = 0$ can then be obtained by solving the system

$$y_t = M_t^T y_t$$

backwards in time, confer also [32, p. 281].

REFERENCES

[1] Vincent Arsigny, Olivier Commowick, Nicholas Ayache, and Xavier Pennec, *A fast and Log-Euclidean polyaffine framework for locally linear registration*, J. Math. Imaging Vis., 33 (2009), pp. 222–238.

[2] Vincent Arsigny, Olivier Commowick, Xavier Pennec, and Nicholas Ayache, *A Log-Euclidean framework for statistics on diffeomorphisms*, in MICCAI 2006, 2006, pp. 924–931.

[3] 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 (2005), pp. 507–523.

[4] M. Faisal Beg, Michael I. Miller, Alain Trouvé, and Laurent Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, IJCV, 61 (2005), pp. 139–157.

[5] F. L. Bookstein, *Linear methods for nonlinear maps: Procrustes fits, thin-plate splines, and the biometric analysis of shape variability*, in Brain warping, Academic Press, 1999, pp. 157–181.

[6] Yan Cao, M. I Miller, R. L Winslow, and L. Younes, *Large deformation diffeomorphic metric mapping of vector fields*, IEEE Transactions on Medical Imaging, 24 (2005), pp. 1216–1230.

[7] GE Christensen, RD Rabbitt, and MI Miller, *Deformable templates using large deformation kinematics*, Image Processing, IEEE Transactions on, 5 (2002).

[8] Colin J Cotter and Darryl D Holm, *Singular solutions, momentum maps and computational anatomy*, nlin/0605020, (2006).

[9] Sune Darkner and Jon Sporring, *Generalized partial volume: An inferior density estimator to parzen windows for normalized mutual information*, in Information Processing in Medical Imaging, Gábor Székely and Horst K. Hahn, eds., vol. 6801, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 436–447.

[10] Paul Dupuis, Ulf Grenander, and Michael I Miller, *Variational problems on flows of diffeomorphisms for image matching*, (1998).

[11] Stanley Durrleman, Marcel Prastawa, Guido Gerig, and Sarang Joshi, *Optimal data-driven sparse parameterization of diffeomorphisms for population analysis*, Information Processing in Medical Imaging: Proceedings of the ... Conference, 22 (2011), pp. 123–134. PMID: 21761651.

[12] Gregory E. Fasshauer and Qi Ye, *Reproducing kernels of generalized sobolev spaces via a green function approach with distributional operators*, Numerische Mathematik, (2011).

[13] N. C. Fox, E. K. Warrington, P. A. Freeborough, P. Hartikainen, A. M. Kennedy, J. M. Stevens, and M. N. Rossor, *Presymptomatic hippocampal atrophy in alzheimer's disease*, Brain, 119 (1996), pp. 2001 –2007.

[14] Ulf Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford University Press, USA, Feb. 1994.

[15] Monica Hernandez, Matias Bossa, and Salvador Olmos, *Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows*, International Journal of Computer Vision, 85 (2009), pp. 291–306.

[16] Clifford R. Jack, Ronald C. Petersen, Yue Cheng Xu, Stephen C. Waring, Peter C. O'Brien, Eric G. Tangalos, Glenn E. Smith, Robert J. Ivnik, and Emre Kokmen, *Medial temporal atrophy on MRI in normal aging and very mild alzheimer's disease*, Neurology, 49 (1997), pp. 786 –794.

[17] Daniel S Marcus, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner, *Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults*, Journal of Cognitive Neuroscience, 22 (2010), pp. 2677–2684. PMID: 19929323.

[18] X. Pennec, R. Stefanescu, V. Arsigny, P. Fillard, and N. Ayache, *Riemannian elasticity: A statistical regularization framework for non-linear registration*, in MICCAI 2005, 2005, pp. 943–950.

[19] J. P.W Pluim, J. B.A Maintz, and M. A Viergever, *Image registration by maximization of combined mutual information and gradient information*, IEEE Transactions on Medical Imaging, 19 (2000), pp. 809–814.

[20] Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache, *The correlation ratio as a new similarity measure for multimodal image registration*, in Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI '98, Springer-Verlag, 1998, p. 1115–1124. ACM ID: 709612.

[21] Alexis Roche, Xavier Pennec, Grégoire Malandain, and Nicholas Ayache, *Rigid registration of 3D ultrasound with MR images: a new approach combining intensity and gradient information*, IEEE Transactions on Medical Imaging, 20 (2001), pp. 1038—1049.

[22] D Rueckert, L I Sonoda, C Hayes, D L Hill, M O Leach, and D J Hawkes, *Nonrigid registration using free-form deformations: application to breast MR images*, IEEE Transactions on Medical Imaging, 18 (1999), pp. 712–721. PMID: 10534053.

[23] Christof Seiler, Xavier Pennec, and Mauricio Reyes, *Geometry-Aware multiscale image registration via OBBTree-Based polyaffine Log-Demons*, in Medical Image Computing and

Computer-Assisted Intervention - MICCAI, Toronto, Canada, 2011.
[24] Stefan Sommer, Mads Nielsen, Francois Lauze, and Xavier Pennec, *A Multi-Scale kernel bundle for LDDMM: towards sparse deformation description across space and scales*, in IPMI 2011, Springer, 2011.
[25] Stefan Sommer, M. Nielsen, and X. Pennec, *Sparsity and scale: Compact representations of deformation for diffeomorphic registration*, in MMBIA at WACV 2012, 2012.
[26] J.-P. Thirion, *Image matching as a diffusion process: an analogy with maxwell's demons*, Medical Image Analysis, 2 (1998), pp. 243–260.
[27] Alain Trouvé, *An infinite dimensional group approach for physics based models in patterns recognition*, 1995.
[28] M. Vaillant, M.I. Miller, L. Younes, and A. Trouvé, *Statistics on diffeomorphisms via tangent space representations*, NeuroImage, 23 (2004), pp. S161–S169.
[29] Tom Vercauteren, Xavier Pennec, Aymeric Perchant, and Nicholas Ayache, *Diffeomorphic demons: efficient non-parametric image registration*, NeuroImage, 45 (2009), pp. 61–72.
[30] William M Wells, III, Paul Viola, and Ron Kikinis, *Multi-Modal volume registration by maximization of mutual information*, (1996).
[31] Younes, *Constrained diffeomorphic shape evolution*, submitted to foundations Comp Math, (2011).
[32] Laurent Younes, *Shapes and Diffeomorphisms*, Springer, 2010.
[33] Ding-Xuan Zhou, *Derivative reproducing properties for kernel methods in learning theory*, Journal of Computational and Applied Mathematics, 220 (2008), pp. 456–463.
[34] Xiahai Zhuang, S. Arridge, D. J Hawkes, and S. Ourselin, *A nonrigid registration framework using spatially encoded mutual information and Free-Form deformations*, IEEE Transactions on Medical Imaging, 30 (2011), pp. 1819–1828.

4.

Paper #3: *Accelerating Multi-Scale Flows for LDDKBM Diffeomorphic Registration*

Peer-reviewed conference paper accepted for oral presentation at the GPUCV workshop at ICCV 2011, Barcelona, Spain, 2011.

Authors:
Stefan Sommer

Notes:

With the goal of reducing computation time with multiple scales, we present a GPU implementation of the kernel bundle landmark registration algorithm. The structure of the algorithm and the massively parallel processors enable a two orders of magnitude speedup over a single threaded CPU implementation. In essence, this shows that mathematically well-founded and computationally heavy algorithms can be used in practice. In the paper, we refer to the kernel bundle framework by the abbreviation *LDDKBM*.

Accelerating Multi-Scale Flows for LDDKBM Diffeomorphic Registration

Stefan Sommer
Department of Computer Science, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen E, Denmark
sommer@diku.dk

Abstract

Registrations in medical imaging and computational anatomy can be obtained using the Large Deformation Diffeomorphic Kernel Bundle Mapping (LDDKBM) framework. This provides a registration algorithm with a solid mathematical foundation while incorporating regularization of deformation at multiple scales. Because the variational formulation of LDDKBM implies a heavy computational burden in the search for optimal registrations, exploiting every possibility for faster computation will improve the usability of the algorithm. We present a parallelization strategy using the multi-scale structure and show that the parallelized method constitutes an example of how the processing power of GPUs can massively reduce the running time: after moving the computation to the GPU, we achieve a two order of magnitude speedup over a single-threaded CPU implementation. Not only does this significantly reduce the cost of using multiple scales, it also allows the algorithm to be used on much larger datasets.

1. Introduction

Registration, finding smooth, one-to-one mappings between landmarks, images, surfaces, or tensors, constitute an important task in medical imaging and computational anatomy. Examples include using image intensity to register scanned brains to an already segmented reference brain, or using sets of landmarks, manually or automatically annotated, to drive the registration of lungs in different phases of the respiratory process.

Much research in registration thrives to create algorithms which produce good matches in reasonable time while having a strong mathematical foundation and plausible model of deformation. The latter properties are important to ensure convergence and existence of optimal solutions as well as allowing meaningful statistics to be performed on the registration results. This is in particular important when using statistical techniques to search for patterns in the data and developing biomarkers in order to ensure actual properties of the data are measured instead of artifacts of the registration algorithm.

The LDDMM framework [16] and the multi-scale LDDKBM extension [12] provide the benefit of having strong mathematical foundations while performing well in applications. However, the well-founded and physically inspired models comes with the cost of heavy computational requirements which necessitates exploiting every possibility for faster computation. This paper presents a strategy for GPU implementation of the multi-scale LDDKBM algorithm for landmark registration. We show how the cost of multiple scales can be eased by utilizing the decoupled structure of the problem, and we present benchmarks evaluating the actual implementations. As we will show, the GPU implementation achieves two orders of magnitude speedup for the computationally most intensive part of the algorithm allowing the LDDKBM method to be used on much larger datasets with increased number of scales.

1.1. Related Work

Besides LDDMM and LDDKBM, many methods for non-rigid registration are currently used for regularization. Examples include elastic methods [10], parametrizations using static velocity fields [1] and the demons algorithm [13, 15]. The deformable template model pioneered by Grenander in [7] and the flow approach by Christensen et al. [5] was paramount in the development of LDDMM together with the theoretical contributions of Dupuis et al. and Trouvé [6, 14]. Algorithms for computing optimal diffeomorphisms have been developed in [2]. The LDDKBM multi-scale extension of LDDMM was introduced in [12] with the evolution equations for optimal registrations presented in [11]. The two-scale case was in addition developed in [3].

GPU implementation of algorithms for image registration in the LDDMM framework has been described in [9, 8]

with freely available source code.[1] The algorithm developed here differs in targeting the multi-scale LDDKBM framework as well as being applicable to landmarks instead of images. Dealing explicitly with the scale-structure is important in order to lessen the speed penalty of including multiple scales. Moreover, the different structure of the landmark algorithm results in a problem with greater focus on computational power than the memory bound image case.

1.2. Content and Outline

We start by a brief introduction to the registration problem and the LDDKBM framework before describing the flow equations for the landmark case. The backwards flow constitute the computationally most intensive part of the optimization, and we describe the structure of the computation of the time-step update before parallelizing the problem and detailing a GPU implementation. We end the paper with benchmarks and conclusion. The paper thus contributes by

(1) describing how registration of landmarks in the LDDKBM framework can be solved using the forwards and backwards flow equations,

(2) presenting a strategy for parallelizing the backwards time-step update,

(3) providing a fast GPU implementation of the algorithm,

(4) and giving benchmarks showing how the problem scales with input size and number of scales and how well the GPU algorithm performs compared to a CPU implementation.

2. LDDKBM Diffeomorphic Registration

The Large Deformation Diffeomorphic Kernel Bundle Mapping framework (LDDKBM) extends the single-scale LDDMM framework by allowing regularization at multiple-scales to be used in the registration. We give a brief overview of the registration problem and how it is treated in LDDKBM. For further details, we refer to the paper [12] introducing LDDKBM and the monograph [16] with extensive details on LDDMM.

Registration of geometric objects is often performed by defining an action of diffeomorphisms on the objects before searching for diffeomorphisms matching the objects through the action. For example, in order to register landmarks $x_1, \dots, x_N$ and $y_1, \dots, y_N$ in $\mathbb{R}^d$, $d = 2, 3$, we search for a diffeomorphism $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ such that $\varphi(x_i) = y_i$. Equivalently, if we wish register images I_0 and I_1, we search for φ such that $I_0 \circ \varphi = I_1$. Frequently, a perfect match is not possible or even not desirable because noisy data may force the diffeomorphism to be highly irregular. Instead, the problem is stated in a variational form as a search for φ minimizing

$$E(\varphi) = E_1(\varphi) + \lambda U(\varphi) \tag{1}$$

where $E_1(\varphi)$ is a regularization measure, $U(\varphi)$ a measure of the quality of the match, and $\lambda > 0$ a weight. A simple and often used choice for U is the L^2-error which takes the form $U(\varphi) = \sum_{i=1}^{N} \|\varphi(x_i) - y_i\|^2$ for landmarks. In the LDDKBM framework, the regularization measure $E_1(\varphi)$ is defined as the minimum energy of paths of diffeomorphisms transporting the identity Id_Ω to φ, i.e.

$$E_1(\varphi) = \min_{w_t \in W, \varphi_{01}^{\Psi(w)} = \varphi} \int_0^1 \|w_s\|_W^2 \, ds \tag{2}$$

with $\varphi_{0t}^{\Psi(w)}$ denoting the path starting at Id_Ω with time-derivative $\partial_t \varphi_{0t}^{\Psi(w)} = \Psi(w_t) \circ \varphi_{0t}^{v}$. The space W is denoted the kernel bundle and consist of a family of vector spaces V_r parameterized by r, the scale. Each vector space V_r can be considered a subset of a tangent space V of a suitable Lie group of diffeomorphisms, and a map Ψ collects parts w_r of a bundle vector $w \in W$ at each scale r to one derivative vector in V by integration $\Psi(w) = \int_{I_W} w_r \, dr$. The norm $\|\cdot\|_{V_r}$ on each V_r is allowed to vary with r, and the bundle norm $\|\cdot\|_W$ is defined by

$$\|w\|_W^2 = \int_{I_W} \|w_r\|_{V_r}^2 dr \, ,$$

i.e., the integral of the energy over all scales.

This bundle norm is chosen to penalize highly varying paths while allowing variation at different scales to be penalized differently. In short, a low value of $E_1(\varphi)$ implies that the path to reach φ, and hence φ itself, is regular.

2.1. Optimization

Optimal paths for (2) are governed by the KB-EPDiff equations which extends the EPDiff equations for LDDMM [11]. These evolution equations assert that the bundle velocity w_0 of the path at time $t = 0$ changes in a specific way throughout the evolution of the path from $t = 0$ to its end at $t = 1$. This property is denoted momentum conservation, and it allows a search for a φ minimizing (1) to be phrased in terms of the initial bundle velocity: if we assume $\varphi = \varphi_{01}^{\Psi(w)}$ then the values of both $E_1(\varphi)$ and $U(\varphi)$ are determined by w_0 and we optimize

$$E(w_0) = E_1(w_0) + \lambda U(w_0) \tag{3}$$

instead of (1). In practice, this can be done by giving an initial guess for w_0, calculating the gradient $\nabla E(w_0) = \nabla E_1(w_0) + \lambda \nabla U(w_0)$, and updating w_0 in a gradient descent or similar fashion.

[1] See http://www.sci.utah.edu/software/13/370-atlaswerks.html

2.2. The Gradient: Integrating the Flows

The gradient necessary for optimizing (3) can be computed using a two step algorithm: the initial bundle velocity w_0 is transported forward in time to obtain the diffeomorphism φ before flowing the gradient at time $t = 1$ backwards to obtain the gradient $\nabla E(w_0)$ at $t = 0$.

For N landmarks $x_1, \ldots, x_N$, the KB-EPDiff equations governing the forward integration take the form

$$\begin{aligned} \Psi(w_t) &= \int_{I_W} \textstyle\sum_{l=1}^{N} K_r(\cdot, x_{t,l}) a_{t,r,l} dr \\ \tfrac{d}{dt} a_{t,r,i} &= -\left(\int_{I_W} \textstyle\sum_{l=1}^{N} D_1 \left(K_s(x_{t,i}, x_{t,l}) a_{t,s,l} \right)^T ds \right) a_{t,r,i} \\ x_{t,i} &= \varphi_{0t}^{\Psi(w)}(x_{0,i}) \, . \end{aligned} \tag{4}$$

The points $x_{t,i}$ denote the particle positions at time t and the set of time-dependent vectors $a_{t,r,i}$ is the momentum of the flow. The vectors have components at each scale and are connected to the bundle velocity $w_{t,i,r}$ trough the kernels $K_r(\cdot, \cdot)$ as expressed by the first evolution equation. The choice of kernels affects the regularization of the deformation; often used choices are Gaussian kernels $K_r(x, y) = \exp(\frac{\|x-y\|^2}{r^2})\mathrm{Id}_d$, which we will use in the rest of the paper. The system (4) is a non-linear ODE and finite if the set of scales I_W is finite. In practice, I_W is a discretization $\{s_1, \ldots, s_R\}$ of an interval $[s_1, s_R]$ using R scalars. The system can be integrated using standard Runge-Kutta integrators such as matlabs `ode45` solver.

Since φ is determined by w_0, which through the evolution of w_t is uniquely linked to w_1, $U(\varphi)$ is determined by w_1. The gradient $\nabla U(w_1)$ is usually known; if U measures the L^2-error, the gradient is just the vector with the ith component being $2(x_{1,i} - y_i)$ where y_i are the target points. To perform gradient descent using w_0, we need the gradient $\nabla U(w_0)$ which can be obtained by differentiating (4) and solving the transpose system backwards.[2] The gradient $\nabla E_1(w_0)$ can be solved simultaneously by adding it to the backwards ODE. Combined, the gradient $\nabla E(w_0)$ can be found as the solution at $t = 0$ of an affine, non-autonomous ODE

$$\dot{y}_t = v_t + M_t y_t \tag{5}$$

integrated from $t = 1$ to $t = 0$. The linear component transports $\nabla U(w_t)$ while the affine component transport $\nabla E_1(w_t)$. A complete derivation of this system is out of scope of this paper; it can be found in the upcoming journal version of [12].

While computing the right-hand side of the system (4) has complexity $O(N^2 \cdot R)$ with N the number of landmarks and R the number of scales, computing the right hand side of the system (5) has complexity $O(N^2 \cdot R^2)$. This computation, we denote it the time-step update, makes the backward integration the computationally most intensive parts of the optimization process, and, therefore, we wish to parallelize the time-step update and accelerate it using GPU hardware.

[2] Confer [16] for a description of this method in the LDDMM case.

3. Parallelization and GPU Implementation

In order to accelerate the integration of the system (5), we aim for producing fast procedures for the time-step update $v_t + M_t y_t$. We first describe the CPU procedures in order to identify options for parallelization before giving details on the GPU implementation and thread grid layout. Note that we avoid using the otherwise standard term kernel for the GPU code units in order to avoid confusion with the Gaussian kernels $K_r(\cdot, \cdot)$ in the LDDKBM framework.

The matrix M_t and the affine component v_t depend on the momenta $a_{t,r,i}$ and particle positions $x_{t,i}$ resulting in the system being non-autonomous. The matrix M_t is not, however, explicitly generated. Instead, the product $M_t y_t$ is evaluated as a sequence of nested loops over particles and scales saving the time to first store and later retrieve the large matrix. The main part of the computation in each loop iteration consists of evaluating the Gaussian kernels $K_r(\cdot, \cdot)$ and computing a sequence of simple floating point operation (flops).

The computation of the update is split into two procedures: the first updates the differential of the particle positions requiring computation of $N \cdot d$ scalars in the output vector with d usually being 3 for 3D registration. The second procedure updates the differential of the momenta. Since the momentum is split over both particles and scales, this procedure updates $N \cdot R \cdot d$ scalars. The structure of the procedures is shown in Algorithm 1 and 2. It is clear that both procedures has complexity $O(N^2 \cdot R)$ in the computation of the kernels, and $O(N^2 \cdot R^2)$ in the computation of the additional flops.

3.1. Bottlenecks and Parallelization Strategy

Matrix-vector products are usually memory-bound operations since every item of the matrix needs to be retrieved from memory while a relatively small amount of calculation is needed for each item. A stored matrix approach to the current problem would imply a memory-access complexity of $O(N^2 \cdot R^2)$ which would dominate the execution time. However, since the matrix is computed as needed as a function of the particle positions and momentum at the given time, the computation need only refer to $O(N \cdot R)$ locations in memory. Combined with the fact that the exponentials needed to compute the Gaussian kernels are relatively expensive to compute and the quadratic scaling in R for the additional flops, the operation is instead primarily computationally bound. This property makes a parallelized implementation ideal for using the processing power provided by

Algorithm 1 Update particles, CPU

```
for i = 1 → N do                          ▷ updating particle i
    for l = 1 → N do                      ▷ loop, all particles
        for s = 1 → R do                  ▷ loop, all scales
            compute and store kernels
        end for
        for s1 = 1 → R do                 ▷ loop, all scales twice
            for s2 = 1 → R do
                retrieve kernels
                compute additional flops
                sum results
            end for
        end for
    end for
    update particle i in output array
end for
```

Algorithm 2 Update momenta, CPU

```
for i = 1 → N do                          ▷ updating momentum (i, si)
    for si = 1 → R do
        for l = 1 → N do                  ▷ loop, all particles
            if si = 1 then                ▷ loop, all scales once per l
                for s = 1 → R do
                    compute and store kernels
                end for
            end if
            for s = 1 → R do              ▷ loop, all scales
                retrieve kernels
                compute additional flops
                sum results
            end for
        end for
        update momentum (i, si) in output array
    end for
end for
```

the large number of cores in GPUs.

The simplest GPU implementation would consist in creating a thread for each particle indexed by i in the first outer loop of the first procedure resulting in N threads total. For the second procedure, creating a thread for each pair (i, si) would result in $N \cdot R$ threads total. However, with a number of particles of up to 300 for the lung dataset we later use for benchmarks, such a strategy would result in poor utilization of the computational units in a fast GPU and would not properly offset time spent on memory access.

For each particle, the procedures compute sums during the loop iteration over particles and scales. Splitting these sums over multiple threads and reducing afterwards constitutes an obvious optimization. We employ this standard strategy to optimize both procedures reducing over both particles and scales. In addition, we cache data in thread group shared memory and split the task of loading data from global memory to group shared memory among the threads. This reduces the number of memory access stalls for the slower global memory while allowing the threads to retrieve data quickly from the faster shared memory.

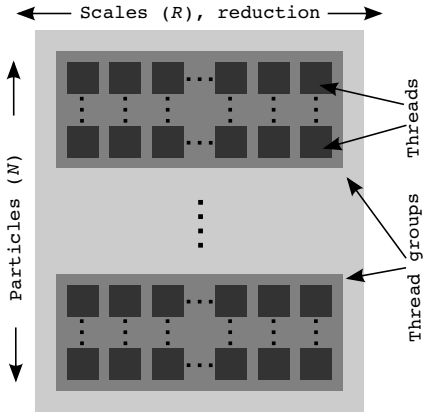


Figure 1. Thread grid layout for the GPU procedures.

The computation of the kernels over scale is the same for all pairs of particles i and l. This allows for splitting the computation of the kernels over threads in a thread group. After a within-group sync, the threads can retrieve the computed values from memory. This decoupling of scales offsets the lower number of compute cores for special mathematical operations and allow a significant speedup. It should also be noted that thread group synchronization within loops can lead to poor performance in some circumstances. This can be avoided by switching the order of the loops though doing so would require reordering of the input arrays in order to ensure coalesced memory access. However, for the problem at hand, the synchronization does not significantly affect performance.

3.2. GPU Implementation

In the optimized versions, a thread for each pair of integers (i, si, k), $i = 1, \ldots, N$, $si = 1, \ldots, R$, $k = 1, \ldots,$ NrRed, is created for both part of the updates, where NrRed control the number of threads working in parallel on the innermost loops. The resulting procedures are shown in Algorithm 3 and 4. Storing in group local memory the computed kernels and the results of each split of the inner loops before reducing is paramount for the fast execution of the GPU procedures. We take care in ordering the accesses to the local memory to ensure coalesced access and use padding of the arrays in order to avoid bank conflicts.

We create a two-dimensional thread grid layout for both kernels with scale and splits of the inner loops along the first dimension and particles along the second. Since the number of scales is usually relatively small, we can cover the first di-

Algorithm 3 Update particle i, GPU

```
for l = k → k + chunk size do                    ▷ loop, particles
    compute kernel si
    retrieve data from global memory
    save kernel and data to thread group local memory
    sync threads in thread group
    for s = 1 → R do                             ▷ loop, all scales
        retrieve kernel s and data from local memory
        compute additional flops
        sum results
    end for
end for
save to thread group local memory
sync threads in thread group
if selected threads then
    reduce over saved results
    update particle i in output array
end if
```

Algorithm 4 Update momentum (i, si), GPU

```
for l = k → k + chunk size do                    ▷ loop, particles
    compute kernel si
    retrieve data from global memory
    save kernel and data to thread group local memory
    sync threads in thread group
    for s = 1 → R do                             ▷ loop, all scales
        retrieve kernel s and data from local memory
        compute additional flops
        sum results
    end for
end for
save to thread group local memory
sync threads in thread group
if selected threads then
    reduce over saved results
    update momentum (i, si) in output array
end if
```

mension with one thread group. A number of thread groups is then needed to cover the entire grid along the second dimension. Here we have some freedom in choosing the actual number of particles covered by each group with the upper limit determined by the maximum number of threads per group and registers per multiprocessor supported by the GPU. For a given number of scales, we experimentally determine the optimal value which is usually the maximum allowed. The thread grid layout is illustrated in Figure 1.

4. Benchmarks: Towards Faster Registration

We perform benchmarks on the computation of the backwards integration time-step update, the computationally most intensive part of the registration algorithm. The dataset [4] consists of annotated landmarks on CT images of different stages of the lung respiratory phases for five patient. Details on the setup can be found in [12]. For each patient, 300 landmarks are available, which is close to the maximum data size allowing registrations to finish within a reasonable time on conventional hardware. In addition, in order to simulate computations on larger datasets which the faster algorithms now allow, we use artificially generated particles.

System 1
4 x Intel Xeon E5520 (quad core) @ 2.27GHz, 32Gb
2 x GeForce GTX 590, 3072Mb
4 x 512 cores @ 607Mhz

System 2
Intel Core 2 Quad Q9450 @ 2.66GHz, 8Gb
3 x Nvidia GeForce GTX 295, 2 x 895Mb
6 x 240 cores @ 576Mhz

Table 1. The two systems used for benchmarking.

The benchmarks will be performed on two systems, confer Table 1. System 1 contains two GeForce GTX590 cards each having two GPU units, while system 2 has three GeForce GTX295 with a total of 6 GPU units. To keep the timings comparable, we evaluate the algorithm running on a single GPU unit against a single-threaded CPU implementation. It is straightforward to split the problem over multiple GPU units and multiple CPU cores with good scaling for low number of units and cores. Thus, the reported timings can to some extend be translated to real performance by dividing by the number of CPU cores and GPU units, respectively. As an example with a higher number of cores, when using all 16 CPU cores of System 1, an OpenMP parallelization resulted in a 10 times speedup for the CPU implementation. Using all available 4 GPU units of system 1 will similarly improve the GPU implementation.

The timings are reported for CUDA[3] implementations of Algorithm 3 and 4 though OpenCL[4] versions have been implemented as well with similar performance. In contrast to the linear scaling of the size of the data needed to be transfered to the GPU memory, the time-step update scales quadratically in both number of particles and scales. Therefore, the time spent on host memory to GPU memory transfers plays an insignificant role compared to the time spent on the actual computation. Optimizing the memory access structure of the CPU implementation in order to improve cache performance is not perceived here; in rough terms, the CPU implementation follows Algorithm 1 and 2.

[3] http://www.nvidia.com/object/cuda_home_new.html
[4] http://www.khronos.org/opencl/

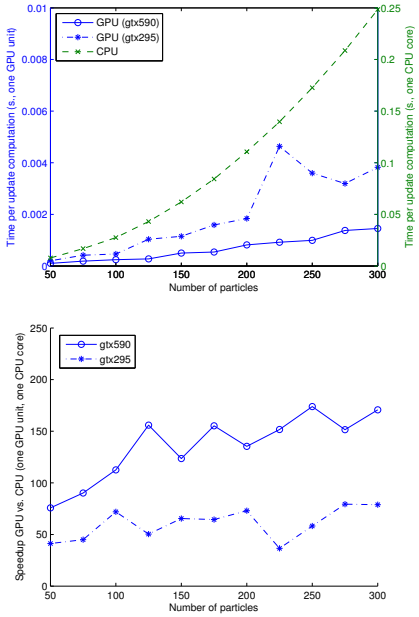


Figure 2. Running time (seconds) and speedup for both systems with between 50 and 300 particles of the lung dataset. CPU running time is reported for system 1. Note the different axes for the GPU and CPU timings. The running time increases quadratically and the speedup of the GPU cores increases with data size.

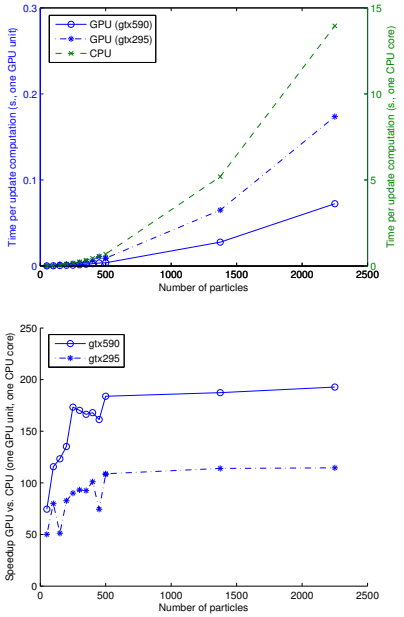


Figure 3. Running time (seconds) and speedup with artificial data. CPU running time is reported for system 1. The increase in running time is quadratic with the speedup curves leveling out (close to 200 times speedup for system 1 and more than 100 times for system 2).

In Figure 2, the time spent for each computation of the time-step update with 5 scales is plotted against the number of particles included from the lung dataset. The quadratic scaling is clear for the CPU implementation. There is some variance in the effectiveness of the GPU implementations as expected from the different utilization of the computational cores for different data sizes. The GPU vs. CPU speedup plot shows increasing benefit of using the GPUs with increasing data size.

Since the speedup curves with the lung data do not level out, it is not clear that the computational power of the GPUs are fully utilized for this dataset. We increase the data size with randomly generated particles and plot the results, again with 5 scales, in Figure 3. The speedup curves for both systems reach plateaus showing reduction in running time of close to 200 for system 1 and slightly more than 100 for system 2. Compared to the fact that the theoretical peak performance of one GTX590 unit is roughly 1.4 times the peak performance of one GTX295 unit (1244 GFLOPS vs. 894 GFLOPS), this gives some indication that the running time correlates with the hardware capability. With large number of particles, we achieve approximately 145 GFLOPS for the simple floating point operations and approximately 0.7e9 evaluations of the Gaussian kernels per second.

Figure 4 shows how the number of scales correspond to the running time with fixed data size (1000 particles). The graphs increase quadratically in the number of scales, though with slower growth for the GPU for a low number of scales. This is most likely a combination of increased utilization of the hardware for the increasing computational load and the fact that the computation of the Gaussian kernels scales linearly in the number of scales. It should be noted that for practical purposes, including more than 16 scales in the registration is hardly useful.

5. Conclusion and Outlook

We have implemented and tested a LDDKBM landmark registration algorithm on GPU hardware and shown that a two orders of magnitude speedup is achievable on the most time intensive part of the algorithm. The result allows the LDDKBM framework to be applied to much larger datasets in practice, and it allows the benefits of including scales in the registration to coincide with fast computation of the optimal registration. The benchmarks show the expected quadratic increase in running time as a function of both number of particles and number of scales. However, the linear complexity of the computation of the Gaussian kernels will likely make the algorithm scale close to linearly in the number of scales with the GPU implementation in practical applications.

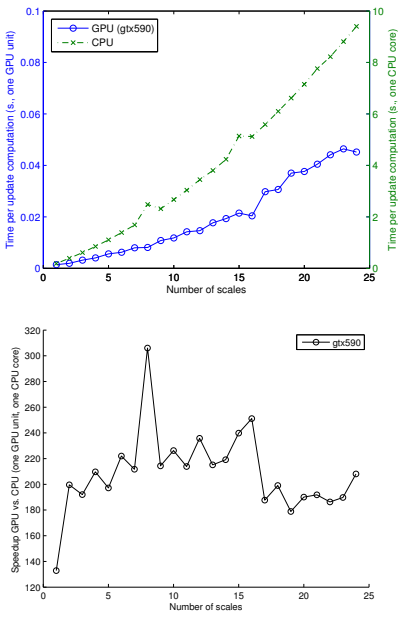


Figure 4. Running time (seconds) and speedup as a function of scale for 1000 particles (artificially generated data, computation on system 1). The increase is quadratic, though with the GPU (GTX590) running time increasing less for lower number of scales.

Acknowledgments

The author wishes to thank Tim Warburton for inspiring presentations on GPU architecture and hardware access.

References

[1] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache. A Log-Euclidean framework for statistics on diffeomorphisms. In *MICCAI 2006*, pages 924–931. 2006. 1

[2] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *IJCV*, 61(2):139–157, 2005. 1

[3] M. Bruveris, F. Gay-Balmaz, D. D. Holm, and T. S. Ratiu. The momentum map representation of images. *0912.2990*, Dec. 2009. 1

[4] R. Castillo, E. Castillo, R. Guerra, V. E. Johnson, T. McPhail, A. K. Garg, and T. Guerrero. A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. *Physics in Medicine and Biology*, 54(7):1849–1870, Apr. 2009. 5

[5] G. Christensen, R. Rabbitt, and M. Miller. Deformable templates using large deformation kinematics. *Image Processing, IEEE Transactions on*, 5(10), 2002. 1

[6] P. Dupuis, U. Grenander, and M. I. Miller. Variational problems on flows of diffeomorphisms for image matching. 1998. 1

[7] U. Grenander. *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford University Press, USA, Feb. 1994. 1

[8] L. K. Ha. *High Performance Multi-scale image processing framework on multiGPUs with applications to unbiased diffeomorphic atlas construction*. PhD thesis, University of Utah, 2011. 1

[9] L. K. Ha, J. Krüger, P. T. Fletcher, S. C. Joshi, and C. T. Silva. Fast parallel unbiased diffeomorphic atlas construction on Multi-Graphics processing units. In *Eurographics Symposium on Parallel Graphics and Visualization, EGPGV 2009*, Munich, Germany, 2009. 1

[10] X. Pennec, R. Stefanescu, V. Arsigny, P. Fillard, and N. Ayache. Riemannian elasticity: A statistical regularization framework for non-linear registration. In *MICCAI 2005*, pages 943–950. 2005. 1

[11] S. Sommer, F. Lauze, M. Nielsen, and X. Pennec. Kernel bundle EPDiff: evolution equations for Multi-Scale diffeomorphic image registration. In *SSVM 2011*. Springer, 2011. 1, 2

[12] S. Sommer, M. Nielsen, F. Lauze, and X. Pennec. A Multi-Scale kernel bundle for LDDMM: towards sparse deformation description across space and scales. In *IPMI 2011*. Springer, 2011. 1, 2, 3, 5

[13] J. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, Sept. 1998. 1

[14] A. Trouvé. An infinite dimensional group approach for physics based models in patterns recognition, 1995. 1

[15] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic demons: efficient non-parametric image registration. *NeuroImage*, 45(1 Suppl):61–72, 2009. 1

[16] L. Younes. *Shapes and Diffeomorphisms*. Springer, 2010. 1, 2, 3

5.

Paper #4:
The Differential of the Exponential Map, Jacobi Fields and Exact Principal Geodesic Analysis

Paper submitted to Foundations of Computational Mathematics, May 2011.

Authors:
Stefan Sommer, François Lauze, and Mads Nielsen

Notes:

Performing statistics in non-linear spaces, in particular on manifolds, requires computational tools to compute directions, distances, and projections. Inspired from the embedded manifold constructed in Paper #6, we develop methods for computing Jacobi fields and the differential of the Exponential map on Riemannian manifolds. Various applications of this is dicussed including estimating sectional curvature. We show how the the algorithms can be used for computing Prinpcal Geodesic Analysis (PGA) without the commonly used tangent space linearization. In the experimental section, we test how curvature affects the results of the *exact PGA* algorithm.

Noname manuscript No.
(will be inserted by the editor)

The Differential of the Exponential Map, Jacobi Fields and Exact Principal Geodesic Analysis

S. Sommer · F. Lauze · M. Nielsen

the date of receipt and acceptance should be inserted later

Abstract The importance of manifolds and Riemannian geometry is spreading to applied fields in which the need to model non-linear structure has spurred wide-spread interest in geometry. The transfer of interest has created demand for methods for computing classical constructs of geometry on manifolds occurring in practical applications. This paper develops initial value problems for the computation of the differential of the exponential map and Jacobi fields on parametrically and implicitly represented manifolds. It is shown how the solution to these problems allow for determining sectional curvatures and provides upper bounds for injectivity radii. In addition, when combined with the second derivative of the exponential map, the initial value problems allow for numerical computation of Principal Geodesic Analysis, a non-linear version of the Principal Component Analysis procedure for estimating variability in datasets. The paper develops algorithms for computing Principal Geodesic Analysis without the tangent space approximation previously used and, thereby, provides an example of how the constructs of theoretical geometry apply to solving problems in statistics. By testing the algorithms on synthetic datasets, we show how curvature affects the result of PGA.

Keywords manifolds, Riemannian metrics, manifold valued statistics, principal component analysis, principal geodesic analysis, geodesic PCA

Mathematics Subject Classification (2010) 65K10 · 57R99

The work is part of the LImB project (Learning Imaging Biomarkers), a joint initiative of the Image Group, Department of Computer Science, University of Copenhagen and Nordic Bioscience Imaging A/S, Herlev, Denmark.

S. Sommer (✉)
Dept. of Computer Science, Univ. of Copenhagen, Copenhagen, Denmark
E-mail: sommer@diku.dk, Tel.: +4535321400

F. Lauze
Dept. of Computer Science, Univ. of Copenhagen, Cph., Denmark, E-mail: lauze@diku.dk

M. Nielsen
Dept. of Computer Science, Univ. of Copenhagen, Copenhagen, Denmark
Synarc Imaging Technologies, Rødovre, Denmark, E-mail: madsn@diku.dk

1 Introduction

Manifolds, sets locally modeled by Euclidean spaces, have a long and intriguing history in mathematics, and topological, differential geometric, and Riemannian geometric properties of manifolds have been studied extensively with results extending far beyond the fields of manifolds themselves. The introduction of high-performance computing in applied fields has widened the use of manifolds, and Riemannian manifolds, in particular, are now used for modeling a range of problems possessing non-linear structure. Applications include shape modeling (complex projective shape spaces [23] and medial representations of surfaces [1,20]), imaging (tensor manifolds in diffusion tensor imaging [9,10,31] and image segmentation and registration [4,32]), and several other fields (forestry [18], human motion modeling [37,27,40]).

To fully utilize the power of manifolds in modeling, it is essential to develop fast and robust algorithms for computing various manifold constructions. Computing intrinsic distances, Jacobi fields, curvatures, and injectivity radii poses important problems [18] as well as solving optimization problems posed on manifolds or in manifold tangent spaces and defining and computing manifold generalizations of common Euclidean space statistics. The papers [6,22,29,24,36,39] address first-order manifold problems, and certain second-order problems have been considered but mainly on limited classes of manifolds [8]. Generalizing linear statistics has been the focus of the papers [21,30,11,13,18].

In this article, we study the second-order problems arising from variations of the initial velocity of geodesics. This will allow us to compute structures fundamental to geometry and to numerically solve certain optimization problems posed in tangent spaces of manifolds. The developed methods apply to manifolds represented both parametrically and implicitly without preconditions such as knowledge of explicit formulas for geodesics. Hence, in addition to being interesting from a geometrical and computational point of view, the algorithms will be useful for applications in several of the mentioned areas.

To exemplify this, we consider the problem of capturing the variation of a set of manifold valued data. The well-known Principal Component Analysis procedure (PCA) has been generalized to manifold valued data with the introduction of Principal Geodesic Analysis (PGA, [13]). The construction is the source of continuing interest from both application oriented authors and the statistical community, most recently with the development of Geodesic PCA (GPCA, [18]). Both PGA and GPCA have been used successfully for a number of applications [13,9,18,41, 35,39].

Until now, there were no algorithm for numerically computing PGA for general manifolds. Linear approximations have been used instead except for special classes of manifolds where geodesics have explicit analytical formulas [35,18]. Because PGA is posed as an optimization problem in the tangent space of the manifold, the tools developed here apply to computing it without linearizing the manifold. We will show how those tools allow us to compute exact PGA for a wide range of manifolds under some assumptions on the optimization problems.

1.1 Related Work

A vast body of mathematical literature describes manifolds and Riemannian structures, and [7,26] provide excellent introductions to the field. Different aspects of numerical computation on implicitly defined manifolds are covered in [44,34,33]. Generalized inverses are important in the study of implicitly defined manifolds, and we will use a result of Decell [5].

An important starting point for our work is the paper of Dedieu and Nowicki [6] where the authors develop an initial value problem (IVP) for the computation of geodesics on implicitly defined manifolds. This result, together with the IVP defining geodesics in the parametrized case [7], constitutes the basis for the IVPs developed in the following sections. A similar approach is taken in [43] for computing Jacobi fields on the infinite dimensional manifold of diffeomorphisms. Several authors have studied the solution of the exponential map inverse problem, often called the logarithm map: in [29,24,36], different schemes are used to evolve an initial path towards a geodesic, and [22,25,39] use shooting methods. We build upon these works by assuming the logarithm problem is solved for the manifolds in question.

An optimization problem can be posed on a manifold in the sense that the domain of the cost function is restricted to the manifold and the sought for optima must reside on the manifold. Such problems are extensively covered in the literature (e.g. [28,42]). The optimization problems we will solve involves the manifold geometry in the cost functions, but the domains will be the linear tangent spaces or subsets thereof with simple geometry. Therefore, the complexity will lie in the cost functions and not the optimization domains, and we will not need to use the optimization algorithms dealing with manifold domains.

The manifold generalization of linear PCA, PGA, was first introduced in [12], but it was formulated in the form most widely used in [13]. It has subsequently been used for several applications. To mention a few, the authors in [13,9] study variations of medial atoms, [41] uses a variation of PGA for facial classification, [35] presents examples on motion capture data, and [39] applies PGA to vertebrae outlines. In addition, finding principal modes in tangent spaces, the procedure labeled linearized PGA in this paper, has been used for analyzing spine deformation modes and deformities in [2,3]. The algorithm presented in [13] for computing PGA with tangent space linearization is most widely used. In contrast to this, [35] computes PGA as defined in [12] without approximations, but only for a specific manifold, the Lie group SO(3). Our recent paper [38] uses the methods presented here to experimentally assess the effect of tangent space linearization, and we show that the algorithms work on high dimensional manifolds modelling real-life data.

A recent wave of interest in manifold valued statistics from the statistical community has lead to the development of Geodesic PCA (GPCA, [18,19,17]). GPCA is in many respects close to PGA but optimizes for the placement of the center point and minimizes projection residuals along geodesics instead of maximizing variance in geodesic subspaces. GPCA uses no linear approximation, but it is currently only computed on spaces where explicit formulas for geodesics exist and on quotients of such spaces.

1.2 Content and Outline

The paper will present the following main contributions:

(1) We construct initial value problems allowing the computation of the differential of the exponential map and Jacobi fields, and second derivative of the exponential map on both parametric and implicitly represented manifolds of finite dimension.
(2) We show how the tools developed allow for numerical computation of the sectional curvature and injectivity radius bounds for the manifolds.
(3) We present an algorithm allowing the computation of PGA without linearizing the problem to the tangent space.
(4) We present examples showing the differences between exact PGA and the linearized PGA previously used, and how the differences depend on the curvature of the manifold.

Due to the generality of the setup, the algorithm in (3) will work for many of the applications using PGA as defined in [13]. In particular, it will apply to those of the above mentioned examples using finite dimensional manifolds with available parametrization or implicit representation. We comment more on the classes of manifolds covered in section 2.1. In addition, we will need some assumptions on the manifold and dataset ensuring the optimization problems are well-behaved so that true global optima are found.

The importance of curvature computations is noted in [18], which lists the ability to compute sectional curvature as a high importance open problem. The result of (2) can be seen as a partial solution to this problem; we are indeed able to numerically compute the sectional curvature, although for the anomalous shape-spaces [23] used in [18] no parametrization or implicit representation is directly available, and hence the methods presented here do not apply.

In the experiments (4), we evaluate how the difference between the methods vary as we increase the curvature of the manifold. This experiment, which to the best of our knowledge has not been made before, is made possible by the generality of the algorithms of (1), which frees us from previous restrictions to specific manifolds such as $SO(3)$ [35] or anomalous shape-spaces [18].

The paper will start by a brief discussion of the required notation and geometry in section 2. We will touch upon the definition of PGA and how curvature and injectivity radius bounds relate to Jacobi fields. The reader already familiar with Riemannian geometry may wish to skip parts of this section. In section 3, we present IVPs for the differential of the exponential map and Jacobi fields and for the second derivative of the exponential map. The actual derivations are lengthy and are, therefore, covered in the appendices. Following this, in section 4, we develop the exact PGA algorithm. We end the paper with experiments in section 5 and concluding remarks.

2 Geometry and Notation

We give a brief discussion of some aspects of differential and Riemannian geometry and, at the same time, introduce the notation used in the rest of the paper. The reader is referred to [7] for an introduction to differential geometry and Riemannian manifolds.

2.1 Manifolds and Their Representations

We will in the paper work with differentiable manifolds of finite dimension, and, in the sequel, M will denote such a manifold of dimension η. We will need M to be sufficiently smooth, i.e. of class C^k for $k = 3$ or 4 depending on the application. A chart of M is then a map $\varphi \in C^k(U, M)$ from an open subset U of $\mathbb{R}^\eta$ to the manifold, and, since a chart provides a coordinate representation of a part of the manifold, it is often called a local parametrization.

Manifolds can be represented without local parameterizations. Let M be a level set of a differentiable map $F : \mathbb{R}^m \to \mathbb{R}^n$. If the Jacobian matrix $D_x F$ has full rank n for all $x \in M$, the level set is said to be regular. In that case, M will be an $(m-n)$-dimensional manifold, and we say that M is *implicitly* defined. The space $\mathbb{R}^m$ is called the embedding space. Throughout this paper, when dealing with implicitly defined manifolds, m and n will denote the dimension of the domain and codomain of F, respectively. We then have $\eta = m - n$ for the dimension η of the manifold,

In addition to local parametrizations and implicit representations, other ways of representing manifolds include discrete triangulations used for surfaces and quotients $\tilde{M}/G$ of a larger manifold $\tilde{M}$ by a group G. The latter is for example the case for Kendall's shape-spaces Σ_d^k [23]. Kendall's shape-spaces for planar points are actually complex projective spaces $\mathbb{C}P^{k-2}$ for which parameterizations are available, and, for points in 3-dimensional space and higher, the shape-spaces are anomalous and not manifolds. The spaces studied in [18] belong to this class.

Our methods do not apply directly to cases where local parametrizations or implicit representations are not available. We note, however, that for the quotients used in [18], $\tilde{M}$ is a high-dimensional sphere and much of the optimization is performed on $\tilde{M}$ instead of M/G. We are currently investigating how our methods can complement this in extending the approach to quotients $\tilde{M}/G$ with $\tilde{M}$ not restricted to being a sphere.

2.2 Curves and Differentiation

We will deal with parametrized entities, most notably curves on manifolds, and we use subscripts for the parameter. For example, a curve on M dependent on t will be denoted x_t. As our curves will normally start at $t = 0$, the starting point of curve x_t will be the point x_0. The subscript notation should not be confused with differentiation with respect to the parameter t. When a local parametrization is available, we will often use it to represent the curve, and we will normally not distinguish between the curve and its expression $x_t = (x_t^1, \ldots, x_t^\eta)$ in parameter space.

The tangent space of M at a point p is, a vector space of dimension η, will be denoted $T_p M$, and the derivative $\frac{d}{dt}x_t$ of a curve x_t evaluated at $\tilde{t}$ then belongs to $T_{x_{\tilde{t}}}M$. We will often write just $\frac{d}{dt}x_{\tilde{t}}$ for such vectors, i.e. $\frac{d}{dt}x_t|_{t=\tilde{t}}$. In addition, when differentiating curves with respect to t, we often use the shorthand $\dot{x}_t$. With these conventions, $\frac{d}{dt}x_t|_{t=0}$, the initial velocity of the curve x_t, will be written $\dot{x}_0$.

The differential of a map $f : M \to N$ will be denoted df and its evaluation at $p \in M$ will be denoted $d_p f$. When bases for $T_p M$ and $T_{f(p)}M$ are specified, or when M and N are Euclidean spaces, we will write Df instead of df. We will encounter maps defined on a product of manifolds, e.g. $(v, w) \mapsto g(v, w) : M \times \tilde{M} \to N$, for

which we will need to distinguish differentiation with respect to one of the variables only. Letting one of the parameters have a fixed value w_0, the differential of the restricted function $v \mapsto g(v, w_0)$ from M to N evaluated at v_0 is denoted $d^v_{(v_0,w_0)}g$. Along the same lines, if V is a submanifold of M, the differential of $f|_V : V \to N$ will be denoted $d^{v \in V} f$ and its evaluation at $v_0 \in V$ will be written $d^{v \in V}_{v_0} f$.

2.3 Riemannian Manifolds and Geodesics

We will work solely with Riemannian manifolds, i.e. differentiable manifolds endowed with a smooth family of inner products on their tangent spaces. More precisely, a Riemannian metric on a manifold M is a smooth map g which associates to $p \in M$ an inner product $\langle \cdot, \cdot \rangle_p$ on T_pM, and, in a local parametrization, g will be a smooth map to the space of symmetric, positive definite matrices of order η. The pair (M, g) is then a Riemannian manifold. When M is a submanifold of $\mathbb{R}^m$, the tangent space T_pM of M at a point p can be identified with a linear subspace of $\mathbb{R}^m$ of dimension η, and the inner product $\langle \cdot, \cdot \rangle_p$ will be chosen to be the restriction of the standard inner product of $\mathbb{R}^m$.

The Riemannian metric determines notions such as length of curves, differentation of vector fields, Christoffel symbols, and geodesics. If x_t is a curve, the length $l(x_t)$ is given by the integral $\int \|\dot{x}_t\| dt$ using the norm $\|\cdot\|$ on $T_{x_t}M$ induced by the metric. Computing directional derivatives of a vector field is done by a connection that associates to a pair (X, Y) of vector fields on M a new vector field denoted $\nabla_Y X$ so that $(\nabla_Y X)(p)$ will be a directional derivative of X at p in the direction $Y(p)$. A special connection, called the Levi-Cività connection, is associated to the Riemannian metric, and the connection defines the covariant derivative $\frac{D}{dt}V_t$ of a vector field V_t along a curve. On implicitly defined manifolds, $\frac{D}{dt}V_t$ is simply the projection of the usual derivative of vector fields onto $T_{x_t}M$, and, in a local parametrization, the covariant derivative of the vector fields $(\partial_{x_1}, \ldots, \partial_{x_\eta})$ defines the Christoffel symbols Γ^k_{ij} of the metric by the relations $\nabla_{\partial_{x_i}} \partial_{x_j} = \sum_{k=1}^{\eta} \Gamma^k_{ij} \partial_{x_k}$. The η^3 functions $\Gamma^k_{ij}(x)$ satisfy the symmetry relation $\Gamma^k_{ij} = \Gamma^k_{ji}$.

Geodesic curves, manifold generalizations of straight lines, are characterized by having vanishing intrinsic acceleration expressed by the covariant derivative of the velocity field, $\frac{D}{dt}\dot{x}_t$ being zero. Geodesics are locally length minimizing and unique in the sense that given a point q and a velocity $v \in T_pM$, the geodesic passing q with velocity v is unique. The map which constructs geodesics given q and v is called the exponential map and denoted Exp. Thus, the unique geodesic is the curve $x_t = \mathrm{Exp}_p tv$.

For points $\tilde{q}$ in a sufficiently small neighborhood of q, the length minimizing curve joining q and $\tilde{q}$ is unique as well. Given q and $\tilde{q}$, the initial direction in which to travel geodesically from q in order to reach $\tilde{q}$ is given by the result of the logarithm map $\mathrm{Log}_q(\tilde{q})$. We get the corresponding geodesic as the curve $t \mapsto \mathrm{Exp}_q(t\mathrm{Log}_q\tilde{q})$, and hence Log_q is the inverse of Exp_q. Subsets $\mathrm{Exp}_q B_r(0)$ of M with $B_r(0)$ being a ball in T_qM and with the radius $r > 0$ sufficiently small are examples of neighborhoods of q in which $\mathrm{Log}_q(\tilde{q})$ is defined. Whenever we use the Log-map, we will restrict to such neighborhoods without explicitly mentioning it.

The gradient $\mathrm{grad}\, h$ of a real valued function $h : M \to \mathbb{R}$ is also defined using the metric: at $p \in M$, $\mathrm{grad}_p h$ is the unique vector in T_pM which represents $d_p h$ in

the sense that $d_p h(v) = \langle \operatorname{grad}_p h, v \rangle$ for all $v \in T_pM$. Whenever a basis of T_pM is specified, or when M is Euclidean, we switch to the usual notation ∇h. Similarly, the Hessian of h is defined by the relation $\operatorname{Hessian}(h)X = \nabla_X \operatorname{grad} h$ for all vector fields X. Again, when a basis of T_pM is specified, or when M is Euclidean, we use the usual notation $H(h)$.

2.4 Geodesic Systems

When a manifold is represented by a parametrization, the value of exponential map can be found as the solution of the IVP

$$\begin{aligned} \ddot{x}_t^k &= -\sum_{i,j}^{\eta} \Gamma_{ij}^k(x_t) \dot{x_t}^i \dot{x_t}^j, \ k = 1, \ldots, \eta \\ x_0 &= q, \ \dot{x}_0 = v \end{aligned} \tag{1}$$

in parameter space at time $t = 1$. Recall that η denotes the dimension of the manifold and that a chart $\varphi : \mathbb{R}^\eta \to M$ is used to connect the parameter space and the manifold. This classical characterization of geodesics is not directly usable when the manifold is represented implicitly and, therefore, neither parametrization nor Christoffel symbols are directly available. To handle this situation, a first order IVP for the computation of the exponential map on implicitly represented manifolds as developed in [6]. Here $\operatorname{Exp}_q v$ can be found as the x-part of the solution of the following IVP at time $t = 1$:

$$\begin{aligned} \dot{p}_t &= -\left(\sum_{k=1}^{n} \mu^k(x_t, p_t) H_{x_t}(F^k) \right) \dot{x}_t \ , \\ \dot{x}_t &= \left(I - D_{x_t} F^\dagger D_{x_t} F \right) p_t \ , \\ x_0 &= q, \ p_0 = v \ . \end{aligned} \tag{2}$$

The map $\mu : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^n$ is defined by $(x, p) \mapsto -(D_x F^T)^\dagger p$, and the symbol $A^\dagger$ denotes the generalized inverse of the possibly non-square or singular matrix A [5].

2.5 Jacobi Fields and Global Geometry

Studying variations of geodesics leads to the notion of Jacobi fields, which encode important geometric information such as curvature and injectivity radius. In order to define Jacobi fields, let $x_{t,s}$ be a family of geodesics parametrized by s, i.e. for each $\tilde{s}$, the curve $t \mapsto x_{t,\tilde{s}}$ is a geodesic. When fixing the position t on the curves but varying the parameter s, we obtain the vector field $\frac{d}{ds} x_{t,0}$, and such a vector field is called a Jacobi field along the geodesic $x_{t,0}$.[1] The Jacobi fields along a given geodesic are uniquely determined by the initial conditions J_0 and $\frac{D}{dt} J_0$, the variation of the initial points $x_{0,s}$ and the covariant derivative of the field at $t = 0$, respectively. Define $q_s = x_{0,s}$, $v_s = \dot{x}_{0,s}$, and $w = \frac{d}{ds} v_0$. If $\frac{d}{ds} q_0 = J_0$ and $w = \frac{D}{dt} J_0$ then $\frac{d}{ds} \operatorname{Exp}_{q_0}(t v_0)$ is equal to J_t [7, Chap. 5]. Therefore, in cases when q_s is constant and J_0 therefore 0, we have the following connection between J_t and $d\operatorname{Exp}$:

[1] Recall that with the notation introduced in section 2.2, $\frac{d}{ds} x_{t,0}$ equals $\frac{d}{ds} x_{t,s}|_{s=0}$.

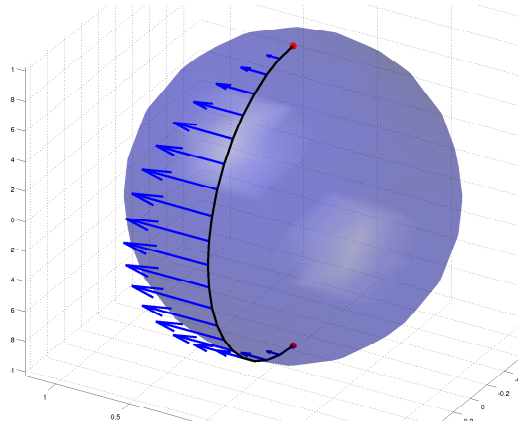


Fig. 1 The sphere $\mathbb{S}^2$ with a Jacobi field along a geodesic connecting the poles. Each pole is a conjugate point to the other since the non-zero Jacobi field vanishes. The injectivity radius is equal to the length of the geodesic, π.

$$d_{v_0}\mathrm{Exp}_{q_0} tw = J_t \ . \tag{3}$$

Jacobi fields can equivalently be defined as solutions to the ODE

$$\frac{D^2}{dt^2} J_t = -R(\dot{x}_t, J_t)\dot{x}_t \tag{4}$$

with R denoting the curvature endomorphism [7, Chap. 5]. For parametrized manifolds, the ODE can be written in parameter space and can, in principle, be used for numerical computations of Jacobi fields. The expressions are somewhat complicated, though, and we will obtain a different IVP by differentiating the system (1). The curvature endomorphism is not easily computed when the manifold is represented implicitly, and, therefore, the above ODE is not directly useful in this case. By differentiating the system (2), we remedy this in the next section.

Besides allowing us to calculate $d_{v_0}\mathrm{Exp}_{q_0}$, Jacobi fields enable us to retrieve various geometric information about the manifold. We can for example estimate the sectional curvature of the manifold at q_0 using a Jacobi field J_t as defined above with $J_0 = 0$ and v_0, w orthonormal. Performing a Taylor expansion of the length $\|J_t\|$, we get

$$\|J_t\| = t - \frac{1}{6} K_{q_0}(\sigma) t^3 + O(t^4)$$

where $K_{q_0}(\sigma)$ is the sectional curvature of the plane span $\{v_0, w\}$ in $T_{q_0}M$ [7, Chap. 5]. For small t, the sectional curvature can then be estimated by

$$K_{q_0}(\sigma) \approx \frac{6}{t^3}(t - \|J(t)\|) \ . \tag{5}$$

Furthermore, if J_t is a non-zero Jacobi field with $J_0 = 0$ along a geodesic x_t and, for some $\tilde{t} > 0$, also $J_{\tilde{t}} = 0$ then $x_{\tilde{t}}$ is called a conjugate point to x_0. This implies that for any $r > \tilde{t}$, the geodesic x_t is not the shortest joining x_0 and x_r [7, Chap. 13]. In this way, we get an upper bound on the injectivity radius of M, which, in general terms, specifies the minimum length of non-minimizing geodesics. Figure 1 illustrates the situation on the sphere $\mathbb{S}^2$.

2.6 Geodesic Subspaces

Linear subspaces are of great importance when studying data in Euclidean spaces; PCA, for example, can be formulated as an optimization problem on the set of linear subspaces. There is no obvious generalization of linear subspaces to manifolds, but, if one accepts the choice of a center point, the notion of geodesic subspaces becomes useful. A subset $\mathrm{Exp}_q V$ of M is called a geodesic subspace centered at q if V is a linear subspace of T_qM. Geodesics between q and any point in the subspace are contained in the subspace, a fact which, in general, is not true for geodesics between arbitrary pairs of points in the subspace. The projection of a point $x \in M$ onto a geodesic subspace $S = \mathrm{Exp}_q V$ is defined as

$$\begin{aligned} \pi_S(x) &= \mathrm{argmin}_{y\in S} d(x,y)^2 = \mathrm{argmin}_{y\in S} \|\mathrm{Log}_y x\|^2 \\ &= \mathrm{Exp}_q(\mathrm{argmin}_{w\in V} \|\mathrm{Log}_{\mathrm{Exp}_q w} x\|^2) \ . \end{aligned} \tag{6}$$

Neither existence or uniqueness of the projection is in general ensured, although, for each geodesic subspace S, the set of points for which uniqueness fail has zero measure in M [18]. Existence of the projection is ensured if S is compact, which, for example, is the case if M is compact and S an embedded submanifold.

2.7 Principal Geodesic Analysis

Principal Component Analysis (PCA) is widely used to model the variability of data in Euclidean spaces. The procedure provides linear dimensionality reduction by defining a sequence of linear subspaces maximizing the variance of the projection of the data to the subspaces or, equivalently, minimizing the reconstruction errors. The kth subspace is spanned by an orthogonal basis $\{v^1, \ldots, v^k\}$ of principal components $v^1, \ldots, v^k$, and the ith principal component is defined recursively by

$$v^i = \mathrm{argmax}_{\|v\|=1} \frac{1}{N} \sum_{j=1}^{N} \left(\langle x_j, v\rangle^2 + \sum_{l=1}^{i-1} \left\langle x_j, v^l \right\rangle^2 \right) \tag{7}$$

when formulated as to maximize the variance of the projection of the dataset $\{x_1, \ldots, x_N\}$ to the subspaces $\mathrm{span}\,\{v^1, \ldots, v^{i-1}\}$.

PCA is dependent on the vector space structure of the Euclidean space and hence cannot be performed on manifold valued datasets. Principal Geodesic Analysis was developed to overcome this limitation. PGA finds geodesic subspaces centered a point $\mu \in M$ with μ usually being an intrinsic mean[2] of the dataset $\{x_1, \ldots, x_N\}$, $x_j \in M$. The kth geodesic subspace S_k of $T_\mu M$ is defined as $\mathrm{Exp}_\mu(V_k)$ with $V_k = \mathrm{span}\,\{v^1, \ldots, v^k\}$ being the span of the principal directions $v^1, \ldots, v^k$ defined recursively by

$$\begin{aligned} v^i &= \mathrm{argmax}_{\|v\|=1, v\in V_{i-1}^{\perp}} \frac{1}{N} \sum_{j=1}^{N} d(\mu, \pi_{S_v}(x_j))^2 \ , \\ S_v &= \mathrm{Exp}_\mu(\mathrm{span}\,\{V_{i-1}, v\}) \ . \end{aligned} \tag{8$*$}$$

[2] The notion of intrinsic mean goes back to Fréchet [14] and Karcher [21]. As in [13], we define it as $\mathrm{argmin}_{\mu\in M} \sum_{j=1}^{N} d(\mu, x_j)^2$. Uniqueness issues are treated in [21].

The notation $V_{i-1}^{\perp}$ denotes the orthogonal complement of V_{i-1} in $T_\mu M$. The term being maximized is the sample variance of the projected data, the expected value of the squared distance to μ, and PGA therefore extends PCA by finding *geodesic subspaces* in which variance is maximized.

Since a method for computing the projection $\pi_{S_k}(x)$ has not been available for general manifolds, PGA has traditionally been computed using the orthogonal projection in the tangent space of μ to approximate the true projection. With this approximation, equation (8∗) simplifies to

$$v^i \approx \operatorname{argmax}_{\|v\|=1} \frac{1}{N} \sum_{j=1}^{N} \left(\left\langle \operatorname{Log}_\mu x_j, v \right\rangle^2 + \sum_{l=1}^{i-1} \left\langle \operatorname{Log}_\mu x_j, v^l \right\rangle^2 \right)$$

which is equivalent to (7), and, therefore, the procedure amounts to performing regular PCA on the vectors $\operatorname{Log}_\mu x_j$. We will refer to PGA with the approximation as *linearized* PGA, and PGA as defined by (8∗) will be referred to as *exact* PGA.

The above and prevalent definition of PGA is developed in [13], but a slightly different definition was introduced in [12]. The latter definition involves only one-dimensional subspaces and uses Lie group structure. In [35], the fact that π_S has a closed form solution on the sphere $\mathbb{S}^3$ when S is a one-dimensional geodesic subspace is used to compute exact PGA with the [12] definition by performing a steepest descent using the gradient of the cost function equivalent to the cost function of (8∗).

Replacing maximization of sample variance by minimization of reconstruction error, we obtain another manifold extension of PCA and thus an alternate definition of PGA:

$$v^i = \operatorname{argmin}_{\|v\|=1, v \in V_{i-1}^{\perp}} \frac{1}{N} \sum_{j=1}^{N} d(x_j, \pi_{S_v}(x_j))^2 \ . \tag{8∗∗}$$

In contrast to vector space PCA, the two PGA definitions are *not* equivalent, a fact showing that the Euclidean and curved situations differ fundamentally. The latter formulation is chosen for Geodesic PCA to avoid instabilities of variance maximization [18], but the optimization algorithms developed in this paper work for both formulations. We will use the variance formulation for the experiments, but we will collectively refer to definitions by (8).

In general, PGA might not be well-defined as the mean might not be unique and both existence and uniqueness may fail for the projections (6) and the optimization problems (8). The convexity bounds of Karcher [21] ensures uniquesness of the mean for sufficiently local data, but setting up sufficient conditions to ensure well-posedness of (6) and (8) is a difficult issue, and here we will just assume well-posedness for the given manifold and dataset.

3 The Differentials

In this section, we aim at developing an initial value problem (IVPs) describing the differential of the exponential map and Jacobi fields, and, in addition, we will differentiate the IVPs a second time and thereby create the tools needed for the

PGA algorithms presented in the next section. The basic strategy is simple: we differentiate the systems of section 2.4 and use the resulting IVPs.

It is a well-known fact that IVPs satisfying natural properties are differentiable with respect to their initial values [16, Chap. I.14]. The important contribution of this section is the explicit expressions for the differentiated systems that allow numerical integration and, in particular for the case of implicitly represented manifolds, are not straightforward to derive. To the best of our knowledge, no IVP describing the differential of the exponential map and Jacobi fields has previously been available in the implicit case; the IVP (11) remedies this situation. As previously noted, the ODE (4) describes Jacobi fields in the parameterized case but the expressions in parameter space are complicated. Therefore, we derive the IVP (10) below, which we find simpler to work with for the applications of this paper.

The presence of the generalized inverse in system (2) proves to be the main source of complexity for the implicit case. We handle the differentiation of this system using the following result of Decell:

Theorem 1 ([5]) *Let A_s and its generalized inverse $A_s^\dagger$ be differentiable s-dependent matrices. Then $\frac{d}{ds}(A_s^\dagger) = \Lambda(A_s, \frac{d}{ds}A_s)$ where*

$$\begin{aligned}\Lambda(A,B) = &- A^\dagger B A^\dagger + \Big(B^T(A^\dagger)^T A^\dagger + A^\dagger (A^\dagger)^T B^T\Big) \\ &- A^\dagger A\Big(B^T(A^\dagger)^T A^\dagger + A^\dagger (A^\dagger)^T B^T\Big) A A^\dagger \ .\end{aligned} \tag{9}$$

We will apply the result with $A_s = D_{x_{t,s}}F$ with $x_{t,s}$ an s dependent family of geodesics and t fixed. To see that $D_{x_{t,s}}F^\dagger$ is differentiable with respect to s when $x_{t,s}$ depends smoothly on s, take a frame of the normal space to M in a neighborhood of $x_{t,s}$, and note that $D_{x_{t,s}}F^\dagger$ is a composition of a invertible map onto the frame depending smoothly on s and the frame itself.

The remaining computations for deriving the systems are lengthy and notationally heavy. At this point, we only state the results and postpone the derivations and the proof of the following theorem to Appendix A.

Theorem 2 *Let x_t be a geodesic in the C^3 manifold M with $x_0 = q$ and $\dot{x}_0 = v$, and let u, w be vectors in $T_{x_0}M$. Assume x_t is contained in a parametrized subset of M. Then the Jacobi field J_t along x_t with $J_0 = u$ and $\frac{D}{dt}J_0 = w$ can be found as the z-part of the solution of the IVP*

$$\begin{aligned}\begin{pmatrix}\dot{y}_t\\ \dot{z}_t\end{pmatrix} &= F^P_{q,v}\left(t, \begin{pmatrix}y_t\\ z_t\end{pmatrix}\right) \ , \\ \begin{pmatrix}y_0\\ z_0\end{pmatrix} &= \begin{pmatrix}w\\ u\end{pmatrix} \ ,\end{aligned} \tag{10}$$

with $F^P_{q,v}$ the map given in explicit form in Appendix A.

Now, let instead $M \subset \mathbb{R}^m$ be defined as a regular zero level set of a C^3 map $F : \mathbb{R}^m \to \mathbb{R}^n$. Then the Jacobi field J_t along x_t with $J_0 = u$ and $\frac{D}{dt}J_0 = w$ can be found as the z-part of the solution of the IVP

$$\begin{aligned}\begin{pmatrix}\dot{y}_t\\ \dot{z}_t\end{pmatrix} &= F^I_{q,v}\left(t, \begin{pmatrix}y_t\\ z_t\end{pmatrix}\right) \ , \\ \begin{pmatrix}y_0\\ z_0\end{pmatrix} &= \begin{pmatrix}w\\ u\end{pmatrix} \ ,\end{aligned} \tag{11}$$

with $F^I_{q,v}$ the map given in explicit form in Appendix A.

The maps $F^P_{q,v}$ and $F^I_{q,v}$ and consequently the systems (10) and (11) are linear in the initial values $(w\ u)^T$ as expected of systems describing differentials. They are non-autonomous due to the dependence on the position on the curve x_t.

The following corollary allows the computation of the derivative of the exponential map:

Corollary 1 *With the assumptions of Theorem 2, let (y_t, z_t) satisfy (10) or (11) with IVs $(w, 0)^T$. Then $d_v\mathrm{Exp}_q w$ is equal to z_1.*

Proof Let J_t be the Jacobi field along x_t with $J_0 = 0$ and $\frac{D}{dt}J_0 = w$. By Theorem 2, $z_1 = J_1$, which, by (3), is equal to $d_v\mathrm{Exp}_q w$.

The result enables us to compute the entire differential $d_v\mathrm{Exp}_q$ by applying the corollary to each element of a basis $\{w^1, \ldots, w^\eta\}$ for T_qM. The matrix having the results in its columns then equals $D_v\mathrm{Exp}_q$. Note that $\mathrm{Exp}_q\mathrm{Log}_q y = y$ implies that $d_y\mathrm{Log}_q = (d_{\mathrm{Log}_q y}\mathrm{Exp}_q)^{-1}$, a fact that allows the corollary to be used for computing $d_y\mathrm{Log}_q$ as well.

We can differentiate the systems (10) and (11) once more if the manifold is sufficiently smooth. The main difficulty here is performing the algebra of the already complicated expressions for $F^P_{q,v}$ and $F^I_{q,v}$. For the implicit case, we will need to find the second derivative of $D_{x_{t,s}}F^\dagger$ and hence extend Decell's result. For simplicity, we consider a family of geodesics $x_{t,s}$ with the start point $x_{0,s}$ constant in s. The derivations and the proof are again postponed to Appendix A.

Theorem 3 *Let $w \in T_qM$ with M of class C^4, and let $x_{t,s}$ be a family of geodesics with $x_{0,s} = q$ and $v_s = \dot{x}_{0,s}$. Define $u = \frac{d}{ds}v_0$, and let $V_{q,v_0,w,u} = \frac{d}{ds}\left(d_{v_s}\mathrm{Exp}_q w\right) = \frac{d}{ds}\left(\frac{d}{dr}\left(\mathrm{Exp}_q v_s + rw\right)\right)$. Assume $x_{t,s}$ is contained in a parametrized subset of M. Then $V_{q,v_0,w,u}$ can be found as the r-part of the solution of the IVP*

$$\begin{aligned}\begin{pmatrix}\dot{q}_t\\ \dot{r}_t\end{pmatrix} &= G^P_{q,v_0,w,u}\left(t, \begin{pmatrix}q_t\\ r_t\end{pmatrix}\right)\ ,\\ \begin{pmatrix}q_0\\ r_0\end{pmatrix} &= \begin{pmatrix}0\\ 0\end{pmatrix}\ ,\end{aligned} \tag{12}$$

with $G^P_{q,v_0,w,u}$ the map given in explicit form in Appendix A.

Now, let instead $M \subset \mathbb{R}^m$ be defined as a regular zero level set of a C^4 map $F : \mathbb{R}^m \to \mathbb{R}^n$. Then $V_{q,v_0,w,u}$ can be found as the r-part of the solution of the IVP

$$\begin{aligned}\begin{pmatrix}\dot{q}_t\\ \dot{r}_t\end{pmatrix} &= G^I_{q,v_0,w,u}\left(t, \begin{pmatrix}q_t\\ r_t\end{pmatrix}\right)\ ,\\ \begin{pmatrix}q_0\\ r_0\end{pmatrix} &= \begin{pmatrix}0\\ 0\end{pmatrix}\ ,\end{aligned} \tag{13}$$

with $G^I_{q,v_0,w,u}$ the map given in explicit form in Appendix A.

We note that solutions to (12) and (13) depend linearly on u even though the systems are not linear.

3.1 Numerical Considerations

The geodesic systems (1) and (2) can in both the parametrized and implicit case be expressed in Hamiltonian forms. In [6], the authors use this property along with symplectic numerical integrators to ensure the computed curves will be close to actual geodesics. This is possible since the Hamiltonian encodes the Riemannian metric. Derivatives of Hamiltonian systems can be expressed in Hamiltonian form, and, therefore, the systems of Theorem 2 and Theorem 3 have Hamiltonian formulations. Using symplectic integrators, we can preserve the Hamiltonians, but the usefulness of this is limited since the Hamiltonians do not have directly interpretable forms in contrast to the case of geodesic systems.

Along the same lines, we would like to use the preservation of quadratic forms for symplectic integrators [15] to preserve quadratic properties of the differential of the exponential map, e.g. the Gauss Lemma [7]. At this point, we have, however, not been able to establish this for the implicit case.

4 Exact PGA

We will provide algorithms for iteratively solving the optimization problems (8) and hence compute exact PGA as defined in [13] without the traditional linear approximation. The algorithms will work for parametrized and implicitly represented manifolds under the following assumptions. First, we require that the PGA problem is well-defined as discussed in section 2.7. Second, the logarithm map must be computable. As noted in the introduction, good implementations exist for both parametric and implicitly represented manifolds. Third, we will need to assume non-existence of local optima for the cost functions of (6) and (8) to ensure the optimization algorithms find the true global solutions. Forth, a local convexity assumption of the residual function, which is satisfied for local data, will be needed. We note that, if the third assumption is left out, it is indeed possible to find examples of manifolds and datasets where the algorithms will get stuck in local optima.

Solving the optimization problems (8) requires the ability to compute the projection operator π_S. We start by finding expressions for the gradients of the cost functions of the optimization problems using the IVPs derived in section 3, and, thereafter, we present the actual algorithms for solving the problems. The overall approach of solving (8) is similar to the approach of [35]. Our solution differs in that we are able to compute π_S and its differential without restricting to the manifold $SO(3)$ and in that we optimize (8) instead of the simpler[3] cost function of [12].

The optimization problems (6) and (8) are posed in the tangent space of the manifold at the sample mean and the unit sphere of that tangent space, respectively. These domains have relatively simple geometry, and, therefore, the complexity of the problems is contained in the cost functions. Because of this, we will not need algorithms for optimizing problems with domains of complicated geometry.

As we are able to compute the gradient of the cost function of the problems, we can use approaches such as steepest descent. Yet, because both prob-

[3] Simpler in the sense that projections in [12] involve only one-dimensional subspaces. The cost function of (8) uses i-dimensional subspaces for $i = 1, \ldots, \eta$.

lems are quadratic, optimization algorithms such as Gauss-Newton or Levenberg-Marquardt are also applicable if the Jacobians are present. For simplicity, we compute gradients and present steepest descent algorithms, but it is straightforward to compute Jacobians instead and use more advanced optimization algorithms.

4.1 The Projection

We consider the projection $\pi_S(x)$ of a point $x \in M$ on a geodesic subspace S. Assume S is centered at $\mu \in M$, let V be a k-dimensional subspace of $T_\mu M$ such that $S = \mathrm{Exp}_\mu V$, and define a residual function $R_{x,\mu} : V \to \mathbb{R}$ by $w \mapsto \|\mathrm{Log}_{\mathrm{Exp}_\mu w} x\|^2$ measuring distances between x and points in S. Computing $\pi_S(x)$ by solving (6) is then equivalent to finding $w \in V$ minimizing $R_{x,\mu}$. To find the gradient of $R_{x,\mu}$, choose an orthonormal basis for V and extend it to a basis for $T_\mu M$. Furthermore, let $w_0 \in V$ and choose an orthonormal basis for the tangent space $T_{\mathrm{Exp}_\mu w_0} M$. Karcher showed in [21] that the gradient $\mathrm{grad}^y \|\mathrm{Log}_y x\|^2$ equals $-2\mathrm{Log}_y x$, and, using this, we get the gradient of the residual function as

$$\nabla_{w_0}^{w \in V} R_{x,\mu} = -2(D_{w_0}\mathrm{Exp}_\mu)^T_{1,\ldots,k}(\mathrm{Log}_{\mathrm{Exp}_\mu w_0} x) \tag{14}$$

with $(D_{w_0}\mathrm{Exp}_\mu)_{1,\ldots,k}$ denoting the first k columns of $D_{w_0}\mathrm{Exp}_\mu$ when expressed using the chosen bases.

4.2 The Gradient of the Projection

In order to optimize (8), we will need to compute gradients of the form

$$\mathrm{grad}_{v_0}^{v \in V_{v_0}^\perp} d(y, \pi_{S_v}(x))^2 \tag{15}$$

with $V_{v_0} = \mathrm{span}\{v^1, \ldots, v^k, v_0\}$, $S_v = \mathrm{Exp}(V_{v_0})$, and $y \in M$ being either the intrinsic mean μ for (8$*$) or x for (8$**$).[4] This will involve the gradient of $\pi_{S_v}(x)$ with respect to v. To derive this, we extend the domain of residual function $R_{x,\mu}$ defined in the previous subsection from V to $T_\mu M$. We will choose bases for $T_\mu M$ and V_{v_0}, and we let $H(R_{x,\mu})$ denote the Hessian of $R_{x,\mu}$ and $H(R_{x,\mu}|_{V_{v_0}})$ denote the Hessian of $R_{x,\mu}$ restricted to V_{v_0} with respect to the bases. Using this notation, we get the following result:

Theorem 4 *Let $\{v^1, \ldots, v^k\}$ be a basis for a subspace $V \subset T_\mu M$. For each $v \in V^\perp$, let V_v be the subspace* $\mathrm{span}\{V, v\}$*, and let $S_v = \mathrm{Exp}_\mu V_v$ be the corresponding geodesic subspace. Fix $v_0 \in V^\perp$ and define $w_0 = \mathrm{Log}_\mu \pi_{S_{v_0}}(x)$ for an $x \in M$. Suppose the matrix $H_{v_0}(R_{x,\mu}|_{V_{v_0}})$ has full rank $k+1$. Extend the orthonormal basis $\{v^1, \ldots, v^k, v_0/\|v_0\|\}$ for V_{v_0} to an orthonormal basis for $T_\mu M$. Then*

$$\begin{aligned} D_{v_0}^{v \in V_{v_0}^\perp} \pi_{S_v}(x) = &-(D_{w_0}\mathrm{Exp}_\mu)\bar{v}_{x,\mu,v_0,S_{v_0}} \left(\nabla_{w_0}^{w \in V_{v_0}^\perp} R_{x,\mu}\right)^T \\ &+ w_0^{k+1}(D_{w_0}\mathrm{Exp}_\mu)E_{x,\mu,v_0,S_{v_0}} \ . \end{aligned} \tag{16}$$

[4] Since v in (8) is restricted to the unit sphere, we will not need the gradient in the direction of v_0, and, therefore, we find the gradient in the subspace $V_{v_0}^\perp$ instead of in the larger space $\mathrm{span}\{v^1, \ldots, v^k\}^\perp$.

The coordinates of the vector $\bar{v}_{x,\mu,v_0,S_{v_0}}$ *in the basis for* V_{v_0} *are contained in the* $(k+1)$*st column of the matrix* $H_{v_0}(R_{x,\mu}|_{V_{v_0}})^{-1}$*, the scalar* w_0^{k+1} *is the* $(k+1)$*st coordinate of* w_0 *in the basis, and* $E_{x,\mu,v_0,S_{v_0}}$ *is the matrix*

$$\begin{pmatrix} -H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)^{-1} B_{w_0,v_0} \\ I_{\eta-(k+1)} \end{pmatrix}$$

with B_{w_0,v_0} *the last* $\eta-(k+1)$ *columns of the matrix* $\left(H_{w_0}\left(R_{x,\mu}\right)\left(V\ v_0\right)\right)^T$ *and* $I_{\eta-(k+1)}$ *the identity matrix.*

The proof of the theorem is presented in Appendix B. The assumption that the Hessian of the restricted residual $R_{x,\mu}|_{V_{v_0}}$ must have full rank is equivalent to the residual $R_{x,\mu}$ having only non-degenerate critical points when restricted to V_{v_0}. It is shown in [21] that $R_{x,\mu}$ is convex at points sufficiently close to x and the assumption is therefore satisfied in such cases. In order to compute the right hand side of (16), it is necessary to compute parts of the Hessian of the non-restricted residual $R_{x,\mu}$. The expression for computing $H_{v_0}(R_{x,\mu})$ is given in Appendix B.

Because $d(y,\pi_{S_v}(x))^2 = \|\mathrm{Log}_y \pi_{S_v}(x)\|^2$, we have

$$\nabla_{v_0}^{v\in V_{v_0}^\perp} d(y,\pi_{S_v}(x))^2 = 2\left((D_{\pi_{S_{v_0}}(x)}\mathrm{Log}_y)(D_{v_0}^{v\in V_{v_0}^\perp}\pi_{S_v}(x))\right)^T (\mathrm{Log}_y \pi_{S_{v_0}}(x)) \ , \tag{17}$$

which, combined with (16), gives (15).

4.3 Exact PGA Algorithm

The expressions for the gradients of the cost functions enable us to iteratively solve the optimization problems (6) and (8) under the mentioned assumptions. We let μ be the intrinsic mean of a dataset $\{x_1,\ldots,x_N\}$ of points in M. The actual algorithms listed below are essentially steepest descent methods.

Algorithm 1 for computing $\pi_S(x)$ updates $w \in V$ instead of the actual point $y \in S$ that we are interested in. The vector w is related to y by $y = \mathrm{Exp}_\mu w$.

Algorithm 1 Calculate $\pi_S(x)$

Require: $x \in M$, $S = \mathrm{Exp}_\mu V$ geodesic subspace.
  $w \Leftarrow$ orthogonal projection of $\mathrm{Log}_\mu x$ onto V {initial guess}
  repeat
    $y \Leftarrow \mathrm{Exp}_\mu w$ {vector to point}
    $g \Leftarrow -2(D_{w_0}\mathrm{Exp}_\mu)^T_{1,\ldots,k}\mathrm{Log}_y x$ {gradient}
    $\tilde{w} \Leftarrow w$ {previous w}
    $w \Leftarrow w - g$ {update w}
  until $\|\tilde{w} - w\|$ is sufficiently small.

For solving (8), we use that

$$\nabla_{v_0}^{v\in V_{v_0}^\perp}\left(\frac{1}{N}\sum_{j=1}^N d(y,\pi_{S_v}(x_j))^2\right) = \frac{1}{N}\sum_{j=1}^N \nabla_{v_0}^{v\in V_{v_0}^\perp} d(y,\pi_{S_v}(x_j))^2 \ . \tag{18}$$

Since v in (8) is required to be on the unit sphere, the optimization will take place on a manifold, and a natural approach to compute iteration updates will use the exponential map. Yet, because of the symmetric geometry of the sphere, we approximate this using the simpler method of adding the gradient to the previous guess and normalizing. When computing the $(k+1)st$ principal direction, we choose the initial guess as the first regular PCA vector of the data projected to $V_k^\perp$ in $T_\mu M$. The algorithm for solving (8$*$) is listed in Algorithm 2, but by exchanging μ with x_j in the gradient computations and updating by subtracting the gradient, the algorithm will solve (8$**$) instead. See Figure 2 for an illustration of an iteration of the algorithm.

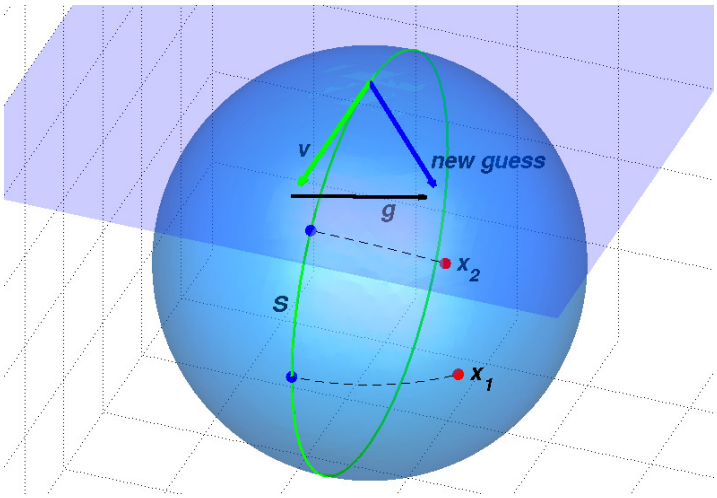


Fig. 2 An iteration of Algorithm 2. The figure shows data points x_1 and x_2 (red points) with projections (blue points) to the geodesic subspace S (green line). The vector v defining S is updated to the new guess by adding the gradient g.

Algorithm 2 Calculate the $(k+1)st$ principal direction of (8$*$).

Require: $\mu, x_1, \ldots, x_N \in M$, $\{v^1, \ldots, v^k\}$ orthogonal basis for $V_k \subset T_\mu M$.
  $v \Leftarrow$ first PCA vector of $\{x_j\}$ projected first to $T_\mu M$
    using Log_μ and then to $V_k^\perp$ {initial guess}
  repeat
    $g_j \Leftarrow \nabla_v^{v \in V_v^\perp} d(\mu, \pi_{S_v}(x_j))^2$ {for each j using (17)}
    $g \Leftarrow \frac{1}{N} \sum_{j=1}^N g_j$ {gradient using (18)}
    $\tilde{v} \Leftarrow v$ {previous v}
    $v \Leftarrow v + g$ {update v}
    $v \Leftarrow v/\|v\|$ {normalize}
  until $\|\tilde{v} - v\|$ is sufficiently small.

5 Experiments

We will perform experiments exemplifying the differences between exact PGA and linearized PGA with synthetic data projected onto low dimensional manifolds on which it is possible to visually identify the differences between the methods. We vary the curvature of the manifolds in order to show how curvature affects the differences, and we compare the curvature approximation (5) and injectivity radius bound with the true values. For a comparison between the methods on high dimensional manifolds modelling real-life data, we refer the reader to [38]. In that paper, we compute and compare exact and linearized PGA on a 50 dimensional manifold containing outlines of human vertebrae captured with lateral X-rays and on a 23 dimensional manifold containing human pose data acquired with tracking software.

The PGA algorithm is implemented in Matlab using Runge-Kutta ODE solvers. For the logarithm map, we use the shooting algorithm developed in [39]. All tolerances used for the integration and logarithm calculations are set at or lower than an order of magnitude of the precision used for the displayed results.

5.1 Synthetic Low-dimensional Data

We consider first surfaces embedded in $\mathbb{R}^3$ and defined by the equation

$$S_c = \{(x_1, x_2, x_3) | c x_1^2 + x_2^2 + x_3^2 = 1\}$$

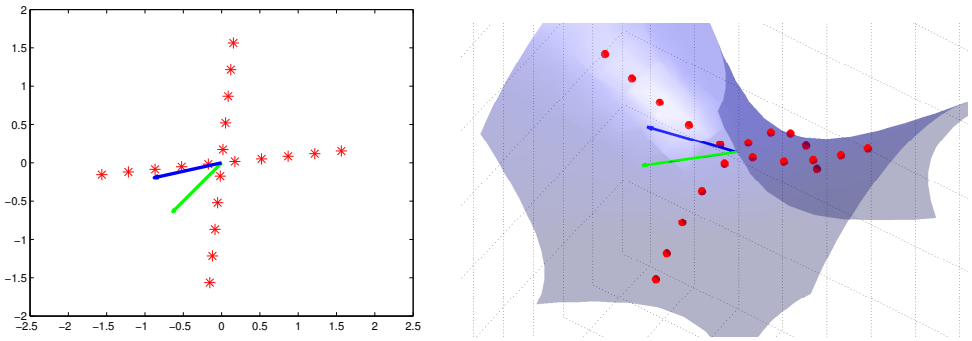
for different values of the scalar c. For $c > 0$, S_c is an ellipsoid and it is equal to $\mathbb{S}^2$ in the case $c = 1$. The surface S_0 is a cylinder and, for $c < 0$, S_c is hyperboloid. Consider the point $p = (0, 0, 1)$ and note that $p \in S_c$ for all c. The curvature of S_c at p is equal to c. Note in particular that for the cylinder case the curvature is zero; the cylinder locally has the geometry of the plane $\mathbb{R}^2$ even though it informally seems to curve.

We evenly distribute 20 points along two straight lines through the origin of the tangent space T_pS_c, project the points from T_pS_c to the surface S_c, and perform linearized and exact PGA using the variance formulation (8$*$). Figure 3 illustrates the situation in T_pS_{-1} and on S_{-1} embedded in $\mathbb{R}^3$, respectively.

Since linearized PCA amounts to Euclidean PCA in T_pS_c, the first principal direction found using linearized PGA divides the angle between the lines for all c. In contrast to this, the variance and the first principal direction found using exact PGA are dependent on c. Table 1 shows the angle between the principal directions found using the two methods, the variances and variance differences for different values of c.

c:	**1**	**0.5**	**0**	**-0.5**	**-1**	**-1.5**	**-2**	**-3**	**-4**	**-5**
angle (°):	0.0	0.1	0.0	22.3	29.2	31.5	32.6	33.8	34.2	34.5
linearized var.:	0.899	0.785	0.601	0.504	0.459	0.435	0.423	0.413	0.413	0.417
exact var.:	0.899	0.785	0.601	0.525	0.517	0.512	0.510	0.508	0.507	0.506
difference:	0.000	0.000	0.000	0.212	0.058	0.077	0.087	0.095	0.094	0.089
difference (%):	0.0	0.0	0.0	4.2	12.5	17.6	20.6	23.0	22.7	21.4

Table 1 Differences between methods for different values of c.



(a) T_pS_{-1} with sampled points and first principal components (blue exact PGA, green linearized PGA).

(b) S_{-1} with projected points and first principal components (blue exact PGA (8), green linearized PGA).

Fig. 3

Let us give a brief explanation of the result. The symmetry of the sphere and the dataset cause the effect of curvature to even out in the spherical case S_1. The cylinder S_0 has local geometry equal to $\mathbb{R}^2$ which causes the equality between the methods in the $c = 0$ case. The hyperboloids with $c < 0$, which can be constructed by revolving a hyperbola around its semi-minor axis, are non-symmetric causing an increase in variance as the first principal direction approaches the hyperbolic axis. The effect increases with the curvature causing the first principal direction to align with the hyperbolic axis for large negative values of c. We see that, for all negative values of c, exact PGA is able to capture more variance in the subspace spanned by the first principal direction than linearized PGA.

Using (5), we can approximate the sectional curvature K_p of S_c at p. The approximation is dependent on the value of the positive scalar t with increasing precision as t decreases to zero. Table 2 shows the result of the sectional curvature approximation for two values of t compared to the real curvature.

c:	**1**	**0**	**-1**	**-2**	**-3**
K_p:	1	0	-1	-2	-3
K_p est., $t = 0.01$:	1.000	0.000	-1.000	-2.000	-3.000
K_p est., $t = 0.1$:	1.000	0.000	-1.001	-2.002	-3.005

Table 2 Sectional curvature at p for different values of c.

Now let J_t be the Jacobi field with $J_0 = 0$ and $\frac{D}{dt}J_0 = (1, 0, 0)^T$ along the geodesic $x_t = \mathrm{Exp}_p t(0, 1, 0)^T$. Figure 4 shows $\|J_t\|$ for different values of c. We see that $\|J_\pi\| = 0$ for the spherical case S_1 showing that x_1 is a conjugate point and hence giving the upper bound π on the injectivity radius. The situation is illustrated in Figure 1. The local geometric equivalence between the cylinder S_0 and $\mathbb{R}^2$ causes the straight line for $c = 0$. For all $c \leq 1$, the injectivity radius of S_c is π, but for $c < 1$, the point x_π not a conjugate point[5]. By looking at $\|J_t\|$, we are only able to detect conjugate points and hence, with this experiment, we only

[5] For $c < 1$, x_π is a *cut* point [7, Chap. 13].

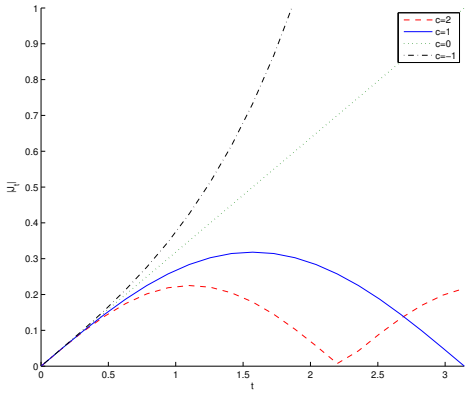


Fig. 4 $\|J_t\|$ for $c = 2, 1, 0, -1$ when $J_0 = 0$, $\frac{D}{dt}J_0 = (1, 0, 0)^T$, and $x_t = \mathrm{Exp}_p t(0, 1, 0)^T$.

get the bound on the injectivity radius for $c \geq 1$. For $c > 1$ the injectivity radius decreases below 1 as seen in the case S_2 with $\|J_{\tilde{t}}\| = 0$ for $\tilde{t} \approx \pi/\sqrt{2}$.

To investigate the difference with more than one principal direction, we consider a four dimensional manifold embedded in $\mathbb{R}^5$ and defined by

$$M_4 = \{(x_1, x_2, x_3, x_4, x_5) | x_1^2 - 2x_2^2 + x_3^2 - 2x_4 + x_5 = 1\} \ .$$

We make the situation more realistic than in the previous experiment by sampling 32 random points in the tangent space T_pM_4, $p = (0, 0, 0, 0, 1)$. Since T_pM_4 is an affine subspace of $\mathbb{R}^5$ orthogonal to the x_5 axis, we can identify it with $\mathbb{R}^4$ by the map $(x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3, x_4, 1)$. We use this identification when sampling by defining a normal distribution in $\mathbb{R}^4$, sampling the 32 points from the distribution, and mapping the results to T_pM_4. The covariance is set to $\Sigma = \mathrm{diag}(2, 1, 2/3, 1/3)$ to get non-spherical distribution and to increase the probability of data spreading over high-curvature parts of the manifold. Table 3 lists the variances and variance differences for the four principal directions for both methods along with angular differences. The lower variance for exact PGA compared to the linearized method for the 2nd principal direction is due to the greedy definition of PGA; when maximizing variance for the 2nd principal direction, we keep the first principal direction fixed. Hence we may get lower variance than what is obtainable if we were to maximize for both principal directions together.

Princ. comp.:	**1**	**2**	**3**	**4**
angle (°):	10.1	10.6	12.0	12.2
linearized var.:	1.58	3.86	4.13	4.35
exact var.:	1.93	3.85	4.24	4.35
difference:	0.35	-0.01	0.11	0.00
difference (%):	21.9	-0.3	2.6	0.0

Table 3 Differences between the methods on M_4. The variances of the data projected to the subspaces spanned by the first k principal directions and the percentage and angular differences are shown for $k = 1, \ldots, 4$.

We clearly see angular differences between the principal directions. In addition, there is significant difference in accumulated variance in the first and third principal direction. We note that the percentage difference is calculated from what corresponds to the accumulated spectrum. The percentage difference of the increase between the second and third principal direction, corresponding to the squared length of the third eigenvalue in regular PCA, is greater.

6 Conclusion

We have developed initial value problems allowing the computation of several important geometric structures on both parametrized and implicitly represented manifolds. We show how the constructed IVPs allow for numerical computation of injectivity radius bounds and sectional curvatures, which partially solves an open problem stated in [18]. Furthermore, the IVPs make possible computation of exact Principal Geodesic Analysis eliminating the need for the traditionally used linear approximations.

The experimental section presents examples of manifold valued datasets where exact PGA improves linearized PGA, and we show how the differences between the methods are dependent on the curvature of the manifolds. The differences are significant and clearly visually identifiable.

We are currently in the process of extending the methods to work for quotient manifolds M/G and thereby allowing the computations to be performed on practically all commonly occurring non-triangulated manifolds. We expect this would allow Geodesic PCA to be computed on general quotient manifolds as well. In addition, we are working on giving a theoretical treatment of the differences between the two formulations (8) of PGA. Finally, we expect to use the automatic computation of sectional curvatures to investigate further the effect of curvature on exact PGA and other statistical methods for manifold valued data.

Acknowledgements

The authors would like to thank P. Thomas Fletcher for fruitful discussions on how to compute exact PGA and Nicolas Courty for important remarks on problems linked to data locality. Furthermore, we wish to thank Søren Hauberg and Morten Engell-Nørregård for their help with producing data for testing the algorithms on high dimensional manifolds.

A Expressions for the Derivative ODEs

We will use tensors on $\mathbb{R}^\eta$ and $\mathbb{R}^m$ for the proofs of Theorem 2 and Theorem 3, and we will use the common identification between tensors and multilinear maps, i.e. the tensor $T : (\mathbb{R}^k)^r \to \mathbb{R}$ defines a map multilinear map $\tilde{T} : (\mathbb{R}^k)^{r-1} \to \mathbb{R}^k$ by $\left\langle \tilde{T}(y_1, \dots, y_{r-1}), y_r \right\rangle = T(y_1, \dots, y_r)$. We will not distinguish between a tensor and its corresponding multilinear map, and hence, in the above case, write T for both maps.

For s-dependent vector fields $v_{s,1}, \dots, v_{s,r}$ and tensor field T_s, we will use the equality

$$\begin{aligned} &\tfrac{d}{ds} T_0(v_{0,1}, \dots, v_{0,r}) \\ &\quad = \left(\tfrac{d}{ds} T_0\right)(v_{0,1}, \dots, v_{0,r}) + T_0(\tfrac{d}{ds} v_{0,1}, \dots, v_{0,r}) + \cdots + T_0(y_{0,1}, \dots, \tfrac{d}{ds} v_{0,r}) \end{aligned} \tag{19}$$

for the derivative with respect to s. If T_{x_s} is a composition of an z-dependent tensor field T_z and an s-dependent curve x_s, the derivative $\frac{d}{ds} T_{x_s}$ equals the covariant tensor derivative $\nabla_{\frac{d}{ds} x_s} T_{x_s}$ [7, Chap. 4]. Since we will only use tensors on Euclidean spaces, such tensor derivatives will consist of component-wise derivatives.

In the following, we let T^P_z be the z-dependent 3-tensor on $\mathbb{R}^\eta$ defined by

$$T^P_z(v_1, v_2, v_3) = -\sum_{i,j,k}^{\eta} \Gamma^k_{ij}(z) v_1^i v_2^j v_3^k$$

such that the kth component of $T^P_{x_t}(\dot{x}_t, \dot{x}_t)$ equals the right hand side of (1). Note that T^p_z is symmetric in the first two components since the Christoffel symbols are symmetric in i and j. Similarly, we let the z-dependent 3-tensor $T^{I,p}_z$ and 2-tensor $T^{I,x}_z$ equal the right hand side of the p and x parts of (2), respectively:

$$\begin{aligned} T^{I,p}_z(v_1, v_2) &= -\left(\sum_{k=1}^{n} \mu^k(z, v_1) H_z(F^k)\right) v_2 \ , \\ T^{I,x}_z(v) &= \left(I - D_z F^\dagger D_z F\right) v \end{aligned}$$

We carry out the proof of Theorem 2 in two parts starting with the parametrized case.

Proof (Theorem 2) Let $x_{t,s}$ be a family of geodesics with $x_{t,0} = x_t$, and define $q_s = x_{0,s}$ and $v_s = \dot{x}_{0,s}$. Assuming $\frac{d}{ds} q_0 = u$ and $\frac{d}{ds} v_0 = w$, the Jacobi field J_t equals $\frac{d}{ds} \mathrm{Exp}_{q_0}(t v_0)$, and, therefore, we can obtain J_t by differentiating the systems (1) and (2).

In the parametrized case, we get, using (19) and symmetry of T^P_z,

$$\begin{aligned} \tfrac{d}{dt^2} \tfrac{d}{ds} x_{t,0} = \tfrac{d}{ds} \ddot{x}_{t,0} &= \tfrac{d}{ds} T^P_{x_{t,0}}(\dot{x}_{t,0}, \dot{x}_{t,0}) \\ &= \nabla_{\frac{d}{ds} x_{t,0}} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + 2 T^P_{x_{t,0}}(\tfrac{d}{dt} \tfrac{d}{ds} x_{t,0}, \dot{x}_t) \ , \\ \tfrac{d}{ds} x_{0,0} = u, \ \tfrac{d}{dt} \tfrac{d}{ds} x_{0,0} &= w \end{aligned} \tag{20}$$

because $x_{t,s}$ are solutions to (1) with initial conditions q_s and v_s. Therefore, setting $y_t = \frac{d}{dt} \frac{d}{ds} x_{t,0}$ and $z_t = \frac{d}{ds} x_{t,0}$, we get (10) with

$$F^P_{q,v}(t, \begin{pmatrix} y_t \\ z_t \end{pmatrix}) = \begin{pmatrix} \nabla_{z_t} T^P_{x_t}(\dot{x}_t, \dot{x}_t) + 2 T^P_{x_t}(y_t, \dot{x}_t) \\ y_t \end{pmatrix} \ .$$

As noted above, the derivative $\nabla_{\frac{d}{ds} x_{t,0}} T^P_{x_s}$ consists of just the component-wise derivatives of T^P_z, i.e. the derivatives of the Christoffel symbols.

For the implicit case, we use the map μ of section 2.4 to define the tensors

$$\begin{aligned} &T^\mu_z(v) = \mu(z, v) \ , \ T^H_z(v_1, v_2) = -\left(\sum_{k=1}^{n} v_1^k H_z(F^k)\right) v_2 \ , \\ &T^D_z(v) = (D_z F)\, v \ , \text{ and } T^{D^\dagger}_z(v) = (D_z F)^\dagger\, v. \end{aligned}$$

Note, in particular, that $T_z^{I,p}(v_1, v_2) = T_z^H(T_z^\mu(v_1), v_2)$. We claim that $\frac{d}{ds}\mathrm{Exp}_{q_0}(tv_0)$ equals the z-part of the solution of (11) with

$$\begin{aligned} &F^I_{q,v}\left(t, \begin{pmatrix} y_t \\ z_t \end{pmatrix}\right) \\ &\quad = \begin{pmatrix} T^{I,p}_{x_t}(p_t, \dot{z}_t) + \nabla_{z_t} T^H_{x_t}(T^\mu_{x_t}(p_t), \dot{x}_t) + T^H_{x_t}(T^\mu_{x_t}(y_t) - \Lambda(T^D_{x_t}, \nabla_{z_t} T^D_{x_t})^T p_t, \dot{x}_t) \\ T^{I,x}_{x_t}(y_t) - \Lambda(T^D_{x_t}, \nabla_{z_t} T^D_{x_t}) T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t} \nabla_{z_t} T^D_{x_t}(p_t) \end{pmatrix} . \end{aligned} \tag{21}$$

Here $p_t = p_{t,0}$ where $p_{t,s}$ are the p-parts of the solutions to (2) with initial conditions q_s and v_s. To justify the claim, we differentiate the system (2). Using (19), we get

$$\begin{aligned} \tfrac{d}{dt}\tfrac{d}{ds} p_{t,0} &= \tfrac{d}{ds}\dot{p}_{t,0} = \tfrac{d}{ds} T^{I,p}_{x_{t,0}}(p_{t,0}, \dot{x}_{t,0}) \\ &= \nabla_{\frac{d}{ds}x_{t,0}} T^H_{x_t}(T^\mu_{x_t}(p_t), \dot{x}_t) + T^H_{x_t}(\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t}(p_t) + T^\mu_{x_t}(\tfrac{d}{ds}p_{t,0}), \dot{x}_t) \\ &\quad + T^{I,p}_{x_t}(p_t, \tfrac{d}{ds}\dot{x}_{t,0}) \end{aligned}$$

and

$$\tfrac{d}{dt}\tfrac{d}{ds} x_{t,0} = \tfrac{d}{ds}\dot{x}_{t,0} = \tfrac{d}{ds} T^{I,x}_{t,0}(p_{t,0}) = \nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t}(p_t) + T^{I,x}_{x_t}(\tfrac{d}{ds}p_{t,0}) .$$

Note that the tensor derivative $\nabla_{\frac{d}{ds}x_{t,0}} T^H_{x_t}$ consists of derivatives of $H_{x_t}(F^k)$. Both the derivatives $\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t}$ and $\nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t}$ involve derivatives of generalized inverses. Therefore, we apply Theorem 1 to differentiate $T^\mu_{x_t}$ and get that

$$\nabla_{\frac{d}{ds}x_{t,0}} T^\mu_{x_t} = -\Lambda(T^D_{x_t}, \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t})^T .$$

The tensor derivative $\nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}$ consists of derivatives of $D_{x_{t,s}}F$. Similarly,

$$\nabla_{\frac{d}{ds}x_{t,0}} T^{I,x}_{x_t} = -\Lambda(T^D_{x_t}, \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t}) T^D_{x_t} - T^{D^\dagger}_{x_t} \nabla_{\frac{d}{ds}x_{t,0}} T^D_{x_t} .$$

By differentiating the initial conditions, we get (11) with $y = \frac{d}{ds}p_{t,0}$, $z = \frac{d}{ds}x_{t,0}$, and $F^I_{q,v}$ as defined in (21).

For computing the second derivatives and proving Theorem 3, we will need to differentiate generalized inverses of matrices twice. For this task, we will use the lemma below, which follows directly from repeated application of the product rule for differentiation and Theorem 1.

Lemma 1 *Let $A_{t,s}$ be s- and t-dependent matrices. If $A_{t,s}$ and $A^\dagger_{t,s}$ are differentiable with respect to both variables and the mixed partial derivative $\frac{\partial^2}{\partial s \partial t} A_{t,s}$ exists, then*

$$\tfrac{\partial^2}{\partial s \partial t}(A^\dagger_{t,s}) = \tilde{\Lambda}(A_{t,s}, \tfrac{\partial}{\partial t}A_{t,s}, \tfrac{\partial}{\partial s}A_{t,s}, \tfrac{\partial^2}{\partial s \partial t}A_{t,s})$$

where

$$\begin{aligned} \tilde{\Lambda}(A,B,C,D) &= -\Lambda(A,C)BA^\dagger - A^\dagger D A^\dagger - A^\dagger B \Lambda(A,C) + Y(A,B,C,D) \\ &\quad - \Big(\Lambda(A,C)A + A^\dagger C\Big) X(A,B) A A^\dagger - A^\dagger A Y(A,B,C,D) A A^\dagger \\ &\quad - A^\dagger A X(A,B)\Big(CA^\dagger + A\Lambda(A,C)\Big) \end{aligned}$$

and

$$\begin{aligned} X(A,B) &= B^T(A^\dagger)^T A^\dagger + A^\dagger (A^\dagger)^T B^T , \\ Y(A,B,C,D) &= D^T(A^\dagger)^T A^\dagger + B^T\Big(\Lambda(A,C)^T A^\dagger + (A^\dagger)^T \Lambda(A,C)\Big) \\ &\quad + \Big(\Lambda(A,C)(A^\dagger)^T + A^\dagger(\Lambda(A,C))^T\Big) B^T + A^\dagger (A^\dagger)^T D^T . \end{aligned}$$

We are then ready to prove Theorem 3. We will again start with the parameterized case, and we will use the tensors introduced in the beginning of this section and the proof of Theorem 2.

Proof (Theorem 3) We compute the q and r parts of $G^P_{q,v_0,w,u}$ separately; denote them $G^{P,q}_{q,v_0,w,u}$ and $G^{P,r}_{q,v_0,w,u}$, respectively. Let $(y^w_{t,s}, z^w_{t,s})$ be solutions to (10) with IV's $(w,0)^T$ and along the geodesics $x_{t,s}$, and let y^w_t and z^w_t denote $y^w_{t,0}$ and $z^w_{t,0}$, respectively. Let also (y^u_t, z^u_t) be solutions to (10) with IV's $(u,0)^T$ along $x_t = x_{t,0}$. Differentiating system (10), we get

$$\tfrac{d}{dt}\tfrac{d}{ds}(z^w_{t,0}) = \tfrac{d}{ds}(\dot z^w_{t,0}) = \tfrac{d}{ds}(y^w_{t,0})$$

and, using symmetry of the tensors,

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds}(y^w_{t,0}) &= \tfrac{d}{ds}(\dot y^w_{t,0}) = \tfrac{d}{ds}\nabla_{z^w_{t,0}} T^P_{x_{t,0}}(\dot x_{t,0}, \dot x_{t,0}) + 2\tfrac{d}{ds} T^P_{x_{t,0}}(y^w_{t,0}, \dot x_{t,0}) \\ &= \nabla_{z^u_t}\nabla_{z^w_t} T^P_{x_t}(\dot x_t, \dot x_t) + \nabla_{\frac{d}{ds} z^w_{t,0}} T^P_{x_t}(\dot x_t, \dot x_t) + 2\nabla_{z^w_t} T^P_{x_t}(y^u_t, \dot x_t) \\ &\quad + 2\nabla_{z^u_t} T^P_{x_t}(y^w_t, \dot x_t) + 2T^P_{x_t}(\tfrac{d}{ds} y^w_{t,0}, \dot x_t) + 2T^P_{x_t}(y^w_t, y^u_t)\ .\end{aligned} \tag{22}$$

Therefore, letting $q_t = \frac{d}{ds} y^w_{t,0}$ and $r_t = \frac{d}{ds} z^w_{t,0}$, we get $G^{P,q}_{q,v_0,w,u}(t, (r_t\ q_t)^T)$ as the right hand side of (22) and $G^{P,r}_{q,v_0,w,u}(t, (r_t\ q_t)^T)$ equal to q_t. The initial values are both 0 since $y^w_{0,s}$ and $z^w_{0,s}$ equal 0 and w, respectively, and, therefore, are not s-dependent.

For the implicit case, we will again compute the r and q parts of $G^I_{q,v_0,w,u}$ separately. Let now $(y^w_{t,s}, z^w_{t,s})$ be solutions to (11) along the geodesics $x_{t,s}$ and with IV's $(w,0)^T$, and let (y^u_t, z^u_t) be solutions to (11) along x_t and with IV's $(u,0)^T$. Let also $p_{t,s}$ denote the p-parts of the solutions to (2) with initial conditions q and v_s, and write $p_t = p_{t,0}$, $y^w_t = y^w_{t,0}$, and $z^w_t = z^w_{t,0}$.

Differentiating system (11), we get

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds} y^w_{t,0} = \tfrac{d}{ds}\dot y^w_{t,0} &= \tfrac{d}{ds} T^{I,p}_{x_{t,0}}(p_{t,0}, \dot z^w_{t,0}) + \tfrac{d}{ds}\nabla_{z^w_{t,0}} T^H_{x_{t,0}}(T^\mu_{x_{t,0}}(p_{t,0}), \dot x_{t,0}) \\ &\quad + \tfrac{d}{ds} T^H_{x_{t,0}}(T^\mu_{x_{t,0}}(y^w_{t,0}) - \Lambda(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}})^T p_{t,0}, \dot x_{t,0})\ .\end{aligned}$$

Using the map $\tilde\Lambda$ defined in Lemma 1, we have

$$\tfrac{d}{ds}\Lambda(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}})^T = \tilde\Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t})^T\ .$$

Combining the equations, we get

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds} y^w_{t,0} &= \nabla_{z^u_t} T^{I,p}_{x_t}(p_t, \dot z^w_t) + T^{I,p}_{x_t}(y^u_t, \dot z^w_t) + T^{I,p}_{x_t}(p_t, \tfrac{d}{dt}\tfrac{d}{ds} z^w_{t,0}) \\ &\quad + \nabla_{z^u_t}\nabla_{z^w_t} T^H_{x_t}(T^\mu_{x_t}(p_t), \dot x_t) + \nabla_{\frac{d}{ds} z^w_{t,0}} T^H_{x_t}(T^\mu_{x_t}(p_t), \dot x_t) \\ &\quad + \nabla_{z^w_t} T^H_{x_t}(T^\mu_{x_t}(y^u_t) - \Lambda(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t})^T p_t, \dot x_t) + \nabla_{z^w_t} T^H_{x_t}(T^\mu_{x_t}(p_t), \dot z^u_t) \\ &\quad + \nabla_{z^u_t} T^H_{x_t}(T^\mu_{x_t}(y^w_t) - \Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t})^T p_t, \dot x_t) \\ &\quad + T^H_{x_t}(T^\mu_{x_t}(\tfrac{d}{ds} y^w_{t,0}) - \Lambda(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t})^T y^w_t, \dot x_t) \\ &\quad - T^H_{x_t}(\tilde\Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t})^T p_t + \Lambda(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t})^T y^u_t, \dot x_t) \\ &\quad + T^H_{x_t}(T^\mu_{x_t}(y^w_t) - \Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t})^T p_t, \dot z^u_t)\ .\end{aligned}$$

Substituting $\frac{d}{ds} z^w_{t,0}$ with r_t and $\frac{d}{ds} y^w_{t,0}$ with q_t, we get $G^{I,q}_{q,v_0,w,u}$ as the right hand side of the equation. Likewise,

$$\begin{aligned}\tfrac{d}{dt}\tfrac{d}{ds} z^w_{t,0} &= \tfrac{d}{ds} T^{I,x}_{x_{t,0}}(y^w_{t,0}) - \tfrac{d}{ds}\Lambda(T^D_{x_{t,0}}, \nabla_{z^w_{t,0}} T^D_{x_{t,0}}) T^D_{x_{t,0}}(p_{t,0}) - \tfrac{d}{ds} T^{D^\dagger}_{x_{t,0}}\nabla_{z^w_{t,0}} T^D_{x_{t,0}}(p_{t,0}) \\ &= \nabla_{z^u_t} T^{I,x}_{x_t}(y^w_t) + T^{I,x}_{x_t}(\tfrac{d}{ds} y^w_{t,0}) \\ &\quad - \tilde\Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t}, \nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}) T^D_{x_t}(p_t) \\ &\quad - \Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t})\nabla_{z^u_t} T^D_{x_t}(p_t) - \Lambda(T^D_{x_t}, \nabla_{z^w_t} T^D_{x_t}) T^D_{x_t}(y^u_t) \\ &\quad - \Lambda(T^D_{x_t}, \nabla_{z^u_t} T^D_{x_t})\nabla_{z^w_t} T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t}\nabla_{z^u_t}\nabla_{z^w_t} T^D_{x_t}(p_t) - T^{D^\dagger}_{x_t}\nabla_{z^w_t} T^D_{x_t}(y^u_t)\ .\end{aligned}$$

Again, after substituting $\frac{d}{ds} y^w_{t,0}$ with q_t as above, we get $G^{I,r}_{q,v_0,w,u}$ as the right hand side of the equation. As for the parametric case, both initial values are zero.

B The Projection Gradient

We prove Theorem 4, and, following this, we show how to compute the Hessian of the residual function $R_{x,\mu}$. We will need the following result for the proof of Theorem 4 to show that equation (16) is independent of the chosen basis.

Lemma 2 *Let S be an open subset of $\mathbb{R}^k$ and $U : S \to M^{k\times(k-1)}$ a C^1 map with the property that for any $v \in S$, the columns of the matrix $(\frac{v}{\|v\|}\ U(v))$ constitute an orthonormal basis for $\mathbb{R}^k$. Let u^j_v denote the jth column of $U(v)$. Then for any $v_0 \in S$ and $w \in \mathbb{R}^k$, $\left\langle \frac{d}{dt} u^j_{v_0+tw}|_{t=0}, v_0 \right\rangle = -\left\langle u^j_{v_0}, w \right\rangle$. As consequence of this, if $\tilde{U} : S \to \mathbb{R}^{k-1}$ denotes the map $v \mapsto U(v)^T \frac{v_0}{\|v_0\|}$ then*

$$D^{v\in \mathrm{span}\,(u^1_{v_0},\ldots,u^{k-1}_{v_0})}_{v_0} \tilde{U}(v) = -I_{k-1}$$

in the basis $u^1_{v_0}, \ldots, u^{k-1}_{v_0}$ for $\mathrm{span}\,(u^1_{v_0}, \ldots, u^{k-1}_{v_0})$.

In the proof below, we adopt the notation of section 4.2, but we will use the alternative formulation $R_{x,\mu}(w) = \|\mathrm{Log}_x \mathrm{Exp}_\mu w\|^2$ for the residual function.

Proof (Theorem 4) Extend the basis $\{v^1, \ldots, v^k, v_0/\|v_0\|\}$ for V_{v_0} to an orthonormal basis for $T_\mu M$. The argument is not dependent on this choice of basis, but it will make the reasoning and notation easier. Let $S \subset T_\mu M \times V^\perp$ be an open neighborhood of (w_0, v_0) and define the map $F_V : S \to \mathbb{R}^\eta$ by

$$F_V(w,v) = \begin{pmatrix} \nabla_w R_{x,\mu} \cdot v^1 \\ \vdots \\ \nabla_w R_{x,\mu} \cdot v^k \\ \nabla_w R_{x,\mu} \cdot v \\ w \cdot u^1(v) \\ \vdots \\ w \cdot u^{\eta-k-1}(v) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} V & v \end{pmatrix}^T \nabla_w R_{x,\mu} \\ U_v^T w \end{pmatrix}$$

with the vectors $u^1(v), \ldots, u^{\eta-(k+1)}(v)$ constituting an orthonormal basis for $V_v^\perp$ for each v and with $(V\ v)$ and U_v denoting the matrices having v^i, v and $u^i(v)$ in the columns, respectively. Since $\langle \nabla_{w_0} R_{x,\mu}, v\rangle = d_{w_0} R_{x,\mu}(v) = 0$ for all $v \in V_{v_0}$ because w_0 is a minimizer for $R_{x,\mu}$ among vectors in in V_{v_0}, we see that $F_V(w_0, v_0)$ vanishes. Therefore, if $D^w_{(w_0,v_0)} F_V$ is non-singular, the implicit function theorem asserts the existence of a map Ψ from a neighborhood of v_0 to $T_\mu M$ with the property that $F_V(\Psi(v), v) = 0$ for all v in the neighborhood. We then compute

$$0 = D_{v_0} F_V(\Psi(v), v) = \left(D^w_{(w_0,v_0)} F_V\right)(D_{v_0}\Psi(v)) + \left(D^v_{(w_0,v_0)} F_V\right)$$

and hence

$$D^{v\in V^\perp_{v_0}}_{v_0} \Psi(v) = -\left(D^w_{(w_0,v_0)} F_V\right)^{-1} \left(D^{v\in V^\perp_{v_0}}_{(w_0,v_0)} F_V\right) . \tag{23}$$

For the differentials on the right hand side of (23), we have

$$D^{v\in V^\perp_{v_0}}_{(w_0,v_0)} F_V = \left(0\ \cdots\ 0\ \ \nabla^{w\in V^\perp_{v_0}}_{w_0} R_{x,\mu}\ \ D^{v\in V^\perp_{v_0}}_{(w_0,v_0)} \left(w_0^T U_v\right)\right)^T$$

and

$$D^w_{(w_0,v_0)} F_V = \begin{pmatrix} \begin{pmatrix} V & v_0 \end{pmatrix}^T d^w_{w_0}(\nabla_w R_{x,\mu}) \\ U^T_{v_0} \end{pmatrix} = \begin{pmatrix} \left(H_{w_0}(R_{x,\mu}) \begin{pmatrix} V & v_0 \end{pmatrix}\right)^T \\ U^T_{v_0} \end{pmatrix} . \tag{24}$$

With the choice of basis, the above matrix is block triangular,

$$D^{w}_{(w_0,v_0)}F_V = \begin{pmatrix} A_{w_0,v_0} & B_{w_0,v_0} \\ 0 & C_{w_0,v_0} \end{pmatrix} , \tag{25}$$

with A_{w_0,v_0} equal to $H_{w_0}(R_{x,\mu}|_{V_{v_0}})$. The requirement that $D^{w}_{(w_0,v_0)}F_V$ is non-singular is fulfilled, because $H_{w_0}(R_{x,\mu}|_{V_{v_0}})$ has rank $k+1$ by assumption and U_{v_0} has rank $\eta-(k+1)$.

Since the first k rows of $D^{v\in V_{v_0}^{\perp}}_{(w_0,v_0)}F_V$ are zero, we need only the last $\eta-k$ columns of $(D^{w}_{(w_0,v_0)}F_V)^{-1}$ in order to compute (23). The vector $\bar{v}_{x,\mu,v_0,S_{v_0}}$ as defined in the statement of the theorem is equal to the $(k+1)$st column. Let $E_{x,\mu,v_0,S_{v_0}}$ be the matrix consisting of the remaining $\eta-(k+1)$ columns. Using the form (25), we have

$$E_{x,\mu,v_0,S_{v_0}} = \begin{pmatrix} -H_{w_0}\left(R_{x,\mu}|_{V_{v_0}}\right)^{-1} B_{w_0,v_0}C^{-1}_{w_0,v_0} \\ C^{-1}_{w_0,v_0} \end{pmatrix} .$$

Assume $\{u^1,\dots,u^j\}$ is chosen such that $\{u^1(v_0),\dots,u^j(v_0)\}$ equals the previously chosen basis for $V_{v_0}^{\perp}$. With this assumption, C_{w_0,v_0} is the identity matrix $I_{\eta-(k+1)}$. In addition, let w_0^{k+1} denote the $(k+1)$st component of w_0, that is, the projection of w_0 onto $v_0/\|v_0\|$. Since $w_0\in V_{v_0}$ and by choice of U_v, Lemma 2 gives

$$D^{v\in V_{v_0}^{\perp}}_{(w_0,v_0)}\left(U_v^T w\right) = w_0^{k+1}D^{v\in V_{v_0}^{\perp}}_{(w_0,v_0)}\left(U_v^T\tfrac{v_0}{\|v_0\|}\right) = -w_0^{k+1}I_{\eta-(k+1)} .$$

Therefore,

$$D^{v\in V_{v_0}^{\perp}}_{(w_0,v_0)}F_V = \begin{pmatrix} 0 & \cdots & 0 & \nabla^{w\in V_{v_0}^{\perp}}_{w_0}R_{x,\mu} & -w_0^{k+1}I_{\eta-(k+1)} \end{pmatrix}^T .$$

Note, in particular, that $D^{v\in V_{v_0}^{\perp}}_{(w_0,v_0)}F_V$ is independent on the actual choice of bases U_v. Combining the equations, we get

$$D^{v\in V_{v_0}^{\perp}}_{v_0}\Psi(v) = -\bar{v}_{x,\mu,v_0,S_{v_0}}(\nabla^{w\in V_{v_0}^{\perp}}_{w_0}R_{x,\mu})^T + w_0^{k+1}E_{x,\mu,v_0,S_{v_0}} .$$

Because $\mathrm{Exp}_\mu\Psi(v)=\pi_{S_v}(x)$, we get (16).

Lets now compute second derivatives, and thereby the Hessian, of the residual function $R_{x,\mu}$. Choose $w_0,v\in T_\mu M$ and let $y=\mathrm{Exp}_\mu w_0$. Working in the orthonormal basis, we have

$$\nabla_{w_0}R_{x,\mu} = 2\left((D_y\mathrm{Log}_x)\,D_{w_0}\mathrm{Exp}_\mu\right)^T\mathrm{Log}_x y .$$

and hence

$$\begin{aligned} &\tfrac{d}{ds}\left(\nabla_{w_0+vs}R_{x,\mu}\right)|_{s=0} \\ &\quad= 2\left(\tfrac{d}{ds}\left(D_{\mathrm{Exp}_\mu(w_0+sv)}\mathrm{Log}_x\right)|_{s=0}\left(D_{w_0}\mathrm{Exp}_\mu\right)\right)^T\mathrm{Log}_x y \\ &\qquad+2\left((D_y\mathrm{Log}_x)\,\tfrac{d}{ds}\left(D_{w_0+vs}\mathrm{Exp}_\mu\right)|_{s=0}\right)^T\mathrm{Log}_x y \\ &\qquad+2\left((D_y\mathrm{Log}_x)\left(D_{w_0}\mathrm{Exp}_\mu\right)\right)^T\tfrac{d}{ds}\left(\mathrm{Log}_x\mathrm{Exp}_\mu(w_0+sv)\right)|_{s=0} . \end{aligned} \tag{26}$$

Note that

$$\tfrac{d}{ds}\left(\mathrm{Log}_x\mathrm{Exp}_\mu(w_0+sv)\right)|_{s=0} = (D_y\mathrm{Log}_x)\left(D_{w_0}\mathrm{Exp}_\mu\right)v .$$

Using that $\frac{d}{ds}(A_s^{-1}) = A_s^{-1}(\frac{d}{ds}A_s)A_s^{-1}$ for a time depedent, invertible matrix A_s[6] and the fact that $\mathrm{Exp}_x\mathrm{Log}_x z = z$ for all z, we get

$$\begin{aligned} \tfrac{d}{ds}\left(D_{\mathrm{Exp}_\mu(w_0+sv)}\mathrm{Log}_x\right)|_{s=0} &= \tfrac{d}{ds}\left(D_{\mathrm{Log}_x(\mathrm{Exp}_\mu w_0+sv)}\mathrm{Exp}_x\right)^{-1}|_{s=0} \\ &= -\left(D_y\mathrm{Log}_x\right)\tfrac{d}{ds}\left(D_{\mathrm{Log}_x(\mathrm{Exp}_\mu w_0+sv)}\mathrm{Exp}_x\right)|_{s=0}\left(D_y\mathrm{Log}_x\right) . \end{aligned}$$

The middle term of this product and the term $\frac{d}{ds}\left(D_{w_0+sv}\mathrm{Exp}_\mu\right)|_{s=0}$ in (26) are both computable using Theorem 3.

[6] See [5, Eq. (2)].

References

1. Harry Blum and Weiant Wathen-Dunn, *A transformation for extracting new descriptors of shape*, Models for the Perception of Speech and Visual Form (1967), 380, 362.
2. Jonathan Boisvert, Farida Cheriet, Xavier Pennec, Nicholas Ayache, and Hubert Labelle, *A novel framework for the 3D analysis of spine deformation modes*, Studies in Health Technology and Informatics **123** (2006), 176–181, PMID: 17108423.
3. ———, *Principal deformations modes of articulated models for the analysis of 3D spine deformities*, Electronic Letters on Computer Vision and Image Analysis **7** (2008), no. 4.
4. Vicent Caselles, Ron Kimmel, and Guillermo Sapiro, *Geodesic active contours*, International Journal of Computer Vision **22** (1995), 61—79.
5. Henry P. Decell, *On the derivative of the generalized inverse of a matrix*, Linear and Multilinear Algebra **1** (1974), no. 4, 357.
6. Jean-Pierre Dedieu and Dmitry Nowicki, *Symplectic methods for the approximation of the exponential map and the newton iteration on Riemannian submanifolds*, Journal of Complexity **21** (2005), no. 4, 487–501.
7. Manfredo Perdigao do Carmo, *Riemannian geometry*, Mathematics: Theory & Applications, Birkhauser Boston Inc., Boston, MA, 1992.
8. Ricardo Ferreira, Joao Xavier, Joaoa Costeria, and Victor Barroso, *Newton algorithms for Riemannian distance related problems on connected locally symmetric manifolds*, Technical Report, Signal and Image Processing Group (SPIG), Institute for Systems and Robotics (ISR) (2008).
9. P. Thomas Fletcher and Sarang Joshi, *Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors*, ECCV Workshops CVAMIA and MMBIA. **3117** (2004), 87—98.
10. ———, *Riemannian geometry for the statistical analysis of diffusion tensor data*, Signal Processing **87** (2007), no. 2, 250–262.
11. P. Thomas Fletcher, Suresh Venkatasubramanian, and Sarang Joshi, *Robust statistics on Riemannian manifolds via the geometric median*, 2008 IEEE Conference on Computer Vision and Pattern Recognition (Anchorage, AK, USA), June 2008, pp. 1–8.
12. P.T. Fletcher, Conglin Lu, and S. Joshi, *Statistics of shape via principal geodesic analysis on Lie groups*, CVPR 2003, vol. 1, 2003, pp. I–95–I–101 vol.1.
13. P.T. Fletcher, Conglin Lu, S.M. Pizer, and Sarang Joshi, *Principal geodesic analysis for the study of nonlinear statistics of shape*, Medical Imaging, IEEE Transactions on **23** (2004), no. 8, 995–1005.
14. M. Fréchet, *Les éléments aléatoires de nature quelconque dans un espace distancié*, Ann. Inst. H. Poincare **10** (1948), 215–310.
15. Ernst Hairer, Christian Lubich, and Gerhard Wanner, *Geometric numerical integration*, Springer, 2002.
16. Ernst Hairer, Syvert P. N?rsett, and Gerhard Wanner, *Solving ordinary differential equations i: Nonstiff problems (Springer series in computational mathematics)*, 2nd ed., Springer, May 2008.
17. Stephan Huckemann and Thomas Hotz, *Principal component geodesics for planar shape spaces*, Journal of Multivariate Analysis **100** (2009), no. 4, 699–714.
18. Stephan Huckemann, Thomas Hotz, and Axel Munk, *Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric Lie group actions*, Statistica Sinica **20** (2010), no. 1, 1–100.
19. Stephan Huckemann and Herbert Ziezold, *Principal component analysis for Riemannian manifolds, with an application to triangular shape spaces*, Advances in Applied Probability **38** (2006), no. 2, 299–319.
20. Sarang Joshi, Stephen Pizer, P Thomas Fletcher, Paul Yushkevich, Andrew Thall, and J S Marron, *Multiscale deformable model segmentation and statistical shape analysis using medial descriptions*, IEEE Transactions on Medical Imaging **21** (2002), no. 5, 538–550, PMID: 12071624.
21. H. Karcher, *Riemannian center of mass and mollifier smoothing*, Communications on Pure and Applied Mathematics **30** (1977), no. 5, 509–541.
22. Herbert Bishop Keller, *Numerical methods for two-point boundary-value problems*, Blaisdell, (Waltham, Mass), 1968.
23. David G. Kendall, *Shape manifolds, procrustean metrics, and complex projective spaces*, Bull. London Math. Soc. **16** (1984), no. 2, 81–121.

24. Eric Klassen and Anuj Srivastava, *Geodesics between 3D closed curves using Path-Straightening*, ECCV 2006, vol. 3951, Springer, 2006, pp. 95–106.
25. Eric Klassen, Anuj Srivastava, Washington Mio, and Shantanu Joshi, *Analysis of planar shapes using geodesic paths on shape spaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), 372—383.
26. John M Lee, *Riemannian manifolds*, Graduate Texts in Mathematics, vol. 176, Springer-Verlag, New York, 1997, An introduction to curvature.
27. Zhengdong Lu, Miguel Carreira-Perpinan, and Christian Sminchisescu, *People tracking with the Laplacian eigenmaps latent variable model*, Advances in Neural Information Processing Systems 20, MIT Press, 2008, pp. 1705–1712.
28. David G. Luenberger, *The gradient projection method along geodesics*, Management Science **18** (1972), no. 11, 620–631, ArticleType: primary_article / Issue Title: Theory Series / Full publication date: Jul., 1972 / Copyright © 1972 INFORMS.
29. Lyle Noakes, *A global algorithm for geodesics*, Journal of the Australian Mathematical Society **64** (1998), 37–50.
30. Xavier Pennec, *Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements*, J. Math. Imaging Vis. **25** (2006), no. 1, 127–154.
31. Xavier Pennec, Pierre Fillard, and Nicholas Ayache, *A Riemannian framework for tensor computing*, Int. J. Comput. Vision **66** (2006), no. 1, 41–66.
32. Xavier Pennec, Charles Guttmann, and Jean-Philippe Thirion, *Feature-based registration of medical images: Estimation and validation of the pose accuracy*, MICCAI 1998, Springer Berlin, 1998, pp. 1107–1114.
33. Patrick J. Rabier and Werner C. Rheinboldt, *On a computational method for the second fundamental tensor and its application to bifurcation problems*, Numerische Mathematik **57** (1990), no. 1, 681–694.
34. W. C. Rheinboldt, *MANPAK: a set of algorithms for computations on implicitly defined manifolds*, Computers & Mathematics with Applications **32** (1996), no. 12, 15–28.
35. Salem Said, Nicolas Courty, Nicolas Le Bihan, and Stephen Sangwine, *Exact principal geodesic analysis for data on so(3)*, EUSIPCO 2007 (2007).
36. Frank Schmidt, Michael Clausen, and Daniel Cremers, *Shape matching by variational computation of geodesics on a manifold*, Pattern Recognition, Springer Berlin, 2006, pp. 142–151.
37. Cristian Sminchisescu and Allan Jepson, *Generative modeling for continuous Non-Linearly embedded visual inference*, In ICML (2004), 759—766.
38. Stefan Sommer, Francois Lauze, Søren Hauberg, and Mads Nielsen, *Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations*, ECCV 2010 (Heraklion, Greece), Lecture Notes in Computer Science, vol. 6316, Springer, Heidelberg, 2010, pp. 43–56.
39. Stefan Sommer, Aditya Tatu, Chen Chen, Dan Jørgensen, Marleen de Bruijne, Marco Loog, Mads Nielsen, and Francois Lauze, *Bicycle chain shape models*, MMBIA/CVPR 2009, 2009, pp. 157–163.
40. Raquel Urtasun, David J. Fleet, Aaron Hertzmann, and Pascal Fua, *Priors for people tracking from small training sets*, 2005 IEEE International Conference on Computer Vision (ICCV), IEEE Computer Society, 2005, pp. 403–410.
41. Jing Wu, W. Smith, and E. Hancock, *Weighted principal geodesic analysis for facial gender classification*, Progress in Pattern Recognition, Image Analysis and Applications, Springer Berlin, 2008, pp. 331–339.
42. Y. Yang, *Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization*, Journal of Optimization Theory and Applications **132** (2007), no. 2, 245–265.
43. Laurent Younes, Felipe Arrate, and Michael I. Miller, *Evolutions equations in computational anatomy*, NeuroImage **45** (2009), no. 1, Supplement 1, S40–S50.
44. Qin Zhang and Guoliang Xu, *Curvature computations for n-manifolds in and solution to an open problem proposed by r. goldman*, Computer Aided Geometric Design **24** (2007), no. 2, 117–123.

6.

Paper #5: *Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations*

Peer-reviewed conference paper accepted for oral presentation at the European Conference on Computer Vision (ECCV) 2010, Heraklion, Greece, 2010.

Authors:
Stefan Sommer, François Lauze, Søren Hauberg, and Mads Nielsen

Notes:

With the algorithms developed in Paper #4, we are able to compute *exact PGA*. Based on experimentation, it became clear that in absence of both significant curvature and spread of the data, the original PGA algorithm is a fairly good approximation of the exact counterpart. In this paper, we present a comparison between the algorithms and evaluate them on two datasets. We investigate if easily computable indicators can predict when the approximate algorithm will perform well compared to its exact equivalent.

Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations

Stefan Sommer[1], François Lauze[1], Søren Hauberg[1], and Mads Nielsen[1,2]

[1] Dept. of Computer Science, Univ. of Copenhagen, Denmark
sommer@diku.dk
[2] Nordic Bioscience Imaging, Herlev, Denmark

Abstract. Manifolds are widely used to model non-linearity arising in a range of computer vision applications. This paper treats statistics on manifolds and the loss of accuracy occurring when linearizing the manifold prior to performing statistical operations. Using recent advances in manifold computations, we present a comparison between the non-linear analog of Principal Component Analysis, Principal Geodesic Analysis, in its linearized form and its exact counterpart that uses true intrinsic distances. We give examples of datasets for which the linearized version provides good approximations and for which it does not. Indicators for the differences between the two versions are then developed and applied to two examples of manifold valued data: outlines of vertebrae from a study of vertebral fractures and spacial coordinates of human skeleton end-effectors acquired using a stereo camera and tracking software.

Key words: manifolds, Riemannian metrics, linearization, manifold valued statistics, Principal Geodesic Analysis (PGA), Geodesic PCA

1 Introduction

This paper treats the effect of linearization when using the non-linear analog of Principal Component Analysis, Principal Geodesic Analysis (PGA, [1]), to estimate the variability in sets of manifold valued data. Until recently, PGA has been performed by linearizing the manifold, which distorts intrinsic distances, but with the introduction of more powerful computational tools [2], PGA can now be computed with true intrinsic distances. We show how simple and fast indicators allow us to approximate the differences between linearized PGA and exact PGA with true intrinsic distances and evaluate the effect of the linearization.

As a test case for the indicators, we perform a comparison between two manifold valued datasets: outlines of vertebrae from a study of vertebral fractures, and human skeleton end-effectors in spatial coordinates recorded using a stereo camera and tracking software. We will show that linearized PGA provides a reasonable approximation in only one of the experiments and that the indicators allow us to predict this before doing the time-intensive computation of exact PGA with intrinsic distances.

1.1 Motivation

A wide variety of problems in computer vision possess non-linear structure and are therefore naturally modeled using Riemannian geometry. In diffusion tensor imaging [3–5], for image segmentation [6] and registration [7], shape spaces [8], and human motion modeling [9, 10], Riemannian manifolds have been used to enforce consistency in data, provide dimensionality reduction, and define more accurate metrics. The wide applicability of manifolds in modeling problems has created the need for statistical tools for manifold data.

Generalizing linear statistical operations to manifolds [1, 11–13] provides examples of the theoretical and computational problems arising when departing from familiar Euclidean spaces. The tools developed when pursuing this have been used successfully for a range of computer vision applications, and the area is the subject of active research [2, 13]. Depending on the level of approximation used in the computations, manifold statistics can be hard to carry out in practice because operations such as finding distances and performing optimization do not admit the closed-form solutions often found in Euclidean spaces [1].

One way of doing manifold statistics is projecting the set of manifold valued data points to the tangent space of a mean point of the manifold. The vector space structure of the tangent space brings back convenient Euclidean statistics, but the distortion of the distances between the data points inherent in the linearization may however lead to sub-optimal solutions to the statistical problems. In contrast to this, some statistical operations can be carried out with true intrinsic manifold distances giving a true picture of the data [2, 13]. This, however, often comes at the cost of increased computational complexity and requires conditions on the locality of data.

Because of the trade-offs between convenient linearization and exact modeling, we seek for ways to evaluate the extent of the distortion between the linearized data and true manifold data; we are interested in determining if performing statistics with intrinsic distances offers significant advantages over the linearized approach. Such knowledge has the potential of saving substantial computation time and to improve results of statistical operations.

1.2 Related Work

The mathematical aspects of manifolds are covered extensively in the literature with [14, 15] providing good references. Numerical and computational aspects of interest in a general setting are considered in the theoretical papers [16, 17] while more specific shape related applications are proposed in [18–20].

Both the mathematical community, e.g. [11], and more applied fields, computer vision in particular [1, 12], have worked with different aspect of statistics on manifolds. A recent wave of interest by statisticians [21, 13] has created new methods with strong links to tools developed in computer vision [13].

The manifold generalization of linear PCA, PGA, was first introduced in [22], but it was formulated in the form most widely used in [1]. It has subsequently been used for several applications. To mention a few, the authors in

[1, 4] study variations of medial atoms, [23] uses a variation of PGA for facial classification, [24] presents examples on motion capture data, and [20] applies PGA to vertebrae outlines. The algorithm presented in [1] for computing PGA with linearization has been most widely used. In contrast to this, [24] computes PGA as defined in [22] without approximations, but only for a specific manifold, the Lie group SO(3). By using ODE formulations of geodesics and taking derivatives, [2] provides algorithms for computing PGA without approximations on wide classes of manifolds.

Geodesic PCA (GPCA, [13, 21]) is in many respects close to PGA but optimizes for the placement of the center point and minimizes projection residuals along geodesics instead of maximizing variance in geodesic subspaces. GPCA uses no linear approximation, but it is currently only computed on spaces where explicit formulas for geodesics exist and on quotients of such spaces.

1.3 Content and Outline

In the next section, we discuss the benefits of using manifolds in modeling, manifold valued statistics, and linearization. Then, in section 3, we consider in detail the specific case of Principal Geodesic Analysis and use synthetic examples to explain the differences between linearized PGA and exact PGA with true intrinsic distances. We progress to developing indicators of these differences, and, in section 4, we compare linearized and intrinsic PGA on real-life examples of manifold valued datasets and analyze the power of the indicators. The paper thus contributes by

(1) developing simple and fast indicators of the difference between linearized PGA and exact PGA that show the effect of linearization,
(2) giving examples of the differences between linearized PGA and exact PGA on real-life datasets from computer vision,
(3) and showing the power of the indicators when applied to the datasets.

2 Manifolds and Manifold Valued Statistics

The interest in manifolds as modeling tools arises from the non-linearity apparent in a variety of problems. We will in the following exemplify this by considering the pose of a human skeleton captured by e.g. a tracking system or motion capture equipment. Consider the position of a moving hand while the elbow and the rest of the body stay fixed. The hand cannot move freely as the length of the lower arm restricts it movement. Linear vector space structure is not present; if we multiply the position of the hand by a scalar, the length of the arm would in general change in order to accommodate the new hand position. Even switching to an angular representation of the pose of the elbow joint will not help; angles have inherent periodicity, which is not compatible with vector space structure.

Though the space of possible hand positions is not linear, it has the structure of a manifold since it possesses the property that it locally can be approximated

by a vector space. Furthermore, we can, in a natural way, equip it with a Riemannian metric [14], which allows us to make precise notions of length of curves on the space and intrinsic acceleration. This in turns defines the Riemannian manifold equivalent of straight lines: geodesics. The length of geodesics connecting points defines a distance metric on the manifold.

2.1 Benefits from Modeling using Manifolds

The main advantages of introducing manifolds in modeling are as follows: consistency in representation, dimensionality reduction, and accuracy in measurements. Consistency ensures the modeled object satisfies the requirements making up the manifold; when moving the position of the hand on the manifold, we are certain the length of the lower arm is kept constant. Such requirements reduce the number of degrees of freedom and hence provide dimensionality reduction. Consistency and dimensionality reduction are therefore closely linked.

Accuracy is connected to the distance measure defined by the Riemannian metric. A reasonable measure of the distance between two positions of the hand will be the length of the shortest curve arising when moving the hand between the positions. Such a curve will, in this example, be a circular arc, and, in the manifold model, the distance will be the length of the arc. In the vector space model, however, the distance will be the length of the straight line connecting the hand positions and, hence, will not reflect the length of an allowed movement of the hand. The manifold model therefore gives a more accurate distance measure.

2.2 Linearizing the Manifold

By linearizing the manifold to the tangent space of a mean point, we can in many applications ensure consistency, but not accuracy, in statistical operations. Let M be a manifold and $\{x_1, \ldots, x_N\}$ a dataset consisting of points on the manifold. An intrinsic mean [11] is defined as a solution to the optimization problem

$$\mu = \operatorname{argmin}_q \sum_{i=1}^{N} d(x_i, q)^2 \tag{1}$$

with $d(x_i, q)$ denoting the manifold distance between the ith data point and the mean candidate q.

Each point p of a manifold has a connected linear space called the tangent space and denoted T_pM. The dimension of T_pM is equal to the dimension of the manifold, which, as in the vector space case, specifies the number of degrees of freedom. Vectors in the tangent space are often mapped back to the manifold using the exponential map, Exp_p, which maps straight lines trough the origin of T_pM to geodesics on M passing p.

If we consider the tangent space of an intrinsic mean, $T_\mu M$, we can represent x_i by vectors w_i in $T_\mu M$ such that $\operatorname{Exp}_\mu w_i = x_i$.[3] The map that sends $x_i \in M$

[3] See Figure 1 for an example of a 2-dimensional manifold with sampled elements of the tangent space of the mean and corresponding points on the manifold.

to $w_i \in T_\mu M$ is called the logarithm map and denoted Log_μ. The vector space structure of $T_\mu M$ allows us to use standard statistical tools on $\{w_1, \ldots, w_N\}$. We could for example infer some distribution in $T_\mu M$, sample a vector v from it, and project the result back to a point p on the manifold so that $p = \mathrm{Exp}_\mu v$. It is important to note that consistency is ensured in doing this; p will be on the manifold and hence satisfy the encoded requirements. Turning to the example of hand positions, we have found a consistent way of sampling hand positions without violating the fixed length of the lower arm.

The above procedure can be seen as a way of linearizing the manifold around the intrinsic mean μ because the tangent space $T_\mu M$ provides a first order approximation of the manifold around μ. Yet, distances between vectors in $T_\mu M$ do not always reflect the manifold distances between the corresponding points on the manifold: distances between w_i and the origin of $T_\mu M$ equal the distances $d(x_i, \mu)$, but the inter-point distances $d(x_i, x_j)$ are not in general equal to the tangent space distances$\|w_i - w_j\|$. Accuracy may therefore be lost as a result of the approximation. In short, linearization preserves consistency but may destroy accuracy.

3 Principal Geodesic Analysis

Principal Component Analysis (PCA) is widely used to model the variability of datasets of vector space valued data and provide linear dimensionality reduction. PCA gives a sequence of linear subspaces maximizing the variance of the projection of the data or, equivalently, minimizing the reconstruction errors. The kth subspace is spanned by an orthogonal basis $\{v^1, \ldots, v^k\}$ of principal components v^i.

PCA is dependent on the vector space structure and hence cannot be performed on manifold valued datasets. Principal Geodesic Analysis was developed to overcome this limitation. PGA centers its operations at a point $\mu \in M$ with μ usually being an intrinsic mean of the dataset $\{x_1, \ldots, x_N\}$, and finds geodesic subspaces, which are images $S = \mathrm{Exp}_\mu V$ of linear subspaces V of the tangent space $T_\mu M$. A projection operator π_S is defined by letting $\pi_S(x)$ be a point in S closest to x. The kth geodesic subspace S_k is then given as $\mathrm{Exp}_\mu(V_k)$, $V_k = \mathrm{span}\,\{v^1, \ldots, v^k\}$, where the principal directions v^i are given recursively by

$$\begin{aligned} v^i &= \mathrm{argmax}_{\|v\|=1, v \in V_{i-1}^\perp} \frac{1}{N} \sum_{j=1}^{N} d(\mu, \pi_{S_v}(x_j))^2 \ , \\ S_v &= \mathrm{Exp}_\mu(\mathrm{span}\,(V_{i-1}, v)) \ . \end{aligned} \tag{2}$$

The term being maximized is the sample variance, the expected value of the squared distance to μ. PGA therefore extends PCA by finding geodesic subspaces in which variance is maximized.

Since the projection $\pi_{S_k}(x)$ is hard to compute, PGA is traditionally approximated by linearizing the manifold. The data $x_1, \ldots, x_N$ are projected to $T_\mu M$

using Log_μ, and regular PCA is performed on $w_i = \mathrm{Log}_\mu x_i$. Equation (2) then becomes

$$v^i \approx \mathrm{argmax}_{\|v\|=1, v\in V_{i-1}^\perp} \frac{1}{N} \sum_{j=1}^{N} \left(\langle w_j, v\rangle^2 + \sum_{l=1}^{k-1} \langle w_j, v^l\rangle^2 \right) . \tag{3}$$

We can define a normal distribution $\mathcal{N}$ in $T_\mu M$ using the result of the PCA procedure, and, in doing so, we have performed the procedure described in section 2.2. We will refer to PGA with the approximation as *linearized* PGA. PGA as defined by (2) without the approximation will be referred to as *exact* PGA. Advances in manifold computations allow exact PGA to be computed on the Lie group SO(3) [24] and, more recently, on wide classes of manifolds [2].

Replacing maximization of the sample variances $d(\mu, \pi_{S_v}(x_j))^2$ by minimization of the squared reconstruction errors $d(x_j, \pi_{S_v}(x_j))^2$, we obtain another manifold extension of PCA and thus an alternate definition of PGA:

$$v^i = \mathrm{argmin}_{\|v\|=1, v\in V_{i-1}^\perp} \frac{1}{N} \sum_{j=1}^{N} d(x_j, \pi_{S_v}(x_j))^2 . \tag{4}$$

In contrast to vector space PCA, the two definitions are not equivalent. It can be shown that, in some cases, solutions to (2) will approach parts of the manifold where the cost function is non differentiable, a problem we have not encountered when solving for (4). We are currently working on a paper giving a theoretical treatment of this phenomenon and other differences between the definitions. The latter formulation is chosen for Geodesic PCA to avoid similar instabilities of variance maximization [13]. In correspondence with this, we will use (4) in the rest of the paper, but we stress that this choice is made only to avoid instabilities in (2) and that all computations presented can be performed using the former definition with only minor changes to the optimization algorithms [2].

3.1 Linearized PGA vs. Exact PGA

Computing the projection map π_S is particularly time-intensive causing the computation of exact PGA to last substantially longer than linearized PGA. To give an example, computing linearized PGA for one of the datasets later in this paper takes 5 seconds with a parallelized Matlab implementation, and computing exact PGA for the same example requires approximately 10 minutes. This time penalty makes it is worth considering the actual gain of computing exact PGA. We will in this section give examples of low dimensional manifolds on which it is possible visually to identify the differences between the methods.

We consider surfaces embedded in $\mathbb{R}^3$ and defined by the equation

$$S_c = \{(x, y, z) | cx^2 + y^2 + z^2 = 1\} \tag{5}$$

for different values of the scalar c. For $c > 0$, S_c is an ellipsoid and equal to the sphere $\mathbb{S}^2$ in the case $c = 1$. The surface S_0 is a cylinder and, for $c < 0$, S_c is an

hyperboloid. Consider the point $p = (0, 0, 1)$ and note that $p \in S_c$ for all c. The curvature of S_c at p is equal to c. Note that in particular for the cylinder case the curvature is zero; the cylinder locally has the geometry of the plane $\mathbb{R}^2$ even though it informally seems to curve.

We evenly distribute 20 points along two straight lines through the origin of the tangent space T_pS_c, project the points from T_pS_c to the surface S_c, and perform linearized and exact PGA. Since linearized PCA amounts to Euclidean

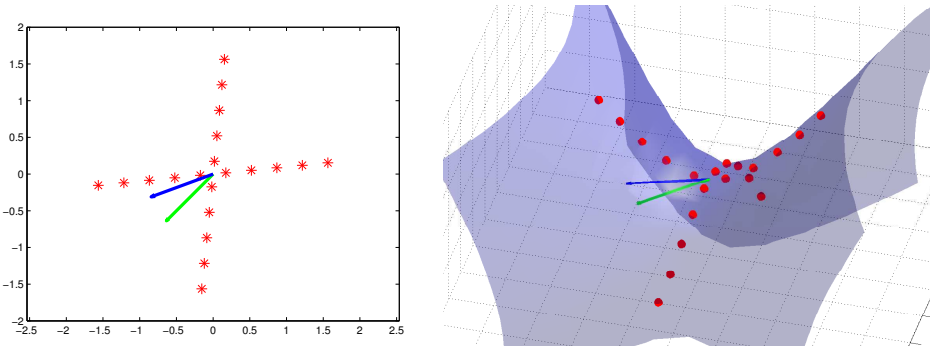


Fig. 1. T_pS_{-2} with sampled points and first principal components (blue exact PGA, green linearized PGA) (left) and S_{-2} with projected points and first principal components (blue exact PGA (2), green linearized PGA) (right).

PCA in T_pS_c, the first principal component divides the angle between the lines for all c. In contrast to this, the corresponding residuals and the first principal component found using exact PGA are dependent on c. Table 1 shows the angle between the principal components found using the different methods, the average squared residuals and differences between squared residuals for different values of c. Let us give a brief explanation of the result. The symmetry of the

Table 1. Differences between methods for selected values of c.

c:	**1**	**0.5**	**0**	**-0.5**	**-1**	**-1.5**	**-2**	**-3**	**-4**	**-5**
angle (°):	0.0	0.1	0.0	3.4	14.9	22.2	24.8	27.2	28.3	28.8
lin. sq. res.:	0.251	0.315	0.405	0.458	0.489	0.508	0.520	0.534	0.539	0.541
exact sq. res.:	0.251	0.315	0.405	0.458	0.478	0.482	0.485	0.489	0.491	0.492
diff (%):	0.0	0.0	0.0	0.1	2.3	5.1	6.7	8.4	8.9	9.0

sphere and the dataset causes the effect of curvature to even out in the spherical case S_1. The cylinder S_0 has local geometry equal to $\mathbb{R}^2$ which causes the equality between the methods in the $c = 0$ case. The hyperboloids with $c < 0$ are non-symmetric causing a decrease in residuals as the first principal component approaches the hyperbolic axis. This effect increases with curvature causing the

the first principal component to align with this axis for large negative values of c.

It is tempting to think that increasing absolute curvature causes increasing differences between the methods. Yet, redoing the experiment with the lines rotated by $\pi/4$ making them symmetric around the x and y axes will produce vanishing differences. Curvature in itself, therefore, does not necessarily imply large differences, and the actual differences are hence dependent on both curvature and the dataset.

3.2 The Difference Indicators

The projection π_S is in (3) approximated using the orthogonal projection in the tangent space $T_\mu M$. We let τ_S denote the difference in residuals arising when using the two projections and aim at approximating τ_S to give an estimate of the gain in precision obtained by using true projections. The subspaces optimizing (4) and (3) will in general differ due to the different projection methods and the fact that residuals are approximated by tangent space distances in (3). We let ρ denote the difference in residuals between the projection of the data to the two subspaces, and we aim at approximating ρ to indicate the gain in accuracy when computing exact PGA.

We start by giving precise definitions for τ_S and ρ before deriving the indicators $\tilde{\tau}_S$ and σ of their values. The term indicators is used to emphasize expected correlation between the values of e.g. τ_S and the indicator $\tilde{\tau}_S$ but with no direct expression for the correlation.

Assume $v_1, \ldots, v_{k-1}$ are principal components and let $v \in T_\mu M$ be such that $v_1, \ldots, v_{k-1}, v$ constitues an orthonormal basis. Let the geodesic subspace S_v be given by $\mathrm{Exp}_\mu \mathrm{span}\{v_1, \ldots, v_{k-1}, v\}$, and let $w_j = \mathrm{Log}_\mu x_j$ for each element of the dataset $\{x_1, \ldots, x_N\}$. We denote by $\hat{\pi}_{S_v}(x_j)$ the point on the manifold corresponding to the orthogonal tangent space projection of w_j, i.e.

$$\hat{\pi}_S(x_j) = \mathrm{Exp}_\mu \left(\langle w_j, v \rangle \, v + \sum_{l=1}^{k-1} \langle w_j, v^l \rangle \, v^l \right) , \tag{6}$$

and define the average projection difference

$$\tau_S = \frac{1}{N} \sum_{j=1}^{N} \left(d(x_j, \hat{\pi}_{S_v}(x_j))^2 - d(x_j, \pi_{S_v}(x_j))^2 \right) . \tag{7}$$

Let now v be an exact PGA principal geodesic component computed using (4) and let $\hat{v}$ be a linearized PGA principal component computed using (3). We let S_v and $S_{\hat{v}}$ denote the geodesic subspaces corresponding to v and $\hat{v}$. The average residual difference is then given by

$$\rho = \frac{1}{N} \sum_{j=1}^{N} \left(d(x_j, \pi_{S_{\hat{v}}}(x_j))^2 - d(x_j, \pi_{S_v}(x_j))^2 \right) . \tag{8}$$

Note that both τ_S and ρ are positive since π_{S_v} minimizes residuals and v minimizes (4).

3.3 The Projection Difference

Since $\pi_{S_v}(x_j)$ is the point in S_v closest to x_j, the differences expressed in each term of (7) measure the difference between $f(\hat{\pi}_{S_v}(x_j))$ and $f(y_j)$ with $y_j \in S_v$ minimizing the map $f(y) = d(x_j, y)^2$. The gradient $\nabla_y f$ vanishes in such a minimum leading us to approximate the difference by the norm of the gradient at $\hat{\pi}_{S_v}(x_j)$. The gradient is readily evaluated since it is given by the component of $-2\mathrm{Log}_{\hat{\pi}_{S_v}(x_j)}(x_j)$ in the tangent space of S_v [11]. We use this to approximate τ_S by

$$\tau_{S_v} \approx \tilde{\tau}_{S_v} = \frac{2}{N} \sum_{j=1}^{N} \|\nabla_{\hat{\pi}_{S_v}(x_j)} f\| \tag{9}$$

and note that each term of the sum, and therefore the entire indicator $\tilde{\tau}_{S_v}$, is inexpensive to compute.

3.4 The Residual Difference

We now heuristically derive an indicator σ that is correlated with ρ. The correlation will be confirmed later by the experiments. Assume for a moment that distances in the tangent space $T_\mu M$ approximate the true manifold distances well. The residual sums $\frac{1}{N}\sum_{j=1}^{N} d(x_j, \pi_{S_{\hat{v}}}(x_j))^2$ and $\frac{1}{N}\sum_{j=1}^{N} d(x_j, \pi_{S_v}(x_j))^2$ will then be close to identical since v is chosen to minimize the latter sum, and $\hat{v}$ is chosen to minimize the sum of tangent space residuals. The difference ρ will therefore be close to zero. Conversely, assume that distances in the tangent space differ greatly from the true manifold distances. On constant curvature spaces like the sphere S_1, these distance differences will generally be uniformly distributed causing the linearized principal component $\hat{v}$ to be close to v and ρ therefore close to zero. On the contrary, the distance differences will vary on spaces with non-constant curvature like S_{-1} where $\hat{v}$ in general is far from v causing ρ to be large. We therefore expect ρ to be correlated with the standard deviation σ of the differences between the tangent space residual approximations and the actual orthogonal projection residuals,

$$\sigma = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(\|w_j - \mathrm{Log}_\mu(\hat{\pi}_{S_{\hat{v}}})\| - d(x_j, \hat{\pi}_{S_{\hat{v}}}(x_j)) - \mu \right)^2}, \tag{10}$$

with μ the mean value of the scalars $\|w_j - \mathrm{Log}_\mu(\hat{\pi}_{S_{\hat{v}}})\| - d(x_j, \hat{\pi}_{S_{\hat{v}}}(x_j))$. We use σ, which again is fast to compute, to indicate the size of ρ.

4 Experiments

We present experiments on the synthetic data of section 3.1 and on two real-life datasets for two purposes: the experiments will show examples where computing exact PGA results in increased accuracy as well as examples where linearized

PGA performs well, and the power of the indicators developed in section 3 will be explored.

When investigating the correlation between the indicator $\tilde{\tau}_{S_{\hat{v}}}$ and the projection difference $\tau_{S_{\hat{v}}}$, we let $\hat{v}$ be the first principal component computed using linearized PGA. In addition, we compare the residual difference ρ with the indicator σ.

4.1 Synthetic Data

We test the indicators on the manifolds S_c with the synthetic data described in section 3.1. Figure 2 shows τ_S as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ and ρ as a function of the indicator σ for each value of c. For both graphs, we see correlation

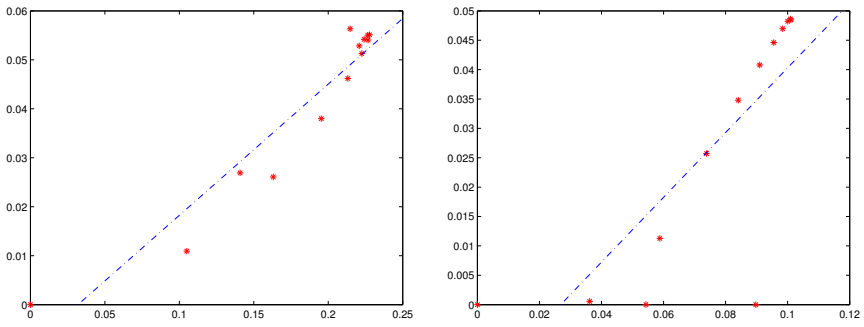


Fig. 2. Synthethic data: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ with the broken line fitted to the points (left) and residual difference ρ as a function of the indicator σ with the broken line fitted to the points (right).

between the indicators and actual differences. For $c = 1$ and $c = 0.5$, σ is relatively high compared to ρ stressing that the indicators only give approximations and that, if full precision is required, exact PGA should be computed.

4.2 Vertebrae Outlines

In this experiment, we consider outlines of vertebrae obtained in a study of vertebral fractures. The dataset of 36 lateral X-rays have been manually annotated by medical experts to identify the outline of the vertebra of each image. To remove variability in the number and placement of points, a resampling is performed to ensured constant inter-point distances. With this equidistance property in mind, the authors in [20] define a submanifold of $\mathbb{R}^{2n}$ on which the outlines naturally reside. We give a brief review of the setup but refer to the paper for details. The equidistance constraint is encoded using a map $F : \mathbb{R}^{2n} \to \mathbb{R}^{n-2}$ with components

$$F^i(P_1, ..., P_n) = d_{i+2,i+1} - d_{i+1,i}, \quad i = 1, .., n-2 \tag{11}$$

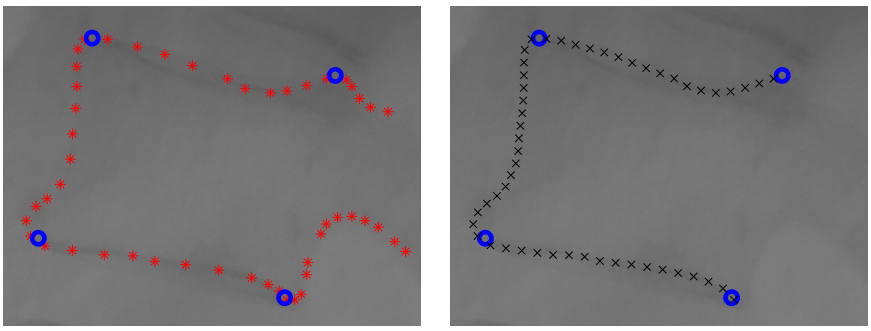


Fig. 3. Manually annotated vertebrae outline (left) and resampled outline (right).

with n the number of points and $d_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2$ the squared distances between points P_i and P_j. The constraint is satisfied for a vertebra outline $c = \{P_1, \ldots, P_n\}$ if $F(c) = 0$. An additional constraint is added to remove scaling effects by ensuring the outline reside on the unit sphere. The preimage $A_n = F^{-1}(0)$ is then a submanifold of $\mathbb{R}^{2n}$, the space of equidistant vertebra outlines. We choose 8 random outlines from the dataset and perform

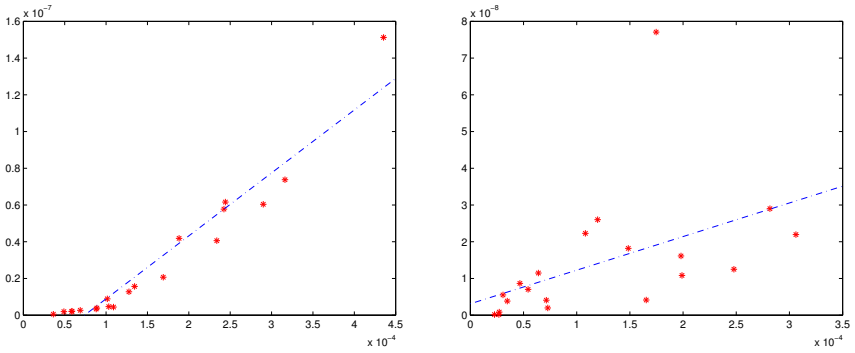


Fig. 4. Vertebrae outlines: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ (left) and residual difference ρ as a function of the indicator σ (right).

linearized PGA and exact PGA. The experiment consists of 20 such selections, and, for each selection, the entities $\tau_{S_{\hat{v}}}$, $\tilde{\tau}_{S_{\hat{v}}}$, ρ and σ are computed and plotted in Figure 4. Though we visually see correlation between the indicators and their respective associated values in the figures, not only are the correlations low, as the indicators and their values have significantly different orders of magnitude, but in reality, both the indicators and the associated values are in the order of the computation tolerance, i.e close to zero from a numerical point of view. As small indicators should imply small values, we can conclude that the indicators

works as required and that, for the example of vertebra outlines, doing statistics on the manifold A_n is helpful in keeping the data consistent, i.e. the equidistance constraint satisfied, but provides little added accuracy.

4.3 Human Poses

In this experiment, we consider human poses obtained using tracking software. A consumer stereo camera[4] is placed in front of a test person, and the tracking software described in [10] is invoked in order to track the pose of the persons upper body. The recorded poses are represented by the human body end-effectors; the end-points of each bone of the skeleton. The placement of each end-effector is given spatial coordinates so that an entire pose with k end-effectors can be considered a point in $\mathbb{R}^{3k}$. To simplify the representation, only the end-effectors of a subset of the skeleton are included, and, when two bones meet at a joint, their end-points are considered one end-effector. Figure 5 shows a human pose with 11 end-effectors marked by thick dots.

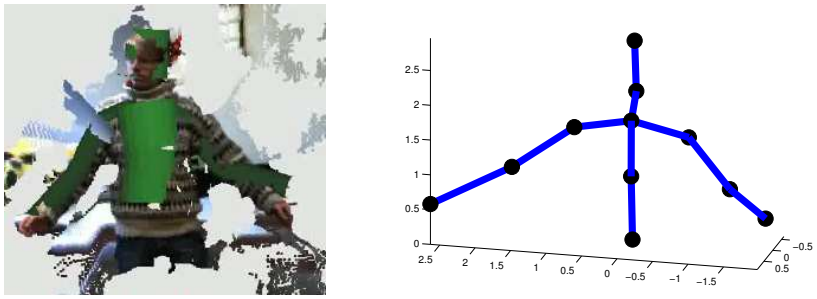


Fig. 5. Camera output superimposed with tracking result (left) and a tracked pose with 11 end-effectors marked by thick dots (right).

The fact that bones do not change length in short time spans gives rise to a constraint for each bone; the distance between the pair of end-effectors must be constant. We incorporate this into a pose model with b bones by restricting the allowed poses to the preimage $F^{-1}(0)$ of the map $F : \mathbb{R}^{3k} \to \mathbb{R}^b$ given by

$$F^i(x) = \|e_{i_1} - e_{i_2}\|^2 - l_i^2 \ , \tag{12}$$

where e_{i_1} and e_{i_2} denote the spatial coordinates of the end-effectors and l_i the constant length of the ith bone. In this way, the set of allowed poses constitute a $3k - b$-dimensional implicitly represented manifold.

We record 26 poses using the tracking setup, and, amongst those, we make 20 random choices of 8 poses and perform linearized PGA and exact PGA. For each experiment, $\tau_{S_{\hat{v}}}$, $\tilde{\tau}_{S_{\hat{v}}}$, ρ, and σ are computed and plotted in Figure 6. The

[4] `http://www.ptgrey.com/products/bumblebee2/`

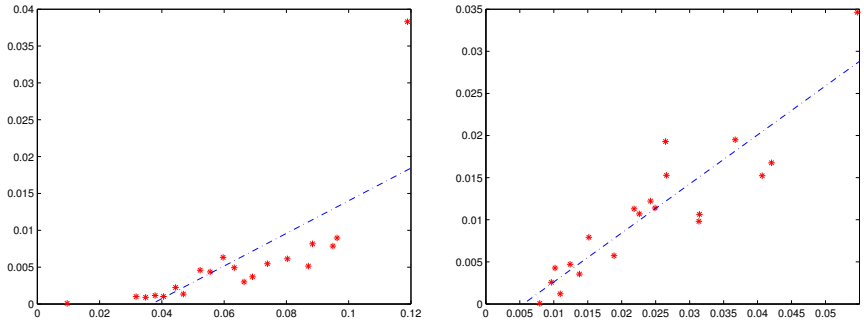


Fig. 6. Human poses: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ (left) and residual difference ρ as a function of the indicator σ (right).

indicators provide a good picture of the projection and residual differences, which are significantly greater than for the vertebra experiment. The indicators and the corresponding true values are now at the same order of magnitude, and the correlation between the indicators and the values they correspond to is therefore significant. The maximal increase in average squared residuals is 1.53 percent with individual squared point residuals changing up to 30.7 percent.

5 Conclusion

In this paper, we have explored the differences between exact PGA and its widely used simplification, linearized PGA. We have developed simple indicators of the loss of accuracy when using the linearized PGA instead of exact PGA. As shown on real-life examples of manifold valued datasets, these indicators provide meaningful insight into the accuracy of the linearized method. The experiments, in addition, show that linearization is in some cases a good and fast approximation, but exact PGA offers better accuracy for other applications.

We are currently working on deriving formal arguments for the correlation between σ and ρ. In the future, we plan to apply the developed indicators to the many uses of PGA, which have previously been computed using the linearized approach, to test whether exact PGA can provide significant increases in accuracy and hence more precise modeling. In order to make better decisions on whether to use linearized or exact PGA, it will be useful to find thresholds for the values of $\tilde{\tau}_{S_{\hat{v}}}$ and σ dependent on the sought for precision. Future research will hopefully lead to such thresholds.

References

1. Fletcher, P., Lu, C., Pizer, S., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. Medical Imaging, IEEE Transactions on **23** (2004) 995–1005

2. Sommer, S., Lauze, F., Nielsen, M.: The differential of the exponential map, jacobi fields, and exact principal geodesic analysis. Submitted. (2010)
3. Fletcher, P.T., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing **87** (2007) 250–262
4. Fletcher, P.T., Joshi, S.: Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. ECCV Workshops CVAMIA and MMBIA. **3117** (2004) 87—98
5. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. J. Comput. Vision **66** (2006) 41–66
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision **22** (1995) 61—79
7. Pennec, X., Guttmann, C., Thirion, J.: Feature-based registration of medical images: Estimation and validation of the pose accuracy. In: MICCAI 1998. Springer Berlin (1998) 1107–1114
8. Kendall, D.G.: Shape manifolds, procrustean metrics, and complex projective spaces. Bull. London Math. Soc. **16** (1984) 81–121
9. Sminchisescu, C., Jepson, A.: Generative modeling for continuous Non-Linearly embedded visual inference. In ICML (2004) 759—766
10. Hauberg, S., Sommer, S., Pedersen, K.S.: Gaussian-like spatial priors for articulated tracking. In: Computer Vision - ECCV 2010, Heraklion, Greece (2010)
11. Karcher, H.: Riemannian center of mass and mollifier smoothing. Communications on Pure and Applied Mathematics **30** (1977) 509–541
12. Pennec, X.: Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. J. Math. Imaging Vis. **25** (2006) 127–154
13. Huckemann, S., Hotz, T., Munk, A.: Intrinsic shape analysis: Geodesic PCA for riemannian manifolds modulo isometric lie group actions. Statistica Sinica **20** (2010) 1–100
14. do Carmo, M.P.: Riemannian geometry. Mathematics: Theory & Applications. Birkhauser Boston Inc., Boston, MA (1992)
15. Lee, J.M.: Riemannian manifolds. Volume 176 of Graduate Texts in Mathematics. Springer-Verlag, New York (1997) An introduction to curvature.
16. Dedieu, J., Nowicki, D.: Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds. Journal of Complexity **21** (2005) 487–501
17. Noakes, L.: A global algorithm for geodesics. Journal of the Australian Mathematical Society **64** (1998) 37–50
18. Klassen, E., Srivastava, A.: Geodesics between 3D closed curves using Path-Straightening. In: ECCV 2006. Volume 3951. Springer (2006) 95–106
19. Schmidt, F., Clausen, M., Cremers, D.: Shape matching by variational computation of geodesics on a manifold. In: Pattern Recognition. Springer Berlin (2006) 142–151
20. Sommer, S., Tatu, A., Chen, C., Jørgensen, D., de Bruijne, M., Loog, M., Nielsen, M., Lauze, F.: Bicycle chain shape models. MMBIA/CVPR 2009 (2009) 157–163
21. Huckemann, S., Ziezold, H.: Principal component analysis for riemannian manifolds, with an application to triangular shape spaces. Advances in Applied Probability **38** (2006) 299–319
22. Fletcher, P., Lu, C., Joshi, S.: Statistics of shape via principal geodesic analysis on lie groups. In: CVPR 2003. Volume 1. (2003) I–95–I–101 vol.1
23. Wu, J., Smith, W., Hancock, E.: Weighted principal geodesic analysis for facial gender classification. In: Progress in Pattern Recognition, Image Analysis and Applications. Springer Berlin (2008) 331–339
24. Said, S., Courty, N., Bihan, N.L., Sangwine, S.: Exact principal geodesic analysis for data on so(3). EUSIPCO 2007 (2007)

7.

Paper #6:
Bicycle Chain Shape Models

Peer-reviewed conference paper presented at the Mathematical Methods in Biomedical Image Analysis (MMBIA) workshop at CVPR 2009, Miami Beach, Florida, 2009.

Authors:

Stefan Sommer, Aditya Tatu, Chen Chen, Dan R. Jørgensen, Marleen de Bruijne, Marco Loog, Mads Nielsen, François Lauze

Notes:

To reduce annotation variation in point based models, we introduce the *bicycle chain shape model* for landmark based representation of 2D shapes. By constraining the pairwise distances between consecutive landmarks, we obtain non-linear shape manifold. We show how the Exponential and logarithm maps can be computed and use this to perform Principal Geodesic Analysis. The dimension reduction is evaluated with the PGA algorithm on a dataset of manually annotated outlines of human vertebrae X-rays.

Bicycle Chain Shape Models

Stefan Sommer [1,2], Aditya Tatu [2], Chen Chen [2], Dan R. Jørgensen [2,3], Marleen de Bruijne [2,4], Marco Loog [2,5], Mads Nielsen [2,3], François Lauze [2]

Abstract

In this paper we introduce landmark-based preshapes which allow mixing of anatomical landmarks and pseudo-landmarks, constraining consecutive pseudo-landmarks to satisfy planar equidistance relations. This defines naturally a structure of Riemannian manifold on these preshapes, with a natural action of the group of planar rotations. Orbits define the shapes. We develop a Geodesic Generalized Procrustes Analysis procedure for a sample set on such a preshape spaces and use it to compute Principal Geodesic Analysis. We demonstrate it on an elementary synthetic example as well on a dataset of manually annotated vertebra shapes from X-ray. We re-landmark them consistently and show that PGA captures the variability of the dataset better than its linear counterpart, PCA.

1. Introduction

There is a wide literature on shape representation and shape analysis in Computer Vision and Medical Imaging as shape understanding is one of the most fundamental task in Image Analysis. A 2-dimensional shape is generally defined as an equivalence class of smooth 1-dimensional submanifolds of $\mathbb{R}^2$ modulo similarity [13]. Computational representations, ranging from the simplest to the most sophisticated, have been suggested in the past, e.g. point set distributions [9, 1], linear point distribution models (PDM) [4], parametric representations via B-splines, levelset representations [16], and their adaptation, as for example, specific shape constraints, soft priors, *etc*..., for an ever growing amount of tasks.

Manual annotations of anatomical structures in medical images, such as X-rays, Ultra Sound, are routinely performed by radiologists and other experts in many clinical studies, resulting in the encoding of shapes as point set distributions. Point set distributions for shape representation and analysis are of tremendous importance in Medical Imaging. Deriving such distributions presupposes consistent annotations, which is not always the case: the following figure shows two annotated vertebra shapes from X-ray images, during a clinical study on vertebra fractures, the first vertebra is annotated with 31 points, the second with 32. Moreover the number of points between corner landmarks (the circular ones) do not match for corresponding pairs. This is caused by the absence

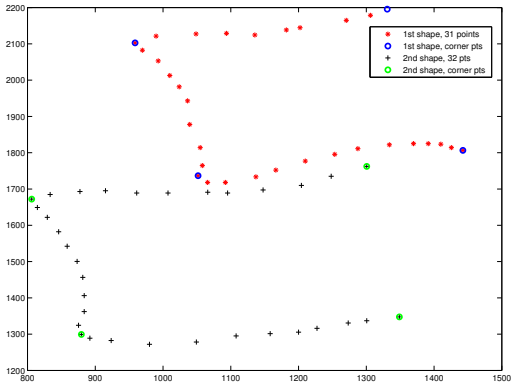


Figure 1. Two annotated vertebrae from a clinical study. The number of annotation points differ.

of clear ground truth landmarks along the endplates of the vertebrae. In order to tackle this somehow common situation, a resampling is necessary; pseudo-landmarks should be placed such that the resulting model is more compact, no additional variation caused by points sliding along the outline should be modelled. Some recent approaches for curves and surfaces were proposed for instance by Davies *et al.* [5] using minimum description length to solve this problem, while, for surfaces, Cates *et al.* used an entropy based particle system approach in [3].

For curves, which are the objects of interest in

[1]Corresponding author email: sommer@diku.dk
[2]Department of Computer Science, University of Copenhagen, Denmark
[3]Nordic Bioscience A/S, Herlev, Denmark
[4]Erasmus MC, Rotterdam, The Netherlands
[5]ICT Group, Delft University of Technology, The Netherlands

this paper, a way to do it is to first impose a fixed number of pseudo-landmarks between landmarks, regularly distributed along the outline between the landmarks. This regularity often takes the form of an equidistance constraint for pseudo-landmarks situated between consecutive landmarks. This can be formulated as setting the variance of the distribution of of planar distances (or square-distances) between consecutive pseudo-landmarks to 0. In a figurative way, a segment between two consecutive landmarks is similar to a segment of a **bicycle chain**, for the links that constitute a bicycle chain have the same length! This has the nice property of minimizing the variability due to the annotation process. But once this resampling has been performed, forgetting this variance constraint induces apparent extra variability which may be difficult to handle due to the non linearity of the constraint. This is illustrated in Figure 2 where the Euclidean mean of the upper and lower curves does not have equidistant pseudo-landmarks introducing extra variability on the horizontal placement of the pseudo-landmarks. We

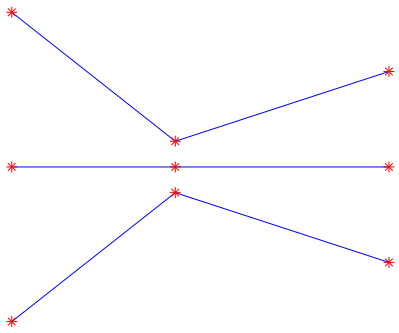


Figure 2. Two 3-point curves and the Euclidean mean.

propose to handle this situation by introducing the constraint explicitly in the descriptions of our preshape spaces. This null-variance can be reformulated as a series of simple quadratic constraints on the pseudo-landmarks and will, for shapes determined by n points in $\mathbb{R}^d$, define implicitly a submanifold of the point set spaces $(\mathbb{R}^d)^n$. Endowed with the metric that comes from the standard Euclidean structure of $(\mathbb{R}^d)^n$, it becomes a Riemannian manifold. In order to obtain point distributions models, Generalized Procrustes Analysis (GPA) [8] should be performed with the induced metric, leading to what we will call Geodesic Generalized Procrustes Analysis (GGPA), while Principal Component Analysis should be replaced by Principal Geodesic Analysis (PGA) [7] in order to take into account the curved structure of the manifold. In the rest of this paper, we will focus to point set configurations in $\mathbb{R}^2$. This will simplify the presentation. Extension to 3D curves can be carried out easily.

So as to be able to compute GGPA and PGA, we need tools for computing Riemannian exponential map, geodesics, and log map on implicitly defined submanifolds. By extending computations of exponential map to provide not only geodesic, but corresponding moving frames, we propose a shooting method for computing Log maps. When it fails, we replace it by a path straightening algorithm based on local properties of geodesics.

This paper is organized as follows. In the next section we introduce the preshape manifolds that we use as well as the geometric tools needed for our statistical analysis: Geodesic Generalized Procrustes analysis and Principal Geodesic Analysis. Exponential and Log maps are discussed in Section 3. In Section 4 we present experiments; the first one on the 3-points toy example and the second on a data set of vertebra coming from a clinical study on vertebra fractures. Finally we conclude in Section 5.

2. Preshape manifolds

In point based models, a typical object consists of q landmark points and n_k, $k = 1, ..., q-1$ ($k = 1, ..., q$, for closed configurations) pseudo landmarks between consecutive landmark points. A segment of this object consists of $n_k + 2$ points, n_k pseudo-landmarks $P_i, i = 2, ..., n_k + 1$ between 2 landmark endpoints P_1, P_{n_k+2}. The objects we consider consist of such configurations with equal (squared) Euclidean distance between the neighboring points in each of the segments. This characteristic distance will generally vary from segment to segment and objects to objects, even when the sequence of numbers $(q, n_1, \ldots, n_{q-1})$ is fixed. We start by describing constraints on segments.

2.1. n-Links Bicycle Chain Manifolds

Here onwards we work on one segment with $n_k = n-2$ pseudo-landmarks between 2 landmark endpoints. Then the equidistant constraint can be written as a simple quadratic constraint $F : \mathbb{R}^{2n} \to \mathbb{R}^{n-2}$ given as

$$F_i(P_1, ..., P_n) = d_{i+2,i+1} - d_{i+1,i}, \quad i = 1, .., n-2 \quad (1)$$

where $d_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2$ is the squared euclidean distance between points P_i and P_j, The configuration space is the subspace of $\mathbb{R}^{2n}$ given by $A_n = F^{-1}(0) \backslash \Delta$, where Δ is the "diagonal" $\Delta = (P, \ldots, P) \in (\mathbb{R}^2)^n$ consisting of segments reduced to a single point, for, while $\Delta \subset F^{-1}(0)$, the rank of F breaks down exactly along Δ. This ensures that A_n is

a submanifold of $(\mathbb{R}^2)^n = \mathbb{R}^{2n}$ [2] The tangent space of A_n at a segment x is given by

$$T_x A_n = \ker(JF(x))$$

the kernel (or null space) of the Jacobian of F at point $x \in A_n \subset \mathbb{R}^{2n}$. By restricting the scalar product of $\mathbb{R}^{2n}$ to $T_x A_n$, A_n is endowed with a Riemannian Metric [2]. We may call A_n a **n-links bicycle chain segment manifold**.

More general point configurations are then built by concatenating these n-links bicycle chain segments, imposing endpoint matching which are linear constraints. When the number q of landmarks points and the numbers n_k, $k = 1, \ldots, q-$ of pseudo-landmarks points are fixed, corresponding configurations form a Riemannian submanifold of the product manifold $\prod_{i=1}^{q} A_{n_k+2}$, and this manifold has also the metric inherited from the embedding space $(\mathbb{R}^2)^N$ with $N = q + \sum_{k=1}^{q-1} n_k$.

Having a Riemannian metric, we can compute length of paths in these manifolds, define geodesic and geodesic distances [2].

2.2. Removing Translation and Scaling

In the following, we denote by $\mathcal{M}$ such a configuration manifold. To work with preshapes in the sense of [9], we need to quotient out translations and scaling from points in $\mathcal{M}$ (although in some applications, scale could be an important feature of the shape). Removing translations is as usual easy. If $\mathcal{M}'$ denotes the submanifold of $\mathcal{M}$ of configurations with centroid at the origin of $\mathbb{R}^2$ then $\mathcal{M} \simeq \mathcal{M}' \times \mathbb{R}^2$, by sending a configuration $S = (S_1, \ldots, S_n)$ to $(S - \bar{S}, \bar{S}$ where $\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i$ is the centroid of S. This decomposes $\mathcal{M}$ into two *orthogonal* factors, which imply that a geodesic path in $\mathcal{M}$ between centered objects in $\mathcal{M}'$ will be in fact a geodesic path in $\mathcal{M}'$. From now on we therefore assume that all our configurations have centroid at $0 \in \mathbb{R}^2$. Following [9], we remove scale by imposing $\|S\|^2 = \sum_{i=1}^{n} \|S_i\|^2 = 1$, i.e by intersecting $\mathcal{M}'$ with the unit sphere of the embedding space. This defines a new submanifold $\mathcal{S}$ of $\mathcal{M}'$, and $\mathcal{S}$ is our preshape manifold.

2.3. Geodesic Generalized Procrustes Analysis

Given a sample set $(S_i)_{i=1\ldots n} \in \mathcal{S}$, our GPA follows [9], but is performed on $\mathcal{S}$. It attempts to compute a set of planar rotations $R_{\bar{\theta}_i}$, $i = 1, \ldots, n$ and a preshape $\bar{\mu} \in \mathcal{S}$ minimizing the misalignment criterion

$$E(\theta_1, \ldots, \theta_n, \mu) = \sum_{i=1}^{n} d(R_{\theta_i} S_i, \mu)^2. \qquad (2)$$

where d is the geodesic distance in $\mathcal{S}$. This will result in an aligned preshape sample $(\bar{S}_i := R_{\bar{\theta}_i} S_i)_{i=1\ldots n}$, $\bar{\mu}$ being the Fréchet mean ([10]) of this sample and the distances $d(\bar{S}_i, \bar{\mu})$ should represent the true *shape* distances to this mean.

The minimization procedure for (2) is sketched in Algorithm 1. We describe briefly the loop steps. A first guess for the rotations is computed by standard Euclidean rigid registration [8] providing candidate rotation angles for each shape. Then we search for the rotations angles that minimize the true geodesic distances in a neighborhood of the previously obtained angles. The Fréchet mean is then computed by adapting the procedure described in [7] to our case. In Figure 3 the need for the minimization search after the initial Euclidean registration is illustrated by showing a base preshape, and rotation of a second preshape with respect to Euclidean and submanifold distances.

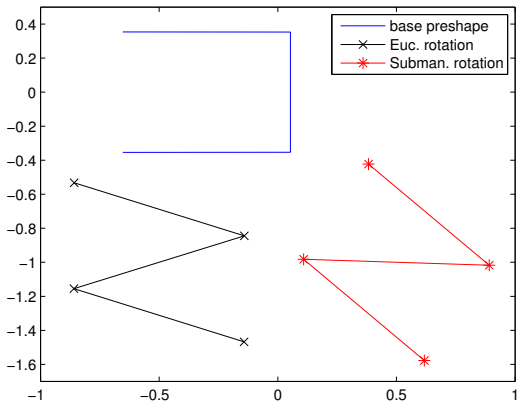


Figure 3. A base preshape and Euclidean and manifold registration.

Algorithm 1 Calculate the mean shape μ' and the aligned shapes S_i''

Require: $S_i \in \mathcal{S}$, $i = 1, ..., m$
$\mu' = S_1$ {initial guess}
repeat
Set $\mu = \mu'$. $S_i' = S_i$ shapes aligned to μ using Euclidean distances.
$S_i'' = S_i'$ shapes aligned to μ using *geodesic* distances.
$\mu' =$ Fréchet mean of (S_i'')
until $d(\mu, \mu') <$ Threshold.
Output: Mean μ', aligned shapes S_i''.

2.4. Principle Geodesic Analysis (PGA)

PGA is a generalization of Principal Component Analysis (PCA) to nonlinear manifolds [7]. We seek to compute a minimum number of tangent vectors at the mean, which generate geodesics that represent as much variability in the data on the manifold as possible. Thus PGA is PCA done on the tangent space of the mean. Unfolding the manifold to this tangent space is performed by the Riemannian Log map. The algorithm can be summarized as:

- Given m preshapes in $\mathcal{S}$ and the mean preshape μ, compute $v_i = Log_\mu(S_i)$, $i = 1, ..., m$, the tangent vectors for each preshape in the tangent space at the mean.
- Compute the covariance matrix $C = \frac{1}{m}\sum_{i=1}^{m} v_i v_i^T$
- Compute the eigenvectors and eigenvalues (e_i, λ_i) of C.

The geodesic paths corresponding to the tangent vectors $e_i \in T_\mu\mathcal{S}$ are the principle geodesic components.

Computing the Fréchet mean and PGA use Exponential map, Log map and geodesics on implicitly defined Riemannian manifolds. They are described in the next section

3. Geodesics on the manifold; the Exp- and Log-map

Geodesics are fundamental to the theory of Riemannian manifolds ([2]). They are closely related to the Exponential map $\mathrm{Exp} : TM \to M$ in the sense that a geodesic γ trough the point p with initial velocity vector v is given by the curve

$$\gamma(t) = \mathrm{Exp}_p tv \ .$$

The map Exp_p is invertible in a sufficiently small neighborhood of 0 in T_pM. When U is such a neighborhood we denote by $\mathrm{Log}_p : \mathrm{Exp}_p(U) \to U$ the inverse of Exp_p.

The distance between two elements of the manifold is given by

$$d_M(p,q) = \inf\left\{l(c) \middle| c \text{ is a curve joining } p \text{ and } q\right\} \ .$$

Here $l(c)$ denotes the length of the curve c. Since geodesics are critical points of the length functional, it is in the case of a complete manifold M sufficient to consider geodesics when computing the distance. Therefore, if p and q are sufficiently close so that only one geodesic joins them,

$$d_M(p,q) = \|\mathrm{Log}_p q\| \ . \tag{3}$$

In general we cannot be sure that a given geodesic joining p and q is length minimizing. In such cases, we define $\mathrm{Log}_p q$ to be the initial direction of *some* geodesic joining p and q and use (3) as a guess on the distance.

Computing Exp_p amounts to solving an initial value ODE problem. This can be done neatly numerically, confer [6]. Computing $\mathrm{Log}_p q$ is substantially harder. We make use of a shooting method ([12], [14]) for computation of Log_p for input values close to p, and a path-straightening method for non-local input.

3.1. Shooting method

A shooting method iteratively improves an initial guess by repeatedly computing a residue or error correction, and updates the initial guess using that. Based on the fact that Log_p is the inverse of Exp_p, our basic algorithm is presented in Algorithm 2. The ability to compute length and direction in Euclidean space and the implicit representation of $\mathcal{S}$ as a submanifold of Euclidean space enables us to compute both the initial guess, update v, and compute the Euclidean error of our guess. When q is close to p these estimates work well and improve the situation in [14] where the embedding space approximations are not at hand and e.g. the update of v therefore is based on numerical estimates of the gradient of a cost functional.

Algorithm 2 Calculate $v = \mathrm{Log}_p q$ on $\mathcal{S}$ by shooting

Require: $p, q \in \mathcal{S}$
  $v \Leftarrow$ projection of $q - p$ to $T_p\mathcal{S}$ {initial guess}
  repeat
    $\tilde{q} \Leftarrow \mathrm{Exp}_p v$ {shot based on guess}
    $\tilde{r} \Leftarrow$ projection of $q - \tilde{q}$ to $T_{\tilde{q}}\mathcal{S}$ {residue at $\tilde{q}$}
    $r \Leftarrow$ par. transport of $\tilde{r}$ to $T_p\mathcal{S}$ {residue at p}
    $v \Leftarrow v + r$ {update v}
  until $\|\tilde{q} - q\|_{\mathbb{R}^{2n}}$ is sufficiently small.

We use the projection of the vector $q - p$ in embedding Euclidean space to the tangent space $T_p\mathcal{S}$ as our initial guess. In each iteration we compute $\mathrm{Exp}_p v$ and express the error by the Euclidean distance $\|q - \mathrm{Exp}_p v\|$. We update v by projecting the Euclidean residue $q - \mathrm{Exp}_p v$ onto the tangent space $T_{\mathrm{Exp}_p v}\mathcal{S}$, parallel transport the resulting vector to $T_p\mathcal{S}$ and add it to v; this procedure is the natural manifold generalization of error correction in Euclidean space.

The parallel transport is computed using a parallel frame along the curve $t \mapsto \mathrm{Exp}_p tv$. We compute the parallel frame by using the fact that parallel vector fields have zero intrinsic acceleration, introduce a Lagrange multiplier, and solve the resulting ODE. The computation of the frame can be nicely coupled with

the computation of $\mathrm{Exp}_p v$ when using the method of [6].

The shooting method relies completely on the quality of the initial guess and updating residues. Both are determined by how well the projections on the tangent spaces approximate the paths on the manifold, or, in other words, how close to linear the manifold is; in an Euclidean manifold the shooting method converges in one iteration whether as it on a torus might not converge at all. It will though always converge locally due to the smoothness of our manifold.

An additional drawback of the shooting method is its sensitivity to numerical errors in the computation of Exp_p. This can especially be a problem if the curvature around the target element q is large, confer [11].

3.2. Path straightening

When the shooting method fails to converge due to large curvature of the manifold, we apply the path straightening method of [15]; we update an initial curve by repeatedly shooting between pairs of points on the initial curve close to each other. The closeness assures the convergence of the shooting method. In each iteration the curve is a piecewise geodesic and by repeatedly changing the points between which we shoot, the non-smooth bends of the curve are removed. Since geodesics are critical points of the length functional, we stop the process when we get no signification reduction of length on each iteration.

Path straightening requires an initial path. In practice we get this path by shooting until we detect non-convergence of the shooting method. We then restart the shooting method with the best guess from the previous run as our new starting point. In practice we always obtain convergence of the shooting method in the second run. Now concatenating the geodesics obtained from the two runs gives a piecewise geodesic connecting the points which can serve as input to the path straightening algorithm. In case this method fails, we explicitly make an initial path.

As noted in [15] we may need to extract a subsequence in order for the path straightening algorithm to convergence to a geodesic. In practice we do not experience such situations, and we accept the possibility of this happening in the same way as we accept that geodesics might not be length minimizing.

4. Experiments

We present two examples illustrating the effect of our manifold setting. We start by discussing the dimensionality reduction gained in a small 3-point example and then progress to study a dataset of vertebrae shapes.

4.1. Illustrative example

In Figure 4 we see three 3-point preshapes with equidistant points. They are all normalized and hence reside on the manifold $\mathcal{S}$. The middle preshape is the Fréchet mean of the upper and lower preshapes, and hence the mean of all three preshapes.

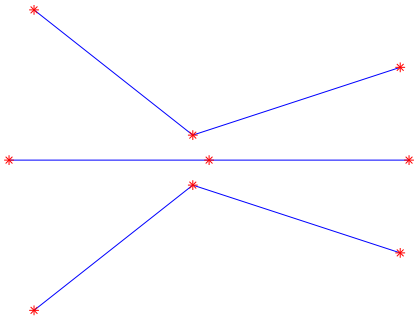


Figure 4. Three 3-point preshapes on the manifold.

The manifold $\mathcal{S}$ has two dimensions. Doing a Principal Geodesic Analysis on the set of the three preshapes, we get one mode of variation. The geodesic corresponding to this mode connects the three preshapes as illustrated in Figure 5. Note that in the figure the preshapes have been placed in the plot as to have zero mean.

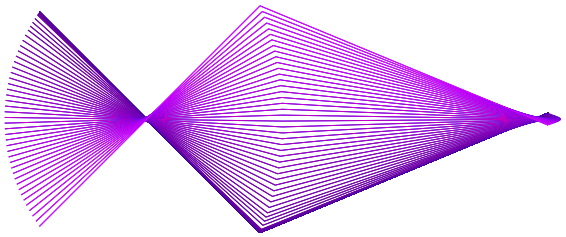


Figure 5. The geodesic corresponding to the only mode of variation obtained from PGA.

Now suppose we disregard our manifold notion and attempt to do Euclidean Principal Component Analysis in the embedding Euclidean space. The Euclidean mean of the three preshapes will again be a straight line, but in this situation the points on the mean will not be equally spaced and hence the mean will not be in $\mathcal{S}$. When computing the PCA we get two modes of variation; one mode representing vertical motion as illustrated in Figure 6, and one mode representing horizontal motion. The latter mode arises from the placement of the points on the straight line mean and is

thus irrelevant. Therefore, in this example the PCA captures only 97.5 percent of the variation in a mode giving relevant information. This contrasts that PGA captures all variation.

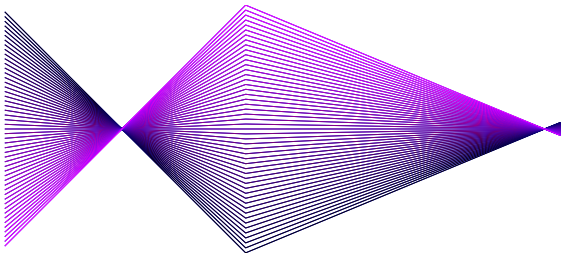


Figure 6. One of two modes of variation for PCA.

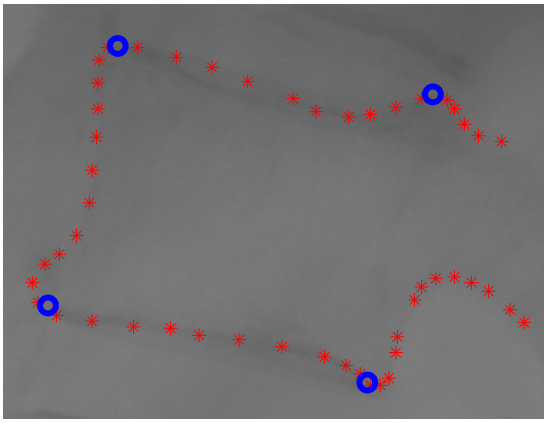


Figure 7. Manually annotated vertebra.

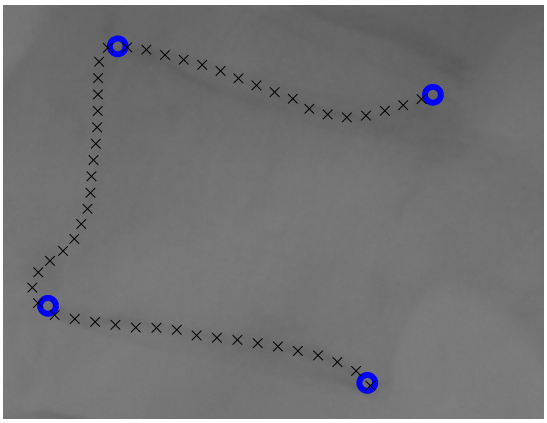


Figure 8. Result of applying redistancing procedure.

4.2. Vertebra shapes

Our dataset of vertebrae consists of 304 manually annotated vertebra shapes on lateral X-rays. For each vertebra, outlines have been manually drawn by choosing points along the contours, assuming a simple linear interpolation between them. Corner points of the vertebra endplates are indicated but do not always match the outlines perfectly. New corner points have been defined as the points of the contour that are closest to the manually annotated corners. This divides the outline into 3 segments, the upper, left and lower ones. For resampling, we fixed the number of pseudo-landmarks per segment to 16, leading to 52 points per shape. The pseudo-landmarks positions were computed segment-wise so as to minimize a squared-distance between the original outline and the new one. Given an n-tuple $\mathcal{P} = (P_1, \ldots, P_n)$ of equidistant-spaced points, with P_1 and P_n being the *fixed* corner points of this segment, let $C_{\mathcal{P}}(t)$ be the piecewise linear curve joining them, and $C_0(t)$ the piecewise linear curve formed by joining the original annotated points for the corresponding segment. We minimize the squared-distance

$$E(\mathcal{P}) = \int (C_{\mathcal{P}} - C_o)^2 \, dt.$$

We start with a configuration $\mathcal{P}$ on the straigth line segment joining P_1 to P_n and perform gradient descent on the corresponding preshape manifold $\mathcal{S}$ using the exponential map. The result of applying the redistancing procedure to the manually annotated vertebra in Figure 7 is shown in Figure 8.

In our illustrative example it is clear that we introduce non-linearity when restricting to the manifold. In order to illustrate that we have significant curvature also in the relatively high dimensional manifold used for the vertebrae, we compute the Fréchet mean vertebra and measure an approximate distance from each vertebra to the tangent space of the mean; we let v_m denote the mean and for each vertebra v we compute $w = \mathrm{Log}_{v_m} v$. We then let x be the distance $\|v - (v_m + w)\|$ between the vertebra and an approximated projection to $T_{v_m}\mathcal{S}$, and record the relative distance $x/\|w\|$. A non-curved manifold would result in zero relative distance. We see a mean relative distance of 12 percent clearly indicating that the manifold is curved. Performing the same computation on the non-normalized manifold $\mathcal{M}'$ gives a mean relative distance of 9 percent indicating that not all curvature arises from the normalization to the unit sphere.

Figure 9 illustrates how PGA provides a more compact description than PCA. The figure shows the normalized sum of the first n eigenvalues as a function of n. It can be seen that in order to capture say 99.5 percent of the variation, we will need 25 eigenvectors

when doing PCA as opposed to only 20 eigenvectors when doing PGA.

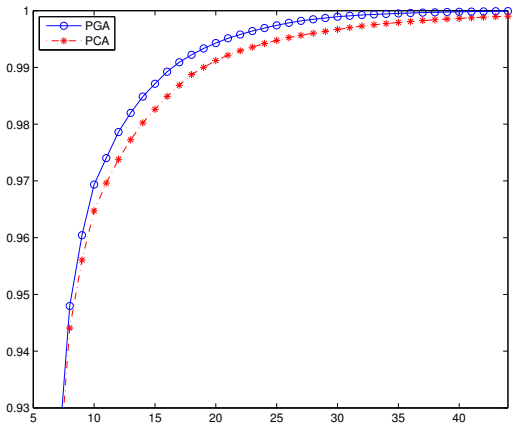


Figure 9. Accumulated spectrum of PGA and PCA.

5. Conclusion

In this paper we have introduced manifolds of preshapes built by constraining distributions of pseudo-landmarks between pairs of consecutive landmarks. This endows these preshape manifolds with a structure of Riemannian manifolds. We have developed tools for computing Exponential maps, Log maps, geodesic distances, allowing us to define a Geodesic GPA and adapt PGA to that situation. We have shown on examples that PGA captures variability better that PCA.

Although we have built our models for planar point configurations, they are clearly not restricted to this case. Other types of length and position constraints can also be used. We are also not restricted to shape manifolds. The techniques presented in this work can be used to perform statistics on other submanifolds of a linear configuration space implicitly defined by a set of smooth constraints. This is the subject of ongoing work.

References

[1] F. Bookstein. *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambrigde University Press, 1991.

[2] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry. Revised Second Edition*. Academic Press, 2003.

[3] J. Cates, P. T. Flechter, M. Styner, M. Shenton, and R. Whitaker. Shape Modeling and Analyis with Entropy-Based Particle Systems. In B. P. F. L. Nico Karssemeijer, editor, *Proceedings of the 20th International Conference on Information Processing in Medical Imaging*, volume 4584 of *LNCS*, pages 333–345, Kerkrade, The Netherlands, 2007. Springer.

[4] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A Minimum Description Length Approach to Statistical Shape Modeling. *IEEE Transactions on Medical Imaging*, 21(5):525–538, may 2002.

[6] J. Dedieu and D. Nowicki. Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds. *Journal of Complexity*, 21(4):487–501, Aug. 2005.

[7] P. T. Fletcher, S. Joshi, C. Lu, and S. M. Pizer. Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, Aug 2004.

[8] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339, 1991.

[9] D. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bulletin of London Mathematical Society*, 16:81–121, 1984.

[10] D. G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory*. Wiley and Sons, 1999.

[11] E. Klassen and A. Srivastava. *Geodesics Between 3D Closed Curves Using Path-Straightening*, volume 3951 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2006.

[12] E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:372—383, 2004.

[13] P. W. Michor and D. Mumford. An overview of the riemannian metrics on spaces of curves using the hamiltonian approach. *math/0605009*, Apr. 2006. Applied and Computational Harmonic Analysis 23 (2007), 74-113.

[14] W. Mio and A. Srivastava. Elastic-string models for representation and analysis of planar shapes. *Proceedings of the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:10—15, 2004.

[15] L. Noakes. A global algorithm for geodesics. *Journal of the Australian Mathematical Society*, 64:37–50, 1998.

[16] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.

8.

Conclusion

To end the thesis, a short summary of the presented work is given together with outlook and possibilities for future work. Confer in addition the contribution summary in the introduction.

1 Summary

The thesis presents work in three main areas: registration and deformation modeling; non-linear statistics and algorithms; and 2D shape modeling. The included papers together with the six papers [1, 2, 3, 18, 17, 38] that are not included in this thesis, constitute, along with the contributions of my colleagues, the results of my PhD studies at the Department of Computer Science, University of Copenhagen.

The developed *kernel bundle* framework and the *higher order kernels* both serve to allow sparse deformation description through increased description capacity. The kernel bundle framework introduces multi-scale representations in LDDMM while keeping much of the mathematical structure of LDDMM. We derive the KB-EPDiff equations, introduce sparse priors, develop a fast GPU-based algorithm, and evaluate the method on clinical data.

Each control point used in LDDMM codes only translational movement which limits the ability to compactly represent non-translational deformation such as rotations and dilations. With higher order kernels, we address this problem by showing how partial derivatives of kernels fit naturally into the LDDMM framework. We derive evolution equations, show connections between the order of the similarity measure and the kernels, and present a matching algorithm using the kernels. The increased description capacity allows registration with very few parameters, and we apply the kernels to register MR scans of patients suffering from Alzheimer's disease.

While Paper #1 and Paper #2 have been been submitted while writing this thesis, the four conference papers on the kernel bundle, Paper #3 and [1, 2, 3], have been published and have received interest in the registration community. The relation between the kernel bundle and different approaches to multi-scale LDDMM registration [7, 8] has been the subject of the recent paper [9]. Introducing sparsity in LDDMM has also been treated by Durrleman et al. [39, 40].

The algorithms for computing the first and second order differential of the Exponential presented in Paper #4 serve as tools for performing non-linear statistics, and, in particular, they allow Principal Geodesic Analysis to be computed without the common

tangent space linearization. We perform comparisons between the exact and linearized algorithm and thereby provide insight into the relation between curvature and spread on non-linear statistics.

Finally, to reduce variation introduced by manual annotations while keeping a consistent model, we have introduced the non-linear *bicycle chain* shape model. The representation results in a non-linear shape space, and we develop the necessary computational tools to perform statistics on the space. A notable benefit of the model is the reduction in dimensionality obtained by keeping the distance constraints enforced in the model. Experiments with the shape model is performed on outlines of human vertebrae obtained with lateral X-rays.

2 Outlook and Future Work

The kernel bundle framework and the higher order kernels will likely complement each other very well, and work on bringing them together will start right when the last word of this conclusion has been put down. This will include exploring sparse priors for the higher order kernels and different regularization for different orders kernels. Performing group wise statistics to learn the spatial locations of high frequency deformation across populations with the kernel bundle and higher order kernels is an interesting path for efficiently increasing sparsity while keeping the necessary flexibility of the deformation model. In addition, we wish to explore the coupling between scale information in images, the similarity measure, and the multi-scale deformation model.

Choosing the appropriate non-linear statistical tools for performing statistics on the deformation models is an interesting problem. While small deformation introduced by e.g. progressing atrophy may be measured using linearized tools, some applications may require more intrinsically non-linear methods. Performing statistics on scans of patients suffering from severe head trauma constitutes an example of this, and we are currently working on registration and statistical methods for such cases. For many applications, the smoothing terms used in the registration models will introduce bias in statistical exploration of registered data. We are looking into the possibility of using the explicit control of the deformation model in LDDMM and the kernel bundle framework to derive a registration formulation with less bias.

The above ideas constitute theoretical and modeling perspectives; using the deformation models for actually performing statistics on patients is the end goal. This requires robust software packages, testing, clinical evaluations, and further collaboration with clinicians. In the end, the work presented should hopefully be theoretically interesting in addition to actually being *useful*.

Bibliography

[1] S. Sommer, M. Nielsen, F. Lauze, and X. Pennec, "A Multi-Scale kernel bundle for LDDMM: towards sparse deformation description across space and scales," in *IPMI 2011*, Springer, 2011.

[2] S. Sommer, F. Lauze, M. Nielsen, and X. Pennec, "Kernel bundle EPDiff: evolution equations for Multi-Scale diffeomorphic image registration," in *SSVM 2011*, Springer, 2011.

[3] S. Sommer, M. Nielsen, and X. Pennec, "Sparsity and scale: Compact representations of deformation for diffeomorphic registration," in *MMBIA at WACV 2012*, 2012.

[4] R. Castillo, E. Castillo, R. Guerra, V. E. Johnson, T. McPhail, A. K. Garg, and T. Guerrero, "A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets," *Physics in Medicine and Biology*, vol. 54, pp. 1849–1870, Apr. 2009.

[5] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults," *Journal of Cognitive Neuroscience*, vol. 22, pp. 2677–2684, Dec. 2010. PMID: 19929323.

[6] L. Risser, F. Vialard, R. Wolz, D. D. Holm, and D. Rueckert, "Simultaneous fine and coarse diffeomorphic registration: application to atrophy measurement in alzheimer's disease," *MICCAI 2010*, vol. 13, no. Pt 2, pp. 610–617, 2010. PMID: 20879366.

[7] L. Risser, F. X. Vialard, R. Wolz, M. Murgasova, D. D. Holm, and D. Rueckert, "Simultaneous multi-scale registration using large deformation diffeomorphic metric mapping," *IEEE Transactions on Medical Imaging*, vol. 30, pp. 1746–1759, Oct. 2011.

[8] M. Bruveris, F. Gay-Balmaz, D. D. Holm, and T. S. Ratiu, "The momentum map representation of images," *0912.2990*, Dec. 2009.

[9] M. Bruveris, L. Risser, and F. Vialard, "Mixture of kernels and iterated Semi-Direct product of diffeomorphism groups," *arXiv:1108.2472*, Aug. 2011.

[10] X. Pennec, "Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements," *J. Math. Imaging Vis.*, vol. 25, no. 1, pp. 127–154, 2006.

[11] M. Frechet, "Les elements aleatoires de nature quelconque dans un espace distancie," *Ann. Inst. H. Poincare*, vol. 10, pp. 215–310, 1948.

[12] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Communications on Pure and Applied Mathematics*, vol. 30, no. 5, pp. 509–541, 1977.

[13] P. Fletcher, C. Lu, and S. Joshi, "Statistics of shape via principal geodesic analysis on lie groups," in *CVPR 2003*, vol. 1, pp. I–95–I–101 vol.1, 2003.

[14] P. Fletcher, C. Lu, S. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 8, pp. 995–1005, 2004.

[15] S. Huckemann and H. Ziezold, "Principal component analysis for riemannian manifolds, with an application to triangular shape spaces," *Advances in Applied Probability*, vol. 38, pp. 299–319, June 2006.

[16] S. Huckemann, T. Hotz, and A. Munk, "Intrinsic shape analysis: Geodesic PCA for riemannian manifolds modulo isometric lie group actions," *Statistica Sinica*, vol. 20, pp. 1–100, Jan. 2010.

[17] S. Hauberg, S. Sommer, and K. S. Pedersen, "Gaussian-like spatial priors for articulated tracking," in *ECCV 2010*, vol. 6311 of *Lecture Notes in Computer Science*, (Heraklion, Greece), pp. 425–437, Springer, Heidelberg, 2010.

[18] S. Hauberg, S. Sommer, and K. S. Pedersen, "Natural metrics and Least-Committed priors for articulated tracking," *Accepted for publication in Elsevier Journal on Image and Vision Computing*, 2011.

[19] R. Davies, C. Twining, and C. Taylor, *Statistical models of shape: optimisation and evaluation*. Springer, Aug. 2008.

[20] I. Dryden and K. Mardia, *Statistical Shape Analysis*. John Wiley & Sons, 1998.

[21] L. Younes, *Shapes and Diffeomorphisms*. Springer, 2010.

[22] D. G. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bull. London Math. Soc.*, vol. 16, pp. 81–121, Mar. 1984.

[23] F. L. Bookstein, "Size and shape spaces for landmark data in two dimensions," *Statistical Science*, vol. 1, no. 2, pp. 181–222, 1986.

[24] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models—their training and application," *Comput. Vis. Image Underst.*, vol. 61, p. 38–59, Jan. 1995.

[25] R. Davies, T. Cootes, and C. Taylor, "A minimum description length approach to statistical shape modelling," in *Information Processing in Medical Imaging*, pp. 50–63, 2001.

[26] P. W. Michor and D. Mumford, "Riemannian geometries on spaces of plane curves," *J. Eur. Math. Soc.*, vol. 8, pp. 1—48, 2004.

[27] P. W. Michor and D. Mumford, "An overview of the riemannian metrics on spaces of curves using the hamiltonian approach," *math/0605009*, Apr. 2006. Applied and Computational Harmonic Analysis 23 (2007), 74-113.

[28] P. W. Michor and D. Mumford, "Vanishing geodesic distance on spaces of submanifolds and diffeomorphisms," *Documenta Mathematica*, vol. 10, p. 26, 2004.

[29] J. Shah, *H^o type Riemannian metrics on the space of planar curves*. 2005.

[30] A. C. G. Mennucci, A. Yezzi, and G. Sundaramoorthi, *Sobolev–type metrics in the space of curves*. 2006.

[31] L. Younes, "Computable elastic distances between shapes," *SIAM J. Appl. Math*, vol. 58, pp. 565—586, 1998.

[32] W. Mio and A. Srivastava, "Elastic-string models for representation and analysis of planar shapes," *Proceedings of the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 10—15, 2004.

[33] S. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "A novel representation for riemannian analysis of elastic curves in rn," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–7, 2007.

[34] H. Blum and W. Wathen-Dunn, "A transformation for extracting new descriptors of shape," *Models for the Perception of Speech and Visual Form*, pp. 380, 362, 1967.

[35] K. Siddiqi and S. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 1 ed., Dec. 2008.

[36] S. Joshi, S. Pizer, P. T. Fletcher, P. Yushkevich, A. Thall, and J. S. Marron, "Multiscale deformable model segmentation and statistical shape analysis using medial descriptions," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 538–550, May 2002. PMID: 12071624.

[37] M. Vaillant, M. Miller, L. Younes, and A. Trouvé, "Statistics on diffeomorphisms via tangent space representations," *NeuroImage*, vol. 23, no. Supplement 1, pp. S161–S169, 2004.

[38] A. Tatu, F. Lauze, S. Sommer, and M. Nielsen, "On restricting planar curve evolution to finite dimensional implicit subspaces with Non-Euclidean metric," *Journal of Mathematical Imaging and Vision*, Aug. 2010.

[39] S. Durrleman, M. Prastawa, G. Gerig, and S. Joshi, "Optimal data-driven sparse parameterization of diffeomorphisms for population analysis," *Information Processing in Medical Imaging: Proceedings of the ... Conference*, vol. 22, pp. 123–134, 2011. PMID: 21761651.

[40] S. Joshi, "Optimal data-driven sparse parameterization of diffeomorphisms for population analysis," (Presentation at the "Geometry for Anatomy" workshop, Banff, Canada), 2011.