



PhD-FSTC-2016-56  
The Faculty of Sciences, Technology and Communication

## DISSERTATION

Defence held on 11/11/2016 in Belvaux

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN BIOLOGIE

by

**Florian BERNARD**

Born on 21 September 1986 in Saarburg, (Germany)

NOVEL METHODS FOR MULTI-SHAPE ANALYSIS

### Dissertation defense committee

Dr Rudi Balling, dissertation supervisor

*Professor, Université du Luxembourg*

Dr Jorge Goncalves, Chairman

*Professor, Université du Luxembourg*

Dr Peter Gemmar, Vice Chairman

*Professor, Hochschule Trier*

Dr Frank Hertel

*Head of Neurosurgery, Centre Hospitalier de Luxembourg*

Dr Stéphane Bordas

*Professor, Université du Luxembourg*



# Acknowledgements

Research is mostly collaborative effort and likewise this thesis was only possible with the help of many people. I consider myself to be in a very lucky position of having such a great team of supervisors that guided me through my time as PhD student, comprising Prof. Rudi Balling, Prof. Peter Gemmar, Prof. Jorge Goncalves and Dr. Frank Hertel. They are not only outstanding scientists in complementary disciplines, but in addition each of them has a great personality which made it always fun to meet and discuss. I appreciate that they always believed in me and gave me all the freedom I needed to pursue my ideas. I want to particularly express my gratefulness to Peter, who, not only during but already several years before my time as PhD student, has made available his support in many ways. This support was an essential ingredient to the success of this endeavour.

Also, the thesis defence committee members deserve to be mentioned. In addition to Rudi, Peter, Jorge and Frank, I would like to express my thankfulness to Prof. Stéphane Bordas for agreeing to be part of my thesis committee.

It is a great pleasure to thank Dr. Johan Thunberg. I always enjoyed discussing new ideas with him and learning from his great mathematical expertise. Not only, but in particular the transformation synchronisation method in Part II is the outcome of our collaboration. Without his regular input and critical feedback this thesis would not have been possible. I am greatly indebted to Andreas Husch, who accompanied me, in various roles, for more than a decade, be it as fellow student, friend, colleague or office mate. It was both encouraging and comforting to have someone who goes along the same path and encounters the same difficulties. Moreover, I want to thank him for filling in for me on various occasions and for all the fruitful discussions we had. I am thankful to Dr. Nikos Vlassis, who awakened my interest in the fascinating realm of mathematical optimisation in the very early stages of my PhD project, which has had a big impact on the entire shape of this thesis. Furthermore, I would like to show my appreciation to Dr. Luis Salamanca for the many helpful discussions, initiating reading clubs, and always being supportive in improving the appearance of my diagrams and posters. I am very grateful to Dr. Hans Lamecker, Dr. Stefan Zachow, Alexander Tack and Dennis Jentsch for our collaboration on the shape-aware surface reconstruction method in Chapter 5. Also, I want to thank Dr. Thomas Bühler, Anija Dokter and Simon Mary for their feedback on parts of this thesis. I want to express my thanks to the Systems Control Group at Luxembourg Centre

## Acknowledgements

---

for Systems Biomedicine (LCSB) for the warm and friendly atmosphere that made me always feel welcome.

I have spent a very pleasant time as visitor in the Computer Vision Group at Technical University Munich. It was a great honour for me that Prof. Daniel Cremers made it possible to visit his group, which I want to thank him for. Also, I want to thank Dr. Frank Schmidt for his excellent supervision during this time, and for sharing his remarkable knowledge of discrete optimisation with me. Furthermore, I am thankful for everyone from the Computer Vision group who contributed in some way to my great time, be it in the form of lectures or discussions related to convex optimisation, lunch break foosball sessions, or after-work dinners.

I also want to give credit to the Fonds National de Recherche, Luxembourg, for believing in my project and funding my work in form of an AFR PhD grant (6538106). I am extremely thankful for everyone who supported me during the grant acquisition phase. In addition to Rudi, Peter, Frank H. and Andreas, I want to mention Prof. Jörg Lohscheller, the LCSB grants office, and all the people that were in some way involved without being in direct contact with me.

It is a pleasure to thank all my friends who gave me the necessary distraction outside of work. I would like to thank my parents for their unconditional support and for making it possible that I could pursue a higher education degree. Last but not least I want to thank Nicole for her patience, support and all the time we have spent together.

*Luxembourg, 16 September 2016*

F. B.

# Abstract

Multi-shape analysis has the objective to recognise, classify, or quantify morphological patterns or regularities within a set of shapes of a particular object class in order to better understand the object class of interest. One important aspect of multi-shape analysis are Statistical Shape Models (SSMs), where a collection of shapes is analysed and modelled within a statistical framework. SSMs can be used as (statistical) prior that describes which shapes are more likely and which shapes are less likely to be plausible instances of the object class of interest. Assuming that the object class of interest is known, such a prior can for example be used in order to reconstruct a three-dimensional surface from only a few known surface points. One relevant application of this surface reconstruction is 3D image segmentation in medical imaging, where the anatomical structure of interest is known a-priori and the surface points are obtained (either automatically or manually) from images. Frequently, Point Distribution Models (PDMs) are used to represent the distribution of shapes, where each shape is discretised and represented as labelled point set. With that, a shape can be interpreted as an element of a vector space, the so-called shape space, and the shape distribution in shape space can be estimated from a collection of given shape samples. One crucial aspect for the creation of PDMs that is tackled in this thesis is how to establish (bijective) correspondences across the collection of training shapes. Evaluated on brain shapes, the proposed method results in an improved model quality compared to existing approaches whilst at the same time being superior with respect to runtime. The second aspect considered in this work is how to learn a low-dimensional subspace of the shape space that is close to the training shapes, where all factors spanning this subspace have local support. Compared to previous work, the proposed method models the local support regions implicitly, such that no initialisation of the size and location of these regions is necessary, which is advantageous in scenarios where this information is not available. The third topic covered in this thesis is how to use an SSM in order to reconstruct a surface from only few surface points. By using a Gaussian Mixture Model (GMM) with anisotropic covariance matrices, which are oriented according to the surface normals, a more surface-oriented fitting is achieved compared to a purely point-based fitting when using the common Iterative Closest Point (ICP) algorithm. In comparison to ICP we find that the GMM-based approach gives superior accuracy and robustness on sparse data. Furthermore, this work covers the transformation synchronisation method, which is a procedure for removing noise that

## Acknowledgements

---

accounts for transitive inconsistency in the set of pairwise linear transformations. One interesting application of this methodology that is relevant in the context of multi-shape analysis is to solve the multi-alignment problem in an unbiased/reference-free manner. Moreover, by introducing an improvement of the numerical stability, the methodology can be used to solve the (affine) multi-image registration problem from pairwise registrations. Compared to reference-based multi-image registration, the proposed approach leads to an improved registration accuracy and is unbiased/reference-free, which makes it ideal for statistical analyses.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Statistical Shape Models . . . . .	3
1.1.1 Correspondences . . . . .	4
1.1.2 Alignment . . . . .	5
1.1.3 Dimensionality Reduction and Probabilistic Interpretation . . . . .	6
1.1.4 Shape Reconstruction . . . . .	8
1.1.5 Challenges . . . . .	8
1.2 Motivation and Contributions . . . . .	9
1.2.1 Motivation and Scope . . . . .	10
1.2.2 Publications . . . . .	11
1.2.3 Content and Contributions of Publications . . . . .	12
1.3 Outline . . . . .	18
<b>I Statistical Shape Models</b>	<b>19</b>
<b>2 Shape Representation</b>	<b>21</b>
<b>3 Fast Correspondences for Statistical Shape Models of Brain Structures</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 Methods . . . . .	24
3.3 Results . . . . .	28
3.4 Conclusion . . . . .	30
<b>4 Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation</b>	<b>33</b>
4.1 Introduction . . . . .	34
4.1.1 PCA Variants . . . . .	34
4.1.2 Deformation Model Variants . . . . .	35

## Contents

---

4.1.3	Aims and Main Contributions . . . . .	36
4.2	Methods . . . . .	36
4.2.1	Notation . . . . .	37
4.2.2	Linear Shape Deformation Models . . . . .	37
4.2.3	Objective and Optimisation Problem . . . . .	38
4.2.4	Theoretical Motivation . . . . .	39
4.2.5	Block Coordinate Descent . . . . .	41
4.2.6	Factor Splitting . . . . .	41
4.3	Experimental Results . . . . .	42
4.3.1	Quantitative Measures . . . . .	42
4.3.2	Brain Structures . . . . .	43
4.3.3	Human Body Shapes . . . . .	48
4.4	Conclusion . . . . .	51
4.5	Supplementary Material . . . . .	51
4.5.1	The matrix $\mathbf{E}$ in eq. (4.8) . . . . .	51
4.5.2	Proximal Operators . . . . .	52
4.5.3	Computational Complexity . . . . .	54
4.5.4	Factor Splitting . . . . .	54
4.5.5	Convergence Plots . . . . .	55
4.5.6	Parameter Random Sampling . . . . .	55
<b>5</b>	<b>Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds</b>	<b>57</b>
5.1	Introduction . . . . .	58
5.2	Related Work . . . . .	59
5.3	Main Contributions . . . . .	62
5.4	Background . . . . .	63
5.4.1	Notation . . . . .	63
5.4.2	Point Distribution Models . . . . .	63
5.5	Problem Formulation . . . . .	64
5.5.1	A Generative Model . . . . .	64
5.5.2	Optimisation using EM . . . . .	66
5.6	Methods . . . . .	66
5.6.1	Surface Reconstruction using an Isotropic GMM . . . . .	67
5.6.2	Surface Reconstruction using an Anisotropic GMM . . . . .	68
5.6.3	Performance Analysis . . . . .	73
5.7	Experiments . . . . .	76
5.7.1	Evaluation Metrics . . . . .	76
5.7.2	Brain Shapes . . . . .	77
5.7.3	Knee Bones: Femur and Tibia . . . . .	81
5.7.4	Liver . . . . .	85

5.7.5 Hip . . . . .	88
5.8 Conclusion and Outlook . . . . .	91
<b>II Transformation Synchronisation</b>	<b>93</b>
<b>6 A Solution for Multi-Alignment by Transformation Synchronisation</b>	<b>95</b>
6.1 Introduction . . . . .	95
6.2 Methods . . . . .	97
6.2.1 Notation and Foundations . . . . .	97
6.2.2 Perfect Information . . . . .	98
6.2.3 Noisy Pairwise Transformations . . . . .	101
6.2.4 Affine Transformations in Homogeneous Coordinates . . . . .	101
6.2.5 Similarity Transformations . . . . .	102
6.2.6 Euclidean Transformations . . . . .	103
6.2.7 Rigid Transformations . . . . .	103
6.3 Experiments . . . . .	103
6.3.1 Noisy Transformations . . . . .	103
6.3.2 Generalised Procrustes Analysis . . . . .	106
6.4 Conclusion . . . . .	110
<b>7 Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation</b>	<b>111</b>
7.1 Introduction . . . . .	111
7.2 Methods . . . . .	113
7.2.1 Overview of Transformation Synchronisation . . . . .	113
7.2.2 Numerical Stability for Large Translations . . . . .	114
7.2.3 Multi-image Alignment . . . . .	115
7.3 Experiments . . . . .	116
7.4 Conclusion . . . . .	120
<b>III Deep Brain Stimulation</b>	<b>121</b>
<b>8 Potentials in Deep Brain Stimulation</b>	<b>123</b>
8.1 DBS Overview . . . . .	123
8.2 Planning . . . . .	124
8.2.1 Navigation . . . . .	124
8.2.2 Patient-specific 3D Models . . . . .	125
<b>Bibliography</b>	<b>145</b>



# 1 Introduction

The central idea of *computer vision* is to build algorithms that model visual perception capabilities, such that computers can “understand” visual input data (e.g. images). Understanding visual inputs is a task that the human visual system handles particularly well, as it can almost instantaneously determine the position, size, shape, colour or texture of objects (Purves et al., 1997). The (implicit or explicit) extraction of such properties of physical objects from images is an important aspect in order make computers understand images. In the era of *big data*, the interest in “teaching” computers how to understand images is growing continuously, as many datasets are simply too large to be analysed manually (e.g. in content-based image or video tagging in the web), or the manual analysis is very time-consuming and thus expensive (e.g. the manual segmentation of three-dimensional medical images).

Whilst *shape* being one of the relevant attributes for image understanding, the study of shapes on their own plays an important role in various other fields of applied science, including archaeology, biology and medicine (Dryden and Mardia, 1998). For example, morphogenesis, the description of the biological development of shape, is a long-standing area of research in biology, with Thompson’s pioneering work dating back to 1917 (Thompson, 1917). Even earlier, Galilei (1638) realised that bones of small and large animals differ not only in scale, but also in shape (translation by Dryden and Mardia (1998)).

From an abstract point of view, *shape analysis* has the objective to recognise, classify, or quantify morphological patterns or regularities of some entities of interest. These patterns may then help in developing theories of underlying concepts that are directly or indirectly related to the morphology. From a computational point of view, shape analysis is mostly tackled in the fields of computer science and applied mathematics, with (medical) image analysis, computational geometry, computer vision, pattern recognition and computer graphics comprising the predominant subfields that deal with the analysis

of shapes. Using a configuration of labelled points as *representation* (cf. Fig. 1.1), Kendall defines shape informally as “what is left when the differences which can be attributed to translations, rotations, and dilatations have been quotiented out” (Kendall, 1984). Kendall’s definition already points out a very important aspect in computational shape analysis, namely the invariance under similarity transformations. In *multi-shape analysis*, i.e. the analysis of a collection of shapes of a particular object class, this invariance is typically obtained by an *alignment* (or *registration*) procedure. In order to achieve the alignment, a mathematical transformation that minimises some distance measure between *corresponding* points is determined.

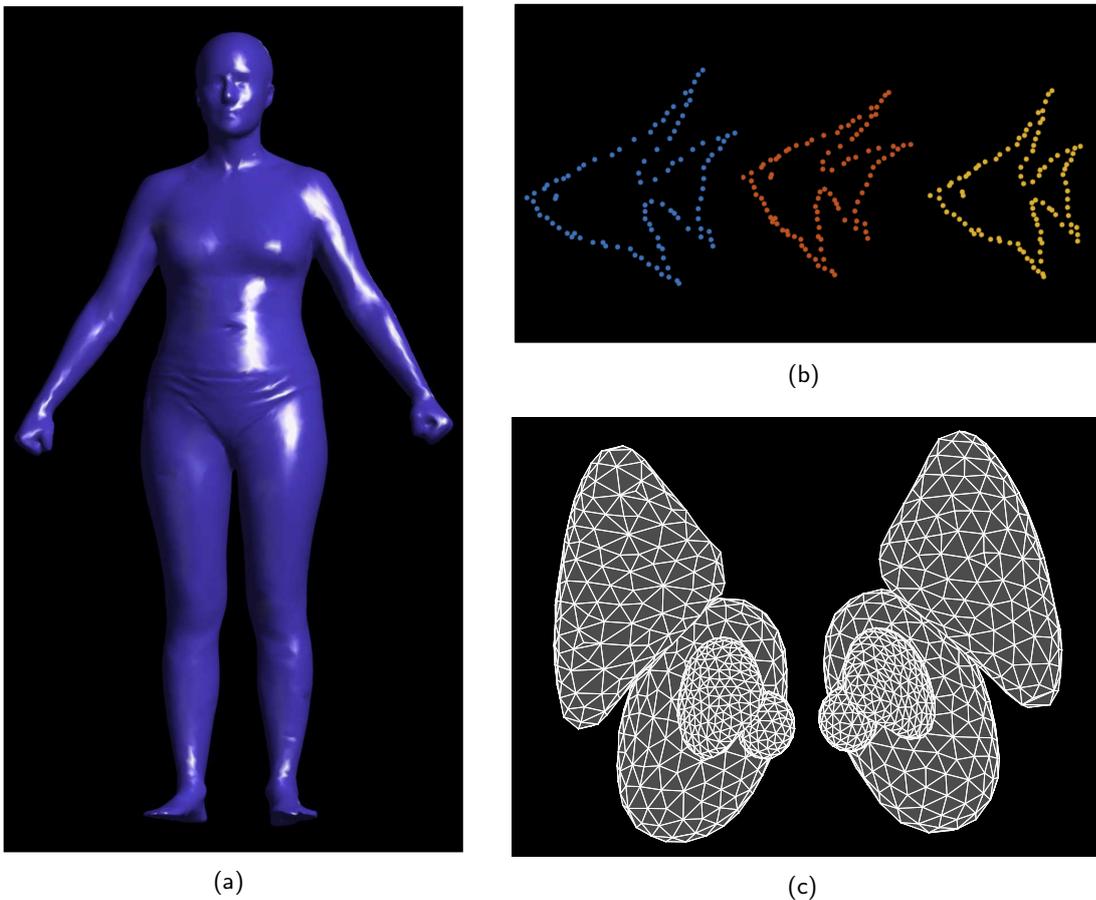


Figure 1.1: Some examples of shapes that are studied in this thesis. All shapes have in common that their underlying representation is based on point sets (note that the labelling of the point sets is not visualised to keep the visualisation clean). (a) Three-dimensional human body shape (Yang et al., 2014) visualised as surface rendering. (b) A collection of three two-dimensional fish shapes (Chui and Rangarajan, 2003) visualised as point-clouds. (c) Brain structure shapes visualised as wire-frame mesh.

The so-called *Statistical Shape Models* (SSMs) constitute a powerful tool for shape analysis. In SSMs, multiple shapes are studied within a statistical framework, where a collection of

shape samples is used to estimate the distribution of the shapes of interest. By assuming a certain class of adaptive models, estimating the shape distribution corresponds to learning the model parameters from training shapes, a common task in machine learning (Bishop, 2006). In certain scenarios, it is of interest to directly analyse the shape distribution. For example, in a medical context it may be relevant to compare the shape distribution of healthy organs to the distribution of pathological ones. Another important application is the use of the shape distribution as a prior for the *reconstruction* of surfaces from incomplete, noisy or ambiguous data. SSM-based *image segmentation* can also be considered a reconstruction problem where the given data takes the form of images. A third potential use-case of SSMs is *recognition* via *classification*, where the goal is to determine the shape distribution that is most likely to generate the given data, given multiple shape distributions (where each distribution pertains to one particular object class).

The overall purpose of this thesis lies in the advancement of current methodologies in the field of multi-shape analysis with respect to various aspects. In the previous paragraphs some of these aspects, such as the correspondence problem, SSMs, or shape-based surface reconstruction, have already been mentioned. Since SSMs can be seen as their common denominator, the first part of this thesis is devoted to topics related to SSMs. Those aspects of SSMs that are necessary for the specification of the focus as well as the main contributions are introduced in the next section. The second part of this thesis deals with the *transformation synchronisation* method. One potential application of this method is to solve the multi-alignment problem, which is relevant for the creation of SSMs (cf. Section 1.1.2). However, since the transformation synchronisation method is more general and not restricted to SSMs only, the second part is presented from a more general perspective.

## 1.1 Statistical Shape Models

A wide range of image processing and computer vision problems is tackled by the utilisation of geometric information. For example the generalised Hough transform (Ballard, 1981) for the detection of arbitrary shapes in images has been introduced more than 30 years ago and is still ubiquitous. Another example of a geometry-based methodology is the shape context descriptor (Belongie et al., 2002) that is used for shape matching.

For capturing geometric relations between multiple shapes of a particular object class, a popular method are the so-called *Point Distribution Models* (PDMs) (Cootes et al., 1992), where a shape is represented by a set of points – a point-cloud – and the analysis is conducted based on these points. The key components of PDMs are

- the establishment of point-wise *correspondences* between the shapes,
- the *alignment* of the collection of shapes, and
- a *dimensionality reduction* in order to find a *low-dimensional representation* that is sufficiently “close” to the training shapes.

Whilst the classical approach for the PDM creation considers these three components individually, there are also approaches that tackle them in a more unified framework. For example, for the creation of a 3D morphable model (Banz and Vetter, 1999) from 2D images, Cashman and Fitzgibbon (2013) formulate a unified energy optimisation problem. Whilst a unified formulation may be advantageous for certain tasks, the obtained energy function is in general highly non-convex and thus hard to optimise. Moreover, the unified approach is rather task-specific and less flexible compared to the classical PDM approach that considers the components individually. Thus, in the following we turn our attention to this classical approach of considering the components individually. Moreover, whilst there are other means for capturing the variability of shape, such as level sets (Cremers et al., 2006), using a medial axis description (m-reps) (Fletcher et al., 2003; Pizer et al., 2003), or parametrised continuous surfaces studied within a Riemannian framework (Kurtek et al., 2010), this work focuses on PDM-based representations only.

Up until this point, the term “SSM” has been used in a rather general way as one would understand it in common language usage, i.e. an SSM is some model of shapes that makes use of a statistical description thereof. From here on, the term “SSM” is used with the implicit assumption of an underlying PDM-based representation (unless stated otherwise). When it is beneficial to stress the point-based nature of an SSM, we speak of a “PDM”.

Whilst many of the elaborations in the remaining part of this section generalise to arbitrary dimensions, in order to keep the presentation simple, for now it is assumed that the considered shapes are three-dimensional.

### 1.1.1 Correspondences

A shape representation that is suitable for PDMs are (labelled) point-clouds. Here, a shape is represented by a finite set of points, commonly (but not necessarily) representing a discretisation of the *surface* of the object of interest. The labelling of the points is used in order to encode correspondences between the points of multiple shapes. For the creation of PDMs it is assumed that for the entire collection of shapes the number of points per shape is the same. Moreover, it is assumed that for each pair of shapes of the collection the labellings induce a bijection. Points across multiple shapes that share the same label are referred to as *homologous*. The task of finding homologous points between two or more shapes is an instance of the *correspondence problem* (van Kaick et al., 2011), which

is a widely studied problem class that appears in various contexts of computer vision. Among them are works considering correspondences between sets of points of images (Maciel, 2002; Maciel and Costeira, 2003), spectral techniques (Leordeanu and Hebert, 2005; Leordeanu, 2010), a Linear Programming relaxation for 3D shapes (Windheuser et al., 2011), graph and hypergraph matching (Zhou and De la Torre, 2013a,b; Nguyen et al., 2015), functional correspondences (Kovnatsky et al., 2015a), or shape matching (Rodolà et al., 2014).

An overview of correspondence methods in the context of SSMs is given in the review paper by Heimann and Meinzer (2009), where the authors distinguish between mesh-to-mesh registration, mesh-to-volume registration, volume-to-volume registration, parametrisation-to-parametrisation registration and population-based (or groupwise) optimisation. The first four approaches are generally of reference-based nature, i.e. a reference object, such as a mesh or volumetric image, is chosen and registrations between the reference and the remaining shapes in the collection are performed. In the latter approach, comprising the population-based methods, all shapes are considered simultaneously by formulating an optimisation problem that is solved jointly for all shapes. The population-based methods usually make use of some compactness assumption (Kotcheff and Taylor, 1998), based on the idea that shapes with “correct” correspondences are “closer” in shape space than those with “bad” correspondences. One popular compactness measure in this context is the minimum description length (MDL) (Davies et al., 2002). With their unified problem formulation and the property that the population-based approaches are in general reference-free, they have attracted a lot of attention in the community (Heimann and Meinzer, 2009). Whilst the reference-based approaches depend on the particular choice of the reference, they however have the advantage that they are in general faster compared to population-based methods. This is because the reference-based methods consider smaller search spaces when solving a series of pairwise registrations.

### 1.1.2 Alignment

The alignment of all shapes in the collection is conducted in order to remove *pose* differences such that only shape differences are captured within the PDM. The pose can be represented by transformations comprising translations, (proper) rotations, potentially uniform scaling and potentially reflections that are all defined with respect to some fixed coordinate system. The exact types of transformations that are allowed depend on the context, but most frequently these are rigid (allowing for translations and proper rotations), euclidean (allowing for translations and orthogonal transformations, also referred to as improper rotations), or similarity transformations (allowing for translations, scale and improper rotations). The multi-alignment problem, i.e. finding transformations that align all poses of shapes in a collection, is a problem that is widely studied. When the correspondences

are not known, this alignment is frequently referred to as *point-set registration* and is closely tied to the correspondence problem. Common approaches to tackle the point-set registration problem include the Iterative Closest Point method (ICP) (Besl and McKay, 1992; Maier-Hein et al., 2012; Maron et al., 2016), the Softassign Procrustes Matching algorithm (Rangarajan et al., 1997) and various probabilistic formulations (Chui and Rangarajan, 2000; Myronenko et al., 2007; Myronenko and Song, 2010; Jian and Vemuri, 2011; Horaud et al., 2011; Rasouliau et al., 2012). When the correspondences are known, aligning the pose of only two shapes is referred to as the *Absolute Orientation Problem* (Horn et al., 1988) or *Procrustes Analysis* (Gower and Dijksterhuis, 2004), which has various closed-form solutions, e.g. by means of Singular Value Decomposition (SVD). The case of aligning more than two shapes is known as *Generalised Procrustes Analysis*, which is frequently tackled by reducing the groupwise problem to a series of pairwise problems by using a fixed or evolving reference shape.

### 1.1.3 Dimensionality Reduction and Probabilistic Interpretation

The representation of a shape as labelled point-cloud allows to treat a shape as a single point in a high-dimensional vector space, the so-called *shape space*. The shape space covers all possible shapes in form of labelled point-cloud configurations. However, since the shapes of interest belong to a particular object class, only a (small) subset of the shape space contains shapes that are *valid* instances of this particular object class. The purpose of the dimensionality reduction step is to identify the manifold (as subset of the shape space) that contains those shapes that are valid instances of the object class of interest.

In principal, any dimensionality reduction technique can be used for this task. Since linear models, which represent a (low-dimensional) linear subspace of the shape space, are advantageous from a computational perspective, they are more frequently used compared to nonlinear approaches. A model that is linear in the high-dimensional shape space allows to model *nonlinear* deformations of the shape in 3D space. Usually, a *linear* model refers to a (linear) subspace of the *normalised* shapes in shape space. A common normalisation is to translate the shapes in such a way that the mean of all shapes in the collection is at the origin of the shape space. Without performing this normalisation explicitly, the proper terminology is to speak of an *affine* subspace, which is graphically illustrated in Fig. 1.2.

The most common way of finding a suitable subspace of the training shapes is by means of Principal Components Analysis (PCA) (Cootes et al., 1992). In this case the subspace is spanned by (a subset of) the eigenvectors of the (sample) covariance matrix of the training shapes in shape space. Frequently, the dimensionality of the resulting subspace, i.e. the number of eigenvectors chosen, is set in such a way that a certain proportion of the variance

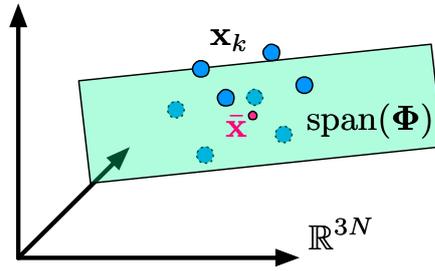


Figure 1.2: Illustration of a  $3N$ -dimensional shape space and an affine subspace (spanned by the columns of  $\Phi$ , shown in green) that is “close” to the training shapes  $\mathbf{x}_k$  (shown by blue circles where the solid contour indicates shapes that lie “above” the subspace and the dashed contour indicates shapes that lie “below” the subspace). The mean shape  $\bar{\mathbf{x}}$  is shown in red.

(commonly between 95% and 100%) of the training data is captured by the model. This is implemented by using the  $M$  eigenvectors corresponding to the  $M$  largest eigenvalues such that the desired amount of variance is captured by the  $M$ -dimensional subspace. The basis vectors of the subspace determined by the PCA-based solution are orthogonal, which is beneficial from a computational perspective since computing the (pseudo-)inverse of the matrix of basis vectors simplifies to transposing the matrix. Nevertheless, the description of the subspace is not restricted to eigenvectors, or orthogonal vectors. In principal, arbitrary vectors, which we refer to as *deformation factors*, or *factors* in short, can be used. For example, in the *sparse shape composition* method (Zhang et al., 2011, 2012) a sparse linear combination of the training shapes is employed.

One of the major advantages of *linear* models is that shapes that were not part of the training data but lie (approximately) in the obtained subspace can (approximately) be written as the sum of the mean shape and a *linear combination* of the factors.

Commonly it is assumed that all training shapes are samples that have been drawn from some (unknown) probability distribution. Within this interpretation, a *statistical* model can be obtained by assuming that the PDM coefficients follow some probability distribution. By estimating this probability distribution from the training shapes, a *generative* shape model can be defined. A common assumption is that the coefficients follow a Gaussian distribution. An appealing property of the PCA-based approach is that, due to the orthogonality of the eigenvectors of the covariance matrix, the covariance matrix of the Gaussian distribution of the coefficients is diagonal, i.e. the coefficients are (linearly) uncorrelated.

With imposing a probability distribution over the coefficients, a distribution over shapes in shape space is obtained (Albrecht et al., 2013). This probabilistic treatment enables

the incorporation of the shape model as prior into problem formulations. With that, the knowledge which shapes are more likely, and which ones are less likely, is integrated. The purpose of using this prior information is to complement insufficient data, for example due to noise, incompleteness, or ambiguities.

### 1.1.4 Shape Reconstruction

One interesting application of shape models is to use them for shape reconstruction. The main idea is to reconstruct the surface of an object of a particular class from only few 3D points that lie on the object’s surface, where the PDM serves as prior in order to augment the small amount of available data, cf. Fig. 1.3. The surface reconstruction is achieved by solving a regularised optimisation problem in order to find the parameters that best fit the PDM to the available 3D points. A particular difficulty is that in general the correspondences between the available 3D points and the PDM are unknown. Thus, the fitting procedure must solve for both the PDM parameters and the correspondences, which is commonly done in an alternating manner similar to ICP.

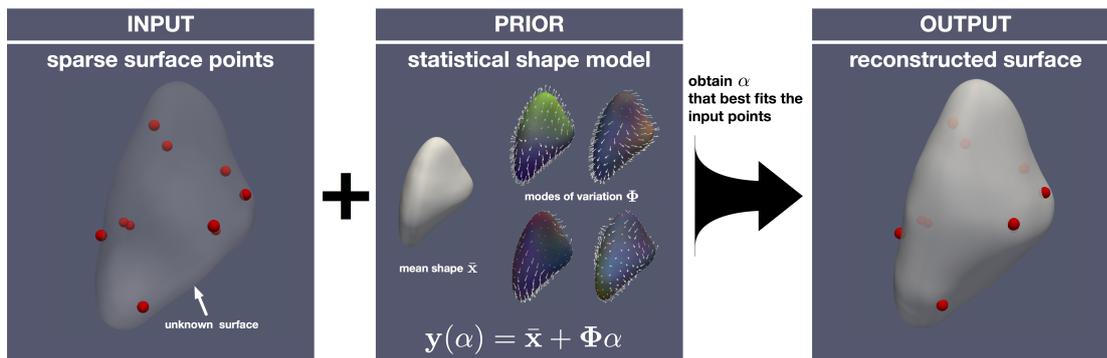


Figure 1.3: Overview of shape reconstruction from sparse 3D point-clouds using an SSM as prior.

### 1.1.5 Challenges

One important characteristic of SSMs is that a particular shape model is only able to represent objects of a specific class. As such, SSMs are only applicable when such contextual information is available, rather than “in the wild”. Examples of cases where this contextual information is available include face recognition or anatomical segmentation in medical imaging.

A particular problem that is highly relevant for SSMs is the correspondence problem, which occurs at various places due to the discrete nature of the PDM. For example, it needs to be solved in order to construct the PDM, but it also occurs during SSM-based shape

reconstruction. A particular difficulty is that in general the *ground truth* correspondences are unknown, and some surrogate, such as model compactness, as described in Section 1.1.1, is used in order to quantify the quality of particular correspondence assignments. Moreover, in some cases a correspondence might not even exist, e.g. when shapes are incomplete. The combinatorial nature of the correspondence problem contributes further towards the complexity from a computational perspective.

Another important aspect is that the shape model must on the one hand be able to generalise well to unseen data, whilst on the other hand it must be specific. The generalisation ability of a shape model can be evaluated using a leave-one-out approach, i.e. the shape model is trained using all but one training samples, and then it is checked how well the so-trained model can represent the left-out training sample. The specificity can be quantified by measuring the discrepancy between shape instances drawn from the shape distribution to the closest training shape (the probabilistic setting assumes that a probability distribution is imposed upon the PDM coefficients, cf. Section 1.1.3). An interesting property is that generalisation and specificity are somewhat conflictive. In particular, a constant model that is equal to one of the training shapes has perfect specificity, whilst having very poor generalisation. On the contrary, an extremely flexible model that spans the entire  $3N$ -dimensional shape space, cf. Fig. 1.2, with a uniform distribution over the coefficients, has perfect generalisation but very poor specificity. However, having either of these extreme cases is not meaningful in practice. Nevertheless, the choice whether generalisation or specificity has a higher priority depends on the particular application. For instance, when performing SSM-based segmentation, the generalisation to unseen data plays a central role. If the SSM is used in a generative manner for the synthesis of new shapes, e.g. for content generation in computer graphics, specificity may have a higher priority compared to generalisation.

One point that plays a major role in practice with respect to the quality of a shape model is that in general only a small amount of training shapes is available for learning the PDM. If the variability in the population is large, and the available training shapes do not cover the full range of variability, the lack of a sufficient amount of training data is even a more severe problem.

## 1.2 Motivation and Contributions

In the previous section we introduced the preliminaries that are necessary to specify the focus and the main contributions of this thesis. The problems covered in this thesis emerged in the context of an SSM-based segmentation problem in the area of deep brain stimulation (DBS) surgery. Thus, several of the subsequently introduced methodologies have in common that they are applied to brain structures. Nevertheless, the relevance of

the proposed techniques goes beyond the field of DBS and they are thus presented and evaluated in a broader context. As such, the main strength of this work are methodologies that are relevant in the wider fields of computer vision and pattern recognition (e.g. image segmentation, surface reconstruction, multi-alignment, or multi-image registration).

### 1.2.1 Motivation and Scope

The problems tackled in this thesis originate from the idea of achieving an interactive segmentation procedure based on SSMs for the segmentation of brain structures for deep brain stimulation (DBS) surgery. Despite a vast amount of individual work in the field of SSMs (Heimann and Meinzer, 2009), as well as in the field of interactive segmentation (McGuinness and O’Connor, 2010), there is only a limited amount of work that considers the use of SSMs in combination with interactive segmentation (cf. Section 5.2). The most common approach is that SSM-based segmentation methods are used in an automated manner first, followed by the subsequent manual correction of the results. However, a disadvantage of this two-step approach is that in the case of a failed automated segmentation, the subsequent manual correction may be very difficult and tedious. An advantage of integrating SSMs into an interactive segmentation procedure is that expert knowledge is incorporated from the very beginning in order to steer the algorithm towards producing the desired result. One such approach is based on *Interactive Shape Models* (van Ginneken et al., 2003), where the user manually moves *points* of the PDM. However, for that the user needs to identify the corresponding position of the model point in the image. This may be difficult, in particular if the object boundary does not exhibit distinct features and when dealing with three-dimensional images.

An approach that appears to be more intuitive and user-friendly is to let the user decide where to annotate the object boundaries in the image (e.g. contours can be drawn in high-contrast regions), and then performing a surface reconstruction by fitting a shape model to these annotations. This manual annotation/model fitting cycle can then be repeated until the result is considered as satisfactory by the user. Thus, the first step towards being able to fully exploit the power of SSMs in an interactive segmentation framework is the development of fundamental tools that facilitate the combination of SSMs and interactive segmentation, which is tackled in this thesis. Whilst interactive segmentation can be seen as motivating factor, it is not the main focus of this work and should only be considered as one possible perspective. Thus, the integration of the presented methodologies into an interactive setting is out of scope. However, in order to highlight some ideas that have shaped this thesis, a brief sketch of how the presented methodologies can be used for interactive segmentation in the context of DBS is given in Chapter 8. Since the applicability of the developed tools is not restricted to interactive segmentation only, they are presented in a broader context. To be more specific, the

following four aspects are considered in this work:

1. The *correspondence problem* for brain structure shapes,
2. learning an affine subspace of the shape space such that the *deformation factors* exhibit *local support*,
3. the *reconstruction of shapes from sparse 3D point-clouds*, and
4. the *transformation synchronisation* method in order to achieve transitive consistency in the set of pairwise linear transformations.

### 1.2.2 Publications

The core part of this thesis are the following publications containing the main contributions:

- (i) F. Bernard, N. Vlassis, P. Gemmar, A. Husch, J. Thunberg, J. Goncalves, F. Hertel: *Fast Correspondences for Statistical Shape Models of Brain Structures*. Proc. SPIE 9784. Medical Imaging 2016: Image Processing, 97840R, San Diego, March 2016, doi:10.1117/12.2206024.
- (ii) F. Bernard, P. Gemmar, F. Hertel, J. Goncalves, J. Thunberg: *Linear Shape Deformation Models with Local Support Using Graph-based Structured Matrix Factorisation*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, June 2016.<sup>1</sup>
- (iii) F. Bernard, L. Salamanca, J. Thunberg, A. Tack, D. Jentsch, H. Lamecker, S. Zachow, F. Hertel, J. Goncalves, P. Gemmar: *Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds*, preprint.
- (iv) F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, J. Goncalves: *A Solution for Multi-Alignment by Transformation Synchronisation*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, June 2015. doi:10.1109/CVPR.2015.7298828.<sup>2</sup>
- (v) F. Bernard, J. Thunberg, A. Husch, L. Salamanca, P. Gemmar, F. Hertel, J. Goncalves: *Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation*. MICCAI Workshop on Spectral Analysis in Medical Imaging. Munich, October 2015.

---

<sup>1</sup>© 2016 IEEE. Reprinted with permission from all authors.

<sup>2</sup>© 2015 IEEE. Reprinted with permission from all authors.

### 1.2.3 Content and Contributions of Publications

In the following, the papers (i)-(v) are brought into context, summarised and their main contributions are highlighted.

(i) *Fast Correspondences for Statistical Shape Models of Brain Structures*

**Background:** As discussed in Section 1.1.1, for the creation of PDMs it is essential that the available training shapes are brought into correspondence, which can be done in a reference-based or in a population-based manner. The main disadvantage of the reference-based methods is the necessity of choosing the reference, whereas the main drawback of the population-based methods is that they are slower due to the larger search space. Whilst in theory the correspondence establishment must only be conducted once, in practice this is rather an iterative procedure (Heimann and Meinzer, 2009). This implies that a correspondence establishment procedure that runs for several hours constitutes a bottleneck in the model generation phase. As such, it is desirable to have a method that establishes correspondences in a fast way such that delays due to the correspondence establishment method are kept small.

**Challenges:** One challenge is to develop a method that works well enough in practice such that no user-interaction or manual post-processing is necessary. Moreover, the choice of reference and the processing time constitute further challenges.

**Paper summary:** This paper tackles the correspondence problem for the creation of SSMs of brain structure meshes by combining point-set registration, the linear assignment problem and a mesh fairing procedure. For each shape, a probabilistic nonlinear point-set registration is performed in order to spatially align the shape point-cloud to the point-cloud of a chosen reference. By solving the linear assignment problem, the labelling of each shape point-cloud is obtained from the probabilistic correspondences. After applying the labelling of each aligned shape to the *original* shape point-cloud, the original shape point-cloud, now in correspondence with the reference, is combined with the mesh topology of the reference. Globally, in general this results in correspondences of good quality. Nevertheless, locally there may be small mesh irregularities since the mesh topology of the reference may be incompatible with that of the individual shapes. This is fixed by mesh fairing, a parametrisation-invariant mesh regularisation procedure. Whilst the evaluation of this method has been conducted in the context of brain structures, the proposed methodologies also apply to other objects with a morphology that is comparable to the considered brain structures, e.g. the objects are rather smooth and of simple nature.

**Contributions:** The contributions of this paper are application-oriented and

should rather be prescribed towards the engineering-side. They can be summarised as follows:

- The combination of well-established existing algorithms in order to *achieve a fully automated method that solves the correspondence problem of brain structures* for the creation of SSMs.
- In comparison to two existing population-based methods the proposed method is *preferable for the brain shapes dataset with respect to model quality*.
- Due to the simplified view of the correspondence problem in the form of individual subproblems, where each of them is computationally efficient, the proposed method is *superior with respect to runtime* compared to two existing population-based approaches.
- Moreover, due to the computational efficiency, it is feasible to evaluate multiple references in order to *reduce the effect of choosing a “bad” reference*.

(ii) *Linear Shape Deformation Models with Local Support Using Graph-based Structured Matrix Factorisation*

**Background:** Dimensionality reduction, commonly performed by PCA, is a key ingredient of SSMs, as discussed in Section 1.1.3. A downside of PCA-based factors is that they exhibit global support, i.e. adjusting a single coefficient deforms the entire shape. However, in certain scenarios, such as interactive shape deformations, or when interpretability is important (e.g. in a medical context), spatially localised deformation factors are more appropriate, such that adjusting a single coefficient results in a spatially localised shape deformation. Whilst several authors propose to use factors with *some* structure, cf. Section 4.1.2, the *Sparse Localised Deformation Components* method (SPLOCS) by Neumann et al. (2013) is among the few approaches that explicitly targets to obtain deformation factors with local support. For editing animated mesh sequences in computer graphics, the authors explicitly model the local support regions, which requires their location and size to be initialised. However, in a pattern recognition context the necessity of initialising the local support regions can be a downside, since it is often desirable to let the data determine where and how large the local support regions should be. Thus, our objective is to find a way of learning local support deformation factors from training data such that the local support regions do not require an initialisation of their location and size.

**Challenges:** One particular challenge is to find a problem formulation that on the one hand does not need an initialisation of the location and size of the local support regions, whilst on the other hand is not too difficult from a computational point of view. The latter is in particular a challenge since in the standard approach

of using PCA, a non-convex problem is (globally) solved. However, the PCA-based solution does not allow to incorporate arbitrary (convex) constraints or regularisers that would enable to impose additional structure upon the factors. Another application-oriented challenge is that the obtained factors should lead to realistic deformations.

**Paper summary:** A method is introduced that obtains a subspace from the collection of training shapes such that the resulting deformation factors have local support. In order to obtain local support deformation factors a matrix factorisation problem is formulated, where sparsity and graph-based smoothness regularisation terms are used in order to achieve spatially localised support regions.

**Contributions:** The contributions in this paper are of methodological nature and are summarised as follows:

- In the proposed method, the *location and size* of the local support regions *are determined automatically* via the sparsity and smoothness regularisers.
- In contrast to the SPLOCS method, where the local support regions are explicitly modelled, the presented approach results in an *implicit clustering* that is part of the optimisation procedure and *no initialisation of the local support region is necessary*.
- The optimisation problem is formulated in terms of a matrix factorisation framework that has a *well-grounded theoretical motivation*.
- Due to the incorporated smoothness regularisation term, the proposed method leads to *more realistic deformations* compared to other methods.

(iii) *Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds*

**Background:** One particular use-case of SSMs is shape reconstruction from sparse 3D point-clouds, where the SSM serves as prior describing plausible shapes and the reconstruction is performed by fitting the SSM points to the given points (cf. Section 1.1.4). Most existing approaches either assume that the correspondences between the available 3D points and the model points are known, or they estimate discrete correspondences in an ICP-like manner (more details on both aspects are discussed in Section 5.2). However, there may not even exist a unique model point that corresponds to a given point. Thus, discrete correspondences as they are frequently used may not be the best choice. Even though in many cases a surface mesh is associated with the SSM, the SSM fitting is mostly performed in a purely point-based manner. However, rather than fitting the SSM in a point-by-point fashion to the available 3D point-cloud, a surface-based fitting would be less sensitive to surface discretisation. This drawback has been tackled in recent work by Taylor et al. (2016), where a hand pose model fitting is performed using surface-based correspondences. In a similar fashion to the work by Taylor et al. (2016), our

objective is to achieve a fitting method for the surface reconstruction that goes beyond a purely point-based fitting. In contrast to their work, we pursue a different approach by formulating the fitting problem in a probabilistic setting.

**Challenges:** The main challenge for SSM-based surface reconstruction is the problem of finding for each available 3D point the corresponding point of the SSM (assuming such a point exists). Moreover, the reconstruction of a surface from a *sparse* point-cloud is challenging due to the small amount of data that is available and the (potentially) resulting ill-posedness of the problem.

**Paper summary:** In this work a surface reconstruction method from sparse data is presented. Given (few) points that lie on the surface of the object that is to be reconstructed, the surface reconstruction is performed with an SSM as prior. The reconstruction is achieved by assuming that the given 3D surface points are samples drawn from a Gaussian Mixture Model (GMM) (Bishop, 2006). The number of mixture components corresponds to the number of points of the PDM, and the mean of each mixture component corresponds to the 3D location of a PDM point, parametrised by the coefficients of the PDM. As such, the surface reconstruction problem can be formulated as a fitting problem of the shape model to the given points, where both correspondences and PDM parameters are the unknowns. Several different approaches to solve this problem are presented by variants and generalisations of the Expectation Maximisation (EM) algorithm (Dempster et al., 1977), where, for each given surface point, the correspondences are modelled in form of a *generating component*, which are treated as latent variable.

**Contributions:** The main contribution of this work is the introduction of a *surface-based* SSM fitting methodology that has superior accuracy and robustness compared to ICP on a broad range of datasets. The surface-based fitting procedure is achieved by the following methodological contributions:

- Due to the probabilistic formulation borrowed from probabilistic point-set registration methods, a shape-aware surface reconstruction method is achieved that does *not require explicit correspondences*.
- In order to achieve a surface-based fitting method, in contrast to a point-based fitting, a GMM with *anisotropic covariance matrices* is used, where the covariance matrices are *oriented according to the surface normals*.
- A *rigorous and self-contained derivation* of the anisotropic GMM formulation is presented that leads to the Expected Conditional Maximisation (ECM) algorithm, which shares the *same convergence properties as the EM algorithm*, whilst at the same time being more general.
- A fast approximation of the anisotropic GMM that has the *same computational complexity as the isotropic GMM-based approach* is introduced.

(iv) *A Solution for Multi-Alignment by Transformation Synchronisation*

**Background:** Obtaining transitive consistency in the set of pairwise transformations, i.e. transforming from coordinate systems A to B, and then from B to C must be identical as transforming A directly to C, has a wide range of applications in vision and machine learning. In the literature, various methods have tackled this synchronisation problem for orthogonal transformations, proper rotations, or permutations (Govindu, 2004; Singer, 2011; Singer and Shkolnisky, 2011; Hadani and Singer, 2011b; Pachauri et al., 2013; Bandeira et al., 2013; Chaudhury et al., 2013; Wang and Singer, 2013; Chatterjee and Govindu, 2013; Boumal, 2015), some of them considering translations in addition. Our objective is to generalise these approaches such that not only orthogonal but general linear/affine transformations can be synchronised in arbitrary dimensions, in order to solve the multi-alignment problem (cf. Section 1.1.2) in a reference-free and direct manner.

**Challenges:** A particular challenge is to solve the multi-alignment problem in arbitrary dimensions in a *direct* way. Previous approaches are iterative (Gower, 1975; Bartoli et al., 2013), where the latter involves a nonlinear refinement, or focus on the 2D/3D cases only (Pizarro and Bartoli, 2011).

**Paper summary:** The *transformation synchronisation* method is a procedure to remove noise that accounts for transitive inconsistency in the set of pairwise linear transformations. The proposed method is of general nature, i.e. it is applicable whenever the set of all pairwise (invertible) linear transformations is not transitively consistent and the objective is to obtain transitive consistency. By first computing all pairwise transformations and then synchronising them, the method can be used to solve the multi-alignment problem that occurs during PDM creation (Section 1.1.2). Since affine transformations can be represented by a homogeneous transformation matrix, they are essentially linear transformations (with certain properties) and can also be handled by the method. Moreover, when considering subsets of linear transformations, such as orthogonal, rigid, similarity or euclidean transformations, the transformation synchronisation method can also retrieve these transformations by means of projections onto the set of interest.

**Contributions:** The main contributions of this paper are rather on the theoretical/mathematical side and can be summarised as follows:

- A *linear algebraic property* is provided that must hold whenever the set of pairwise linear transformations is transitively consistent.
- From this algebraic relation, a *least-squares problem* with an orthogonality constraint is introduced in order to obtain transitively consistent pairwise transformations from noisy pairwise transformations. The least-squares problem can be *solved in a direct way* via means of SVD.

One particular improvement of this work considering the numerical stability is described in (v). In addition, various enhancements of this work have been developed that are not part of this thesis. Among them are a formulation that considers weights for the pairwise transformations, including the case of missing data where only a subset of the pairwise transformations is available, the direct formulation of an optimisation problem rather than stating a linear algebraic property, as well as distributed methods (Thunberg et al., 2015).

- (v) *Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation*

**Background:** Multi-image registration, i.e. finding spatial transformations that align a collection of images, has various applications in image processing and vision, including image stitching or the normalisation of medical images to a common coordinate system. Multi-image registration is usually performed in a reference-based manner, where the reference may be *fixed* or *evolving* (Joshi et al., 2004; Reuter et al., 2012), or by simultaneous optimisation over all transformations (Learned-Miller, 2006; Wachinger and Navab, 2013). However, choosing a reference induces a bias, and the simultaneous optimisation approaches are local methods that rely on a good initialisation. Whilst the type of admissible transformations (e.g. affine or nonlinear) generally depends on the context, we focus on the affine multi-image registration problem, which is for example relevant for the affine average template construction, but also as initialisation for nonlinear multi-image registration. Our objective is to solve the (affine) multi-image registration problem in an unbiased/reference-free manner that does not require an initialisation. For that, we assume that the set of pairwise transformations is given such that the problem reduces to a synchronisation problem.

**Challenges:** A particular challenge when applying the transformation synchronisation method to multi-image registration is that in principal the coordinate origin for each image can be arbitrary. Thus, the translation components in the pairwise transformations may be very large, which impairs the numerical stability of the original SVD-based solution of the transformation synchronisation method.

**Paper summary:** In this paper, the transformation synchronisation is extended with respect to numerical stability such that it can be applied to (affine) multi-image registration. Using this method, the groupwise affine registration of images can be tackled via computing the set of pairwise transformations using any pairwise (affine) image registration procedure and then synchronising them.

**Contributions:** The main contribution of this work is the introduction of a *numerically stable implementation of the transformation synchronisation method* that enables the practical use of the transformation synchronisation method for multi-image registration. From a practical point of view this has mainly two

implications:

- Since the approach is *reference-free*, the method is *unbiased* and thus well-suited for statistical analyses.
- Experiments suggest that the reference-based multi-image registration method performs significantly worse for certain choices of reference images. In this case the particular choice of the reference is essential. Since this is not the case when using the proposed transformation synchronisation, the method is *more robust* with respect to outliers and achieves *higher registration accuracy* on average compared to reference-based registration.

### 1.3 Outline

Part I of this thesis is devoted to topics that are related to SSMs, starting with a brief introduction of the shape representation in Chapter 2, followed by presenting the papers (i)-(iii) in Chapters 3-5. Whilst the chosen order of presentation corresponds to the chronological order of first establishing correspondences, creating the PDM and eventually using the PDM, there is no direct dependency between Chapters 3-5, so they can be read independently. Whilst, each of the Chapters 3-5 is self-contained, in order to better grasp their inter-relations it is recommended that Chapter 2 is read first. Part II contains the papers (iv) and (v) in Chapters 6 and 7, respectively, concerning the *transformation synchronisation* method, where Chapter 6 should be read before Chapter 7. In Part III, a brief sketch of applying the methodologies introduced in Parts I and II to deep brain stimulation (DBS) surgery is presented.

Parts I and II do not directly depend on each other, so it is left to the reader with which one to start. However, Parts I and II should be read before Part III.

# Statistical Shape Models **Part I**



## 2 Shape Representation

This chapter has the purpose of briefly introducing the representation of shapes as they are used throughout Chapters 3-5. As mentioned in Section 1.1.1, a suitable shape representation for PDMs are labelled point-clouds. We assume that the points of such a point-cloud represent vertices of a triangular surface mesh. In the following, this shape representation is formalised and particular characteristics appearing in Chapters 3-5 are clarified.

**Surface meshes:** Let  $\{\mathcal{M}_k\}_{k=1}^K$  be a collection of  $K$  shape surfaces, each of them defined as a triangular surface mesh  $\mathcal{M}_k = (\mathbf{X}_k, \mathcal{F}_k)$ . The matrix  $\mathbf{X}_k \in \mathbb{R}^{N \times 3}$  contains  $N$  point positions (also referred to as vertex or landmark positions) in 3 dimensions and can be interpreted as labelled point-cloud. The row index of the matrix  $\mathbf{X}_k$  can be seen as the label for each point, which is used to encode correspondences across multiple shapes. Thus, the  $i$ -th row,  $1 \leq i \leq N$ , in each  $\mathbf{X}_k$  for  $k = 1, \dots, K$  is a corresponding (or homologous) point. The set  $\mathcal{F}_k \subseteq \{(p, q, s) : p, q, s \in \{1, \dots, N\}, |\{p, q, s\}| = 3\}$  contains oriented triangular faces and represents the mesh topology.

**PDMs:** For defining the PDM it is common to vectorise the matrix  $\mathbf{X}_k \in \mathbb{R}^{N \times 3}$  by stacking its columns, i.e.  $\mathbf{x}_k = \text{vec}(\mathbf{X}_k) \in \mathbb{R}^{3N}$ , and to arrange the entire shape collection in the matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K] \in \mathbb{R}^{3N \times K}$ . For example, a PCA-based PDM is given by computing the deformation factors as the eigenvectors of the sample covariance matrix  $\mathbf{C} = \frac{1}{K-1}(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_K^T)(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_K^T)^T$ , where  $\bar{\mathbf{x}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k$  denotes the mean of all shapes in  $3N$ -dimensional shape space and  $\mathbf{1}_K$  is the constant column vector of size  $K$  containing ones. Let  $\Phi \in \mathbb{R}^{3N \times M}$  be a matrix containing the first  $M$  eigenvectors of  $\mathbf{C}$  with largest eigenvalues. For  $\alpha$  being a variable in  $\mathbb{R}^M$ , the PDM  $\mathbf{y}(\alpha) : \mathbb{R}^M \rightarrow \mathbb{R}^{3N}$  is a vector-valued function defined as

$$\mathbf{y}(\alpha) = \bar{\mathbf{x}} + \Phi\alpha, \quad (2.1)$$

## Chapter 2. Shape Representation

---

where  $\boldsymbol{\alpha} \in \mathbb{R}^M$  is the shape deformation parameter. The deformation of vertex  $i$  through  $\boldsymbol{\alpha}$  is denoted by

$$y_i(\boldsymbol{\alpha}) = \bar{x}_i + \boldsymbol{\Phi}_i \boldsymbol{\alpha}, \quad (2.2)$$

where the three rows of vertex  $i$  are selected appropriately from  $\mathbf{y}(\boldsymbol{\alpha})$ ,  $\bar{\mathbf{x}}$  and  $\boldsymbol{\Phi}$  to obtain  $y_i(\boldsymbol{\alpha}) \in \mathbb{R}^3$ ,  $\bar{x}_i \in \mathbb{R}^3$  and  $\boldsymbol{\Phi}_i \in \mathbb{R}^{3 \times M}$ . Due to vectorisation, the rows with indices  $\{1, \dots, N\}$ ,  $\{N+1, \dots, 2N\}$  and  $\{2N+1, \dots, 3N\}$  of  $\mathbf{X}$ ,  $\mathbf{x}_k$ ,  $\bar{\mathbf{x}}$ ,  $\boldsymbol{\Phi}$  or  $\mathbf{y}(\boldsymbol{\alpha})$ , correspond to the  $x$ ,  $y$  and  $z$  components of the  $N$  vertices, respectively.

**Shared topology:** The purpose of the methodology described in Chapter 3 is to establish correspondences between the vertex matrices  $\mathbf{X}_k$  of the meshes  $\mathcal{M}_k = (\mathbf{X}_k, \mathcal{F}_k)$  for all  $k = 1, \dots, K$ . However, since correspondences are encoded by the order of the rows of the  $\mathbf{X}_k$  matrices, when performing a modification of the order of the rows of a particular matrix  $\mathbf{X}_k$ , the triangular faces  $\mathcal{F}_k$  need to be updated as well. Instead of having an individual mesh topology for each mesh, we perform the correspondence establishment in such a way that, after all meshes have been processed, all meshes share the *same* triangular face structure  $\mathcal{F}$ , such that the mesh collection can be written as  $\{\mathcal{M}_k = (\mathbf{X}_k, \mathcal{F})\}_{k=1}^K$ . In order to simplify the notation, in the subsequent Chapters 4 and 5 we usually refer only to the shape collection  $\{\mathbf{X}_k\}_{k=1}^K$  that have a common mesh topology.

**Graph-based vertex connectivity:** In Chapter 4, the shared triangular face structure  $\mathcal{F}$  only plays a role for visualisation and is thus not introduced explicitly. However, an additional layer of connectivity between the vertices is encoded by a weighted undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\omega})$  that is shared by all shapes. The purpose of this graph is to model how much the deformation of a certain vertex should affect the deformation of another vertex. In this graph, the node set  $\mathcal{V} = \{1, \dots, N\}$  enumerates all  $N$  vertices, the edge set  $\mathcal{E} \subseteq \{1, \dots, N\}^2$  represents the connectivity of the vertices, and  $\boldsymbol{\omega} \in \mathbb{R}_+^{|\mathcal{E}|}$  is the weight vector which defines the weight for each edge. The (scalar) weight  $\omega_e$  of edge  $e = (i, j) \in \mathcal{E}$  denotes the affinity between vertex  $i$  and  $j$ , where “close” vertices have high value  $\omega_e$ . The graph can either encode pairwise *spatial* connectivity, in which case it can be derived from the triangular face structure  $\mathcal{F}$ , or affinities that are not of spatial nature (e.g. symmetries, or prior anatomical knowledge in medical applications).

**Surface normals:** In Chapter 5, the mesh structure of the shapes is used in order to compute surface normals. Thus, for each vertex with index  $i$ , the “left” and “right” neighbour vertices are denoted to have index  $i_2$  and  $i_3$ , respectively. With that, the surface normal at vertex  $i$  depending on the shape deformation parameter  $\boldsymbol{\alpha}$ , is given by  $n_i(\boldsymbol{\alpha}) = \frac{(y_{i_2}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha})) \times (y_{i_3}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha}))}{\|(y_{i_2}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha})) \times (y_{i_3}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha}))\|}$ , cf. (5.19).

# 3 Fast Correspondences for Statistical Shape Models of Brain Structures

## Abstract

Statistical Shape Models based on Point Distribution Models are powerful tools for image segmentation or shape analysis. The most challenging part in the generation of Point Distribution Models is the identification of corresponding landmarks among all training shapes. Since in general the true correspondences are unknown, correspondences are frequently established under the hypothesis that correct correspondences lead to a compact model, which is mostly tackled by continuous optimisation methods. In favour of the prospect of an efficient optimisation, we present a simplified view of the correspondence problem for statistical shape models that is based on point-set registration, the linear assignment problem and mesh fairing. At first, regularised deformable point-set registration is performed and combined with solving the linear assignment problem to obtain correspondences between shapes on a global scale. With that, rough correspondences are established that may not yet be accurate on a local scale. Then, by using a mesh fairing procedure, consensus of the correspondences on a global and local scale among the entire set of shapes is achieved. We demonstrate that for the generation of Statistical Shape Models of deep brain structures, the proposed approach is preferable over existing population-based methods both in terms of a significantly shorter runtime and in terms of an improved quality of the resulting shape model.

## 3.1 Introduction

Statistical Shape Models (SSMs) play a crucial role in medical image segmentation. The general idea of SSMs is to learn the statistical distribution of shapes from training data. A common choice for modelling shape variability is the Point Distribution Model (PDM) based on principal components analysis (PCA) (Heimann and Meinzer, 2009). The most challenging part of the PDM generation is the identification of homologous points, i.e.

points that correspond to each other, in the training shapes. Several approaches for finding these correspondences have been proposed (Heimann and Meinzer, 2009), among them are registration-based (Durrleman et al., 2009, 2014; Lüthi et al., 2013; Rasoulian et al., 2012) and population-based methods, where correspondences are established such that the resulting SSM is compact (Cates et al., 2006; Davies et al., 2010, 2002; Heimann et al., 2005; Thodberg, 2003).

The population-based methods have in common that the landmark positions are optimised by *moving* landmarks on the shape surface. One has to take particular care that the landmarks do not leave the shape surface. This can be accomplished by reparametrisation, i.e. a mapping of the shapes to a manifold that does not allow that landmarks leave the shape surface (e.g. planes), or manifolds where an efficient reprojection back onto the shape surface is possible (e.g. spheres (Heimann et al., 2005)). One way of recasting landmarks onto the shape surface that is frequently used in conjunction with reparametrisation, is the iterative reprojection of the landmarks onto the shape surface after (free) landmark translation (Dalal et al., 2007; Heimann et al., 2005; Munsell et al., 2012). A drawback of iterative reprojection is that it is performed repeatedly during optimisation, which hampers an efficient optimisation.

In order to enable an efficient optimisation, we present a simplified view of the correspondence problem for SSMs that is based on a multi-stage processing pipeline. At first, a regularised deformable point-set registration is performed and combined with solving the linear assignment problem for obtaining correspondences between shapes on a global scale. Then, by using a mesh fairing procedure (Desbrun et al., 1999) that is invariant to mesh parametrisations, consensus of the correspondences on a global *and* local scale among the entire set of shapes is achieved. Since all involved methods can be implemented in an efficient way, the evaluation of multiple putative groupwise correspondences comes into reach, which again adds a hint of population-basedness.

We demonstrate that for the generation of SSMs of deep brain structures the proposed approach is preferable over existing population-based methods due to the significantly shorter runtime and an improved quality of the resulting shape model.

### 3.2 Methods

Let  $\{\mathcal{M}_k\}_{k=1}^K$  be the set of  $K$  shape surfaces, each of them represented as a triangle mesh  $\mathcal{M}_k = (\mathbf{X}_k, \mathcal{F}_k)$ , with  $\mathbf{X}_k \in \mathbb{R}^{N \times 3}$  being a matrix that contains the 3D vertex (or landmark) positions in each row and  $\mathcal{F}_k \subseteq \{(p, q, s) : p, q, s \in \{1, \dots, N\}, |\{p, q, s\}| = 3\}$  being the set of oriented triangular faces that represents the mesh topology.

We assume that within each triangular mesh  $\mathcal{M}_k$  the distance between all pairs of

neighbour vertices is approximately equal (uniform surface sampling) and that the number of landmarks  $N$  is equal across all  $K$  shapes. As such, it is guaranteed that there exists a bijection between each pair of *landmark matrices*  $\mathbf{X}_k$  and  $\mathbf{X}_{k'}$  for  $k, k' = 1, \dots, K$ . Generating meshes with these properties from binary voxel-based segmentations is performed using the *iso2mesh* tool (Fang and Boas, 2009). It is assumed that all shape surfaces are rigidly aligned, which in our case is achieved by (rigid) image registration of the images that are used to create the binary voxel-based segmentations (Bernard et al., 2014).

An overview of the processing pipeline is shown in Fig. 3.1, where the interaction between the following subproblems (SP) is visualised.

- (SP1) For all vertex matrices  $\mathbf{X}_1, \dots, \mathbf{X}_K$ , find a deformable transformation that spatially aligns them with a reference vertex matrix  $\mathbf{X}_r$ . The choice of reference mesh  $\mathcal{M}_r = (\mathbf{X}_r, \mathcal{F}_r)$  will be described at the end of this section.
- (SP2) Find a bijective mapping between the *transformed* vertex matrices and the reference vertex matrix.
- (SP3) Construct meshes from the topology  $\mathcal{F}_r$  of the reference mesh and the *original* vertex matrices after they have been brought into correspondence with  $\mathbf{X}_r$  using the bijective mapping from SP2.
- (SP4) Apply mesh fairing as parametrisation-invariant regularisation.
- (SP5) Project the smoothed meshes back onto the original surfaces.

(SP1) Using deformable point-cloud registration, for each mesh  $\mathcal{M}_k$  the vertices  $\mathbf{X}_k$  are transformed such that they are in agreement with the vertices  $\mathbf{X}_r$  of the reference mesh. For that, the Coherent Point Drift (CPD) algorithm (Myronenko and Song, 2010) for point-cloud registration is used. CPD considers the registration of two point-sets as probability density estimation problem, where an optimisation procedure based on Expectation Maximisation (EM) is used to estimate the parameters of a Gaussian Mixture Model. Using Tikhonov regularisation, the method ensures that during registration all points move coherently, i.e. points that are close move in a similar direction. Assuming that all objects represent the same class of (anatomical) structures (e.g. the thalamus), and due to the uniform surface sampling of the vertices in the original meshes, the vertex matrices  $\tilde{\mathbf{X}}_k$  after the deformation also exhibit an approximately uniform surface sampling on the surface of the reference mesh. However, even with both point-clouds having the same number of points, CPD-based registration does not directly produce a bijective mapping between both point-sets, i.e. for a vertex there may be multiple valid assignments due to the probabilistic nature of CPD.

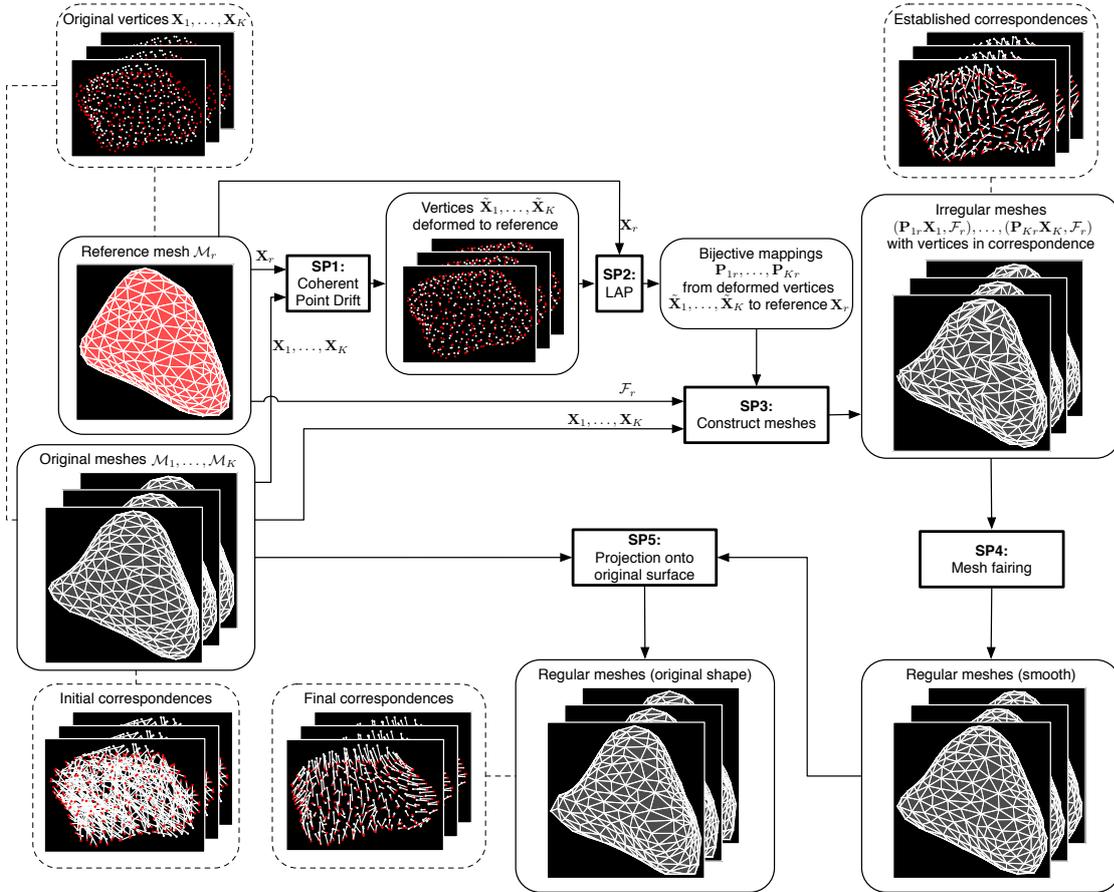


Figure 3.1: Processing pipeline for finding correspondences between meshes  $\mathcal{M}_1, \dots, \mathcal{M}_k$  by using reference mesh  $\mathcal{M}_r$ . Rectangles represent *methods*, rounded rectangles represent *data* and rounded rectangles drawn with a dashed line visualise *additional information* of the connected data node. The reference is shown in red and the objects  $1, \dots, K$  are shown in white/grey.

(SP2) To deal with this, next a bijective mapping between each transformed landmark matrix  $\tilde{\mathbf{X}}_k$  and the reference  $\mathbf{X}_r$  is established. Since each mapping can be represented by the ordering of the rows of the matrices  $\tilde{\mathbf{X}}_k$ , the objective is to find the permutation matrices  $\mathbf{P}_{1r}, \dots, \mathbf{P}_{Kr}$  that rearrange the rows of  $\tilde{\mathbf{X}}_k$  by  $\mathbf{P}_{kr}\tilde{\mathbf{X}}_k$  such that they best fit  $\mathbf{X}_r$ . This is a common problem that is known as linear assignment problem (LAP), which can be solved using the Auction algorithm (Bertsekas, 1998) with quasi-quadratic average-case time complexity<sup>1</sup>.

(SP3) Whilst the bijective mappings  $\mathbf{P}_{1r}, \dots, \mathbf{P}_{Kr}$  have been established using the transformed point-clouds  $\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_K$ , these mappings are now applied to the *original* vertex matrices  $\mathbf{X}_1, \dots, \mathbf{X}_K$ . This results in the set of meshes  $\{(\mathbf{P}_{kr}\mathbf{X}_k, \mathcal{F}_r)\}_{k=1}^K$ , where

<sup>1</sup>code available on <http://www.mathworks.com/matlabcentral/fileexchange/48448>

the faces  $\mathcal{F}_r$  of the reference mesh are joined with the reordered vertex matrices  $\mathbf{P}_{kr}\mathbf{X}_k$ . Here, the original vertex positions are used for mesh construction in order to retain the original shape (in contrast to the deformed shape after CPD-based deformation), whilst at the same time being in correspondence with  $\mathbf{X}_r$  (due to the *coherency* of the deformation from  $\mathbf{X}_k$  to  $\tilde{\mathbf{X}}_k$  for all  $k$ ). Combining the mesh topology  $\mathcal{F}_r$  with the vertices  $\mathbf{P}_{1r}\mathbf{X}_1, \dots, \mathbf{P}_{Kr}\mathbf{X}_K$  may lead to irregular meshes on a *local scale*, i.e. there may be small self-intersections and non-smoothness (see the box titled “Irregular meshes” in Fig. 3.1 top right). However, globally the correspondences are in general of good quality due to the well-known robustness of CPD (see the box titled “Established correspondences” in Fig. 3.1). Local irregularities are present because the graphs of the meshes  $\{\mathcal{M}_k\}_{k=1}^K$  are not isomorphic, and whilst there exists a bijection between their vertices, this bijection does in general not induce a diffeomorphism between the surfaces.

**(SP4)** In order to mend mesh irregularities, a volume-preserving mesh fairing procedure based on curvature flow is applied (Desbrun et al., 1999). This can be seen as a parametrisation-invariant noise removal procedure that relies merely on intrinsic properties of the surface. It has the effect that vertices are moved mostly along their normals and as such the triangular faces do not drift excessively over the mesh surface. With that, the previously established correspondences are globally kept intact whilst at the same time achieving consensus of all  $K$  meshes on a local scale. The latter is because the translations of the vertices  $\mathbf{X}_1, \dots, \mathbf{X}_K$  that are required to obtain smooth meshes depend on the mesh topology  $\mathcal{F}_r$ , which is *the same* for all meshes. Under the hypothesis that the intrinsic surface properties of all meshes are similar, this leads to globally *and* locally consistent correspondences (see the box titled “Final correspondences” in Fig. 3.1).

**(SP5)** In the previous step not only the desirable global and local consistency of groupwise correspondences is established, but additionally the mesh fairing may have the unwanted effect of altering the objects’ shapes. As compensation, the smoothed meshes are (orthogonally) projected on a per-vertex basis onto the original surface. This projection keeps the correspondences and the mesh regularity intact whilst at the same time recovering the original shape. Let  $\{\bar{\mathcal{M}}_k = (\bar{\mathbf{X}}_k, \mathcal{F}_r)\}_{k=1}^K$  be the set of meshes after the projection onto the original shape surfaces.

The outcome of the described procedure is dependent on the chosen reference mesh  $\mathcal{M}_r$ . However, since the individual subproblems can be solved efficiently, several possible choices for the reference mesh can be evaluated.

Comparable to population-based methods, where compactness of the resulting shapes in  $3N$ -dimensional shape space is optimised, we pursue a similar objective. Rewriting  $\bar{\mathbf{X}}_k \in \mathbb{R}^{N \times 3}$  as column vector  $\mathbf{x}_k \in \mathbb{R}^{3N}$  by stacking all columns, a simple way of measuring compactness in shape space is to use the sum of the distances between each

## Chapter 3. Fast Correspondences for Statistical Shape Models of Brain Structures

---

pair of shape vectors, i.e.  $c_r = \sum_{k \neq k'} \|\mathbf{x}_k - \mathbf{x}_{k'}\|_2$ . Once multiple compactness scores have been evaluated, the reference mesh with index  $r$  is selected that leads to the lowest value  $c_r$ . Whilst for smaller problems it is feasible to perform this procedure for all  $K$  choices as reference, for larger problems only a subset of all reference mesh candidates can be evaluated.

### 3.3 Results

The performance of the presented approach is compared with the particle-based entropy optimisation (PBEO) method by Cates et al. (2006) and the approach by Heimann et al. (2005) based on minimum description length (MDL). The objective is to create a PDM of eight deep brain structures from  $K = 17$  manually created segmentations in multi-modal magnetic resonance images (Bernard et al., 2014). The considered brain structures are *Substantia Nigra & Subthalamic Nucleus* as compound object (256 landmarks); *Nucleus Ruber* (128 landmarks); *Putamen & Globus pallidus* as compound object (256 landmarks); and *Thalamus* (256 landmarks), where all of them are modelled for both the left and right brain hemisphere. For capturing shape variability and inter-relations between the individual brain structures at the same time, a single multi-object PDM with a total number of  $N = 1792$  landmarks is created.

The average distance (AD) and the Hausdorff distance (HD) are used as surface-based distance measures. Evaluated are the following scores: (i) Object reconstruction quality, measuring how accurate the generated object representation approximates the ground truth shape. (ii) Model specificity, measuring the difference between sample shapes drawn from the PDM to the closest training shape (Styner et al., 2003). (iii) Model generalisation, measuring to what extent the PDM can represent unseen instances using leave-one-out cross validation (Styner et al., 2003). (iv) Runtime of the optimisation procedure. In Table 3.1 the results of the evaluation are presented.

	Object Reconstruction		Specificity		Generalisation		Time
	AD	HD	AD	HD	AD	HD	
Proposed	<b>0.93±0.04</b>	2.97±0.27	<b>1.53±0.15</b>	<b>4.89±0.67</b>	<b>1.41±0.13</b>	4.77±0.59	<b>6.07 min</b>
PBEO	0.98±0.04	<b>2.33±0.18</b>	1.54±0.14	5.03±0.71	1.44±0.13	<b>4.74±0.82</b>	182.74 min
MDL	1.30±0.10	8.27±1.34	1.77±0.15	6.76±0.57	1.75±0.14	8.41±1.20	545.74 min

Table 3.1: Various measures for the three evaluated methods. Scores of object reconstruction and generalisation are averaged over  $K = 17$  shapes/runs. Specificity scores are averaged over 1000 draws of shapes from the PDM.

The quantitative evaluation shows that both our method and the PBEO outperform the MDL-based approach. Our method is comparable to PBEO with respect to object reconstruction, specificity and generalisation. However, our method improves significantly upon runtime compared to both other methods. Qualitative results are shown in Figs. 3.2

and 3.3, which both show the brain structures with the camera pointing from inferior to superior.

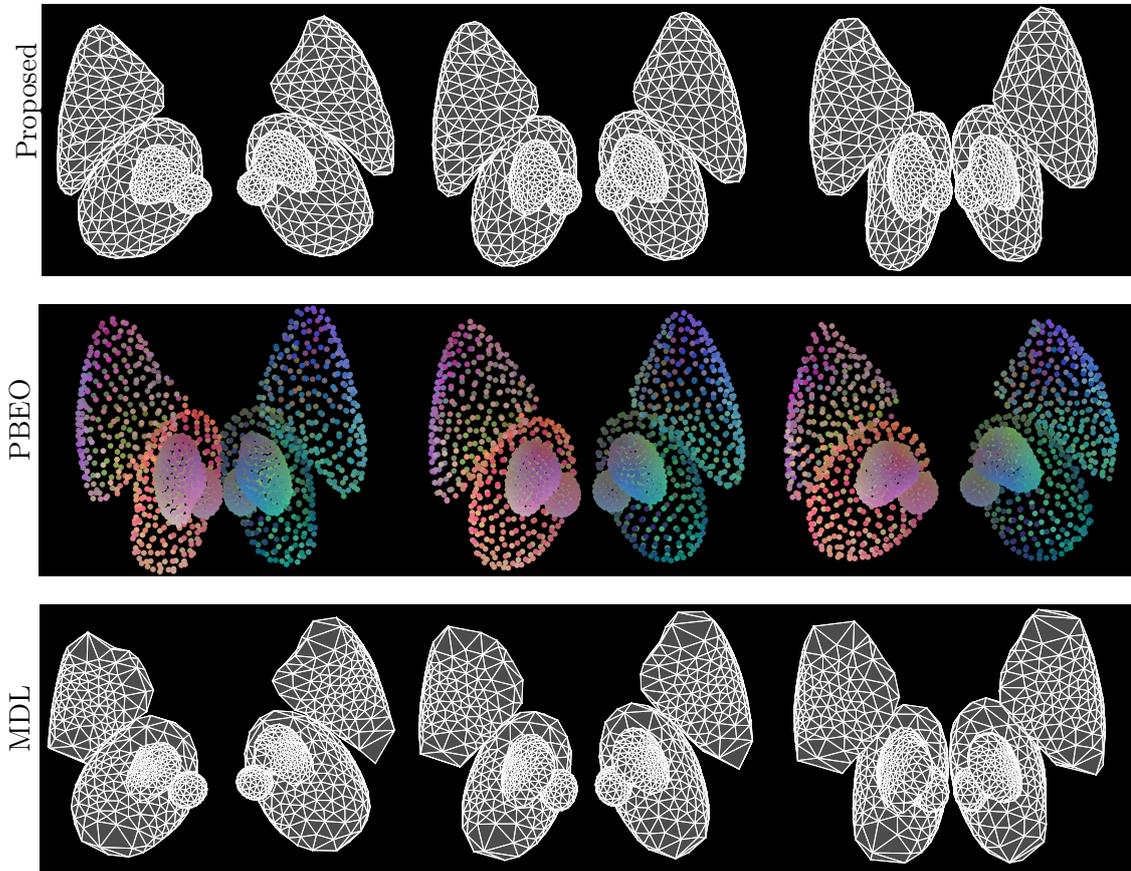


Figure 3.2: First mode of variation of the PDM for the three evaluated methods (rows). The centre column shows the mean shape and the left (right) column shows the deformation at  $-2$  ( $2$ ) times the standard deviation along the first eigenvector. For the particle-based method the colour-coding of the first eigenvector is used to enable a better visualisation (cf. Fig. 3.3). (Best viewed on screen when zoomed in).

In Fig. 3.2, the mean shapes and the first modes of variation with respect to two standard deviations of PCA-based PDMs are depicted. In Fig. 3.3, the deformations for the first three modes are shown by means of colour-coding.

The MDL-based approach has a comparably large mesh reconstruction error (HD), which can be explained by the fact that the surface sampling is not uniform, as can be seen in Figs. 3.2 and 3.3, in the bottom row, respectively. A redistribution of the landmarks on the surface can be used to tackle this problem (Heimann et al., 2006). A flaw of the PBEO method is that the deformations of the resulting PDMs are not (spatially) smooth everywhere. In Fig. 3.3 (middle row), this can be seen by the different colours of the vertices annotated by the red circles, which indicates that the displacement of

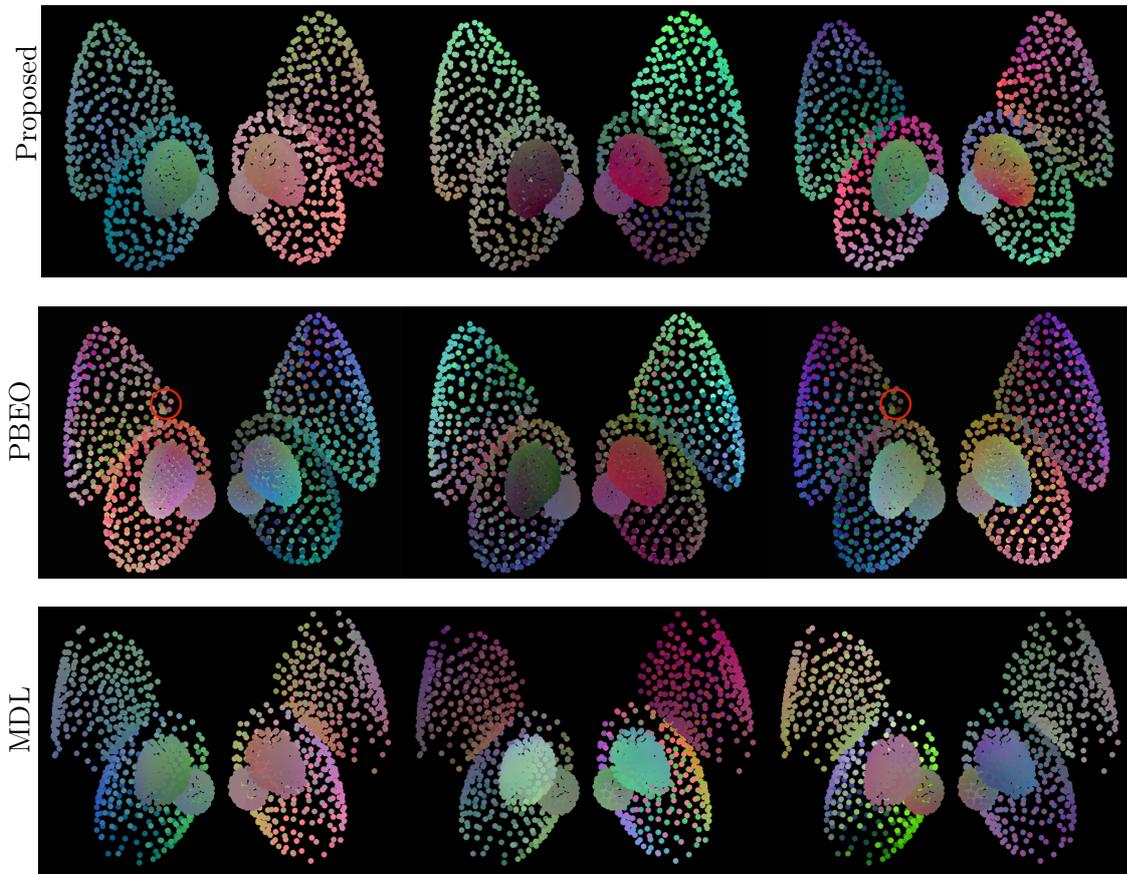


Figure 3.3: Deformations of the PDM for the first 3 eigenvectors (columns) for the three evaluated methods (rows). The displacement for each landmark is colour-coded, where the  $x$ ,  $y$  and  $z$  components of the displacement are linearly mapped to the interval  $[0, 1]$  and then used as  $R$ ,  $G$  and  $B$  values, i.e. colour depicts direction and brightness depicts the amount of the displacement. (Best viewed on screen when zoomed in).

spatially close landmarks during deformation along the first and third eigenvectors is not coherent, i.e. neighbour points move in different directions. Clearly, when speaking of anatomical correspondences, this is not desirable. In contrast, both, the MDL-based method and our proposed method have smooth deformations, which is necessary for not having self-intersections of the meshes during deformation.

### 3.4 Conclusion

We have presented a powerful and efficient method for solving the correspondence problem for the creation of SSMs of brain structures. Our method relies on the well-established algorithms CPD, LAP and mesh fairing, each of them being proven methods on their own. Combining these approaches in the described manner leads to a powerful tool for

the establishment of correspondences.

Whilst being reference-based, an appealing characteristic is that our method is computationally inexpensive and thus the evaluation of multiple reference meshes becomes feasible. The deformable point-cloud registration in combination with the LAP can be seen as initial establishment of globally reliable but locally non-smooth vertex correspondences. The subsequent regularisation using mesh fairing refines the vertex positions such that the correspondences are on the one hand locally smooth and on the other hand consistent among the entire set of shapes. Allowing multiple evaluations of different reference meshes, as well as the mesh fairing procedure being run with the *same* topology for each mesh, both add a hint of population-basedness to our approach.

In our case of eight (comparatively simple) shapes of deep brain structures from 17 subjects we have experimentally demonstrated that our method is preferable with respect to model quality. We claim to have a better model quality compared to MDL since quantitatively our method outperforms MDL as can be seen in Table 3.1. Whilst our method is comparable to the PBEO method with respect to the quantitative scores, qualitatively our method leads to more realistic results due to the smooth deformations. Furthermore, the runtime is reduced significantly, which is crucial since model-building takes in practice multiple runs due to testing, evaluation and optimisation of the model, as pointed out by Heimann and Meinzer (2009).

## Acknowledgements

The present project is supported by the National Research Fund, Luxembourg (6538106, 5748689, 8864515). The authors would like to point out that most of this work was carried out when Nikos Vlassis was with LCSB, Luxembourg.



# 4 Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

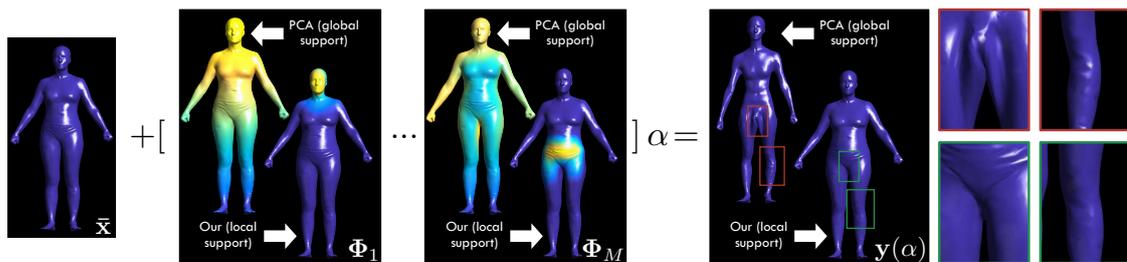


Figure 4.1: Global support factors of PCA lead to implausible body shapes, whereas the local support factors of our method give more realistic results.

## Abstract

Representing 3D shape deformations by high-dimensional linear models has many applications in computer vision and medical imaging. Commonly, using Principal Components Analysis a low-dimensional subspace of the high-dimensional shape space is determined. However, the resulting factors (the most dominant eigenvectors of the covariance matrix) have global support, i.e. changing the coefficient of a single factor deforms the entire shape. Based on matrix factorisation with sparsity and graph-based regularisation terms, we present a method to obtain deformation factors with local support. The benefits include better flexibility and interpretability as well as the possibility of interactively deforming shapes locally. We demonstrate that for brain shapes our method outperforms the state of the art in local support models with respect to generalisation and sparse reconstruction, whereas for body shapes our method gives more realistic deformations.

### 4.1 Introduction

Due to their simplicity, linear models in high-dimensional space are frequently used for modelling nonlinear deformations of shapes in 2D or 3D space. For example, Active Shape Models (ASM) (Cootes and Taylor, 1992), based on a Statistical Shape Model, are popular for image segmentation. Usually, surface meshes, comprising faces and vertices, are employed for representing the surfaces of shapes in 3D. Dimensionality reduction techniques are used to learn a low-dimensional representation of the *vertex coordinates* from training data. Frequently, an affine subspace close to the training shapes is used. To be more specific, mesh deformations are modelled by expressing the vertex coordinates as the sum of a mean shape  $\bar{\mathbf{x}}$  and a linear combination of  $M$  modes of variation  $\Phi = [\Phi_1, \dots, \Phi_M]$ , i.e. the vertices deformed by the weight or coefficient vector  $\alpha$  are given by  $\mathbf{y}(\alpha) = \bar{\mathbf{x}} + \Phi\alpha$ , see Fig. 4.1. Commonly, by using Principal Components Analysis (PCA), the modes of variation are set to the most dominant eigenvectors of the sample covariance matrix. PCA-based models are computationally convenient due to the orthogonality of the eigenvectors of the (real and symmetric) covariance matrix. Due to the diagonalisation of the covariance matrix, an axis-aligned Gaussian distribution of the weight vectors of the training data is obtained. A problem of PCA-based models is that eigenvectors have *global support*, i.e. adjusting the weight of a single factor affects *all* vertices of the shape (Fig. 4.1).

Thus, in this work, instead of eigenvectors, we consider more general *factors* as modes of variation that have *local support*, i.e. adjusting the weight of a single factor leads only to a spatially localised deformation of the shape (Fig. 4.1). The set of all factors can be seen as a dictionary for representing shapes by a linear combination of the factors. Benefits of factors with local support include more realistic deformations (cf. Fig. 4.1), better interpretability of the deformations (e.g. clinical interpretability in a medical context (Sjostrand et al., 2007)), and the possibility of interactive local mesh deformations (e.g. editing animated mesh sequences in computer graphics (Neumann et al., 2013), or enhanced flexibility for interactive 3D segmentation based on Statistical Shape Models (Bernard et al., 2015a, 2016b)).

#### 4.1.1 PCA Variants

PCA is a non-convex problem that admits the efficient computation of the global optimum, e.g. by Singular Value Decomposition (SVD). However, the downside is that the incorporation of arbitrary (convex) regularisation terms is not possible due to the SVD-based solution. Therefore, incorporating regularisation terms into PCA is an active field of research and several variants have been presented: Graph-Laplacian PCA (Jiang et al., 2013) obtains factors with smoothly varying components according to a given

graph. Robust PCA (Candès et al., 2011) formulates PCA as a convex low-rank matrix factorisation problem, where the nuclear norm is used as convex relaxation of the matrix rank. A combination of both Graph-Laplacian PCA and Robust PCA has been presented by Shahid et al. (2015). The Sparse PCA (SPCA) method (Hein and Bühler, 2010; Zou et al., 2006) obtains sparse factors. Structured Sparse PCA (SSPCA) (Jenatton et al., 2010) additionally imposes structure on the sparsity of the factors using group sparsity norms, such as the mixed  $\ell_1/\ell_2$  norm.

#### 4.1.2 Deformation Model Variants

In the work by Last et al. (2011), the flexibility of shape models has been increased by using PCA-based factors in combination with a *per-vertex* weight vector, in contrast to a single weight vector that is used for all vertices. In the works by Cootes and Taylor (1995); Wang and Staib (2000), it is shown that additional elasticity in the PCA-based model can be obtained by manipulation of the sample covariance matrix. Whilst both approaches increase the flexibility of the shape model, they result in global support factors.

In the work by Sjostrand et al. (2007), SPCA is used to model the anatomical shape variation of the 2D outlines of the corpus callosum. In the work by Üzümcü et al. (2003), 2D images of the cardiac ventricle were used to train an Active Appearance Model based on Independent Component Analysis (ICA) (Hyvärinen et al., 2001). Other applications of ICA for Statistical Shape Models are presented in the works by Suinesiaputra et al. (2004); Zewail et al. (2007). The Orthomax method, where the PCA basis is determined first and then rotated such that it has a “simple” structure, is used in the work by Stegmann et al. (2006). The major drawback of SPCA, ICA and Orthomax is that the spatial relation between vertices is completely ignored.

The Key Point Subspace Acceleration method based on Varimax, where a statistical subspace and key points are automatically identified from training data, is introduced in the work by Meyer and Anderson (2007). For mesh animation, in the work by Tena et al. (2011) the clusters of spatially close vertices are determined first by spectral clustering, and then PCA is applied for each vertex cluster, resulting in one sub-PCA model per cluster. This two-stage procedure has the problem, that, due to the independence of both stages, it is unclear whether the clustering is optimal with respect to the deformation model. Also, a blending procedure for the individual sub-PCA models is required. A similar approach of first manually segmenting body regions and then learning a PCA-based model has been presented in the work by Yang et al. (2014).

The *Sparse Localised Deformation Components* method (SPLOCS) obtains localised deformation modes from animated mesh sequences by using a matrix factorisation formulation with a weighted  $\ell_1/\ell_2$  norm regulariser (Neumann et al., 2013). Local support factors

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

---

are obtained by explicitly modelling local support regions, which are in turn used to update the weights of the  $\ell_1/\ell_2$  norm *in each iteration*. This makes the non-convex optimisation problem even harder to solve and dismisses convergence guarantees. With that, a suboptimal initialisation of the support regions, as presented in the work, affects the quality of the found solution.

The *compressed manifold modes* method (Kovnatsky et al., 2015b; Neumann et al., 2014) has the objective to obtain local support basis functions of the (discretised) Laplace-Beltrami operator of a *single* input mesh. In the work by Kovnatsky et al. (2015a), the authors obtain smooth functional correspondences between shapes that are spatially localised by using an  $\ell_1$  norm regulariser in combination with the *row* and *column Dirichlet energy*. The method proposed in the work by Rustamov et al. (2013) is able to localise *shape differences* based on functional maps between two shapes. Recently, the *Shape-from-Operator* approach has been presented (Boscaini et al., 2015), where shapes are reconstructed from more general intrinsic operators.

### 4.1.3 Aims and Main Contributions

The work presented in this paper has the objective of learning local support deformation factors from training data. The main application of the resulting shape model is recognition, segmentation and shape interpolation (Bernard et al., 2015a, 2016b). Whilst our work remedies several of the mentioned shortcomings of existing methods, it can also be seen as complementary to SPLOCS, which is more tailored towards artistic editing and mesh animation. The most significant difference to SPLOCS is that we aim at letting the training shapes *automatically determine the location and size* of each local support region. This is achieved by formulating a matrix factorisation problem that incorporates regularisation terms which simultaneously account for *sparsity* and *smoothness* of the factors, where a graph-based smoothness regulariser accounts for smoothly varying neighbour vertices. In contrast to SPLOCS or sub-PCA, this results in an implicit clustering that is part of the optimisation and does *not require an initialisation* of local support regions, which in turn simplifies the optimisation procedure. Moreover, by integrating a *smoothness prior* into our framework we can *handle small training datasets*, even if the desired number of factors exceeds the number of training shapes. Our optimisation problem is formulated in terms of the Structured Low-Rank Matrix Factorisation framework (Haeffele et al., 2014), which has *appealing theoretical properties*, cf. Section 4.2.4.

## 4.2 Methods

First, we introduce our notation and linear shape deformation models. Then, we state the objective and its formulation as optimisation problem, followed by the theoretical

motivation. Finally, the block coordinate descent algorithm and the factor splitting method are presented.

### 4.2.1 Notation

$\mathbf{I}_p$  denotes the  $p \times p$  identity matrix,  $\mathbf{1}_p$  the  $p$ -dimensional vector containing ones,  $\mathbf{0}_{p \times q}$  the  $p \times q$  zero matrix, and  $\mathbb{S}_p^+$  the set of  $p \times p$  symmetric positive semi-definite matrices. Let  $\mathbf{A} \in \mathbb{R}^{p \times q}$ . We use the notation  $\mathbf{A}_{\mathcal{A}, \mathcal{B}}$  to denote the submatrix of  $\mathbf{A}$  with the rows indexed by the (ordered) index set  $\mathcal{A}$  and columns indexed by the (ordered) index set  $\mathcal{B}$ . The colon denotes the full (ordered) index set, e.g.  $\mathbf{A}_{\mathcal{A}, :}$  is the matrix containing all rows of  $\mathbf{A}$  indexed by  $\mathcal{A}$ . For brevity, we write  $\mathbf{A}_r$  to denote the  $p$ -dimensional vector formed by the  $r$ -th column of  $\mathbf{A}$ . The operator  $\text{vec}(\mathbf{A})$  creates a  $pq$ -dimensional column vector from  $\mathbf{A}$  by stacking its columns, and  $\otimes$  denotes the Kronecker product.

### 4.2.2 Linear Shape Deformation Models

Let  $\mathbf{X}_k \in \mathbb{R}^{N \times 3}$  be the matrix representation of a shape comprising  $N$  points (or vertices) in 3 dimensions, and let  $\{\mathbf{X}_k : 1 \leq k \leq K\}$  be the set of  $K$  training shapes. We assume that the rows in each  $\mathbf{X}_k$  correspond to homologous points. Using the vectorisation  $\mathbf{x}_k = \text{vec}(\mathbf{X}_k) \in \mathbb{R}^{3N}$ , all  $\mathbf{x}_k$  are arranged in the matrix  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K] \in \mathbb{R}^{3N \times K}$ . We assume that all shapes have the same pose, are centred at the mean shape  $\bar{\mathbf{X}}$ , i.e.  $\sum_k \mathbf{X}_k = \mathbf{0}_{N \times 3}$ , and that the standard deviation of  $\text{vec}(\mathbf{X})$  is one.

Pairwise relations between vertices are modelled by a weighted undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\omega})$  that is shared by all shapes. The node set  $\mathcal{V} = \{1, \dots, N\}$  enumerates all  $N$  vertices, the edge set  $\mathcal{E} \subseteq \{1, \dots, N\}^2$  represents the connectivity of the vertices, and  $\boldsymbol{\omega} \in \mathbb{R}_+^{|\mathcal{E}|}$  is the weight vector. The (scalar) weight  $\omega_e$  of edge  $e = (i, j) \in \mathcal{E}$  denotes the affinity between vertex  $i$  and  $j$ , where ‘‘close’’ vertices have high value  $\omega_e$ . We assume there are no self-edges, i.e.  $(i, i) \notin \mathcal{E}$ . The graph can either encode pairwise *spatial* connectivity, or affinities that are not of spatial nature (e.g. symmetries, or prior anatomical knowledge in medical applications).

For the standard PCA-based method (Cootes and Taylor, 1992), the modes of variation in the  $M$  columns of the matrix  $\Phi \in \mathbb{R}^{3N \times M}$  are defined as the  $M$  most dominant eigenvectors of the sample covariance matrix  $\mathbf{C} = \frac{1}{K-1} \mathbf{X} \mathbf{X}^T$ . However, we consider the generalisation where  $\Phi$  contains general  $3N$ -dimensional vectors, the *factors*, in its  $M$  columns. In both cases, the (linear) deformation model (modulo the mean shape) is given by

$$\mathbf{y}(\boldsymbol{\alpha}) = \Phi \boldsymbol{\alpha}, \quad (4.1)$$

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

---

with weight vector  $\boldsymbol{\alpha} \in \mathbb{R}^M$ . Due to vectorisation, the rows with indices  $\{1, \dots, N\}$ ,  $\{N+1, \dots, 2N\}$  and  $\{2N+1, \dots, 3N\}$  of  $\mathbf{y}$  (or  $\boldsymbol{\Phi}$ ), correspond to the  $x$ ,  $y$  and  $z$  components of the  $N$  vertices of the shape, respectively.

### 4.2.3 Objective and Optimisation Problem

The objective is to find  $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1, \dots, \boldsymbol{\Phi}_M]$  and  $\mathbf{A} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_K] \in \mathbb{R}^{M \times K}$  for a given  $M < 3N$  such that, according to eq. (4.1), we can write

$$\mathbf{X} \approx \boldsymbol{\Phi} \mathbf{A}, \quad (4.2)$$

where the factors  $\boldsymbol{\Phi}_m$  have *local support*. Local support means that  $\boldsymbol{\Phi}_m$  is sparse and that all *active* vertices, i.e. vertices that correspond to the non-zero elements of  $\boldsymbol{\Phi}_m$ , are connected by (sequences of) edges in the graph  $\mathcal{G}$ .

Now we state our problem formally as an optimisation problem. The theoretical motivation thereof is based on works by Bach et al. (2008); Burer and Monteiro (2005); Haeffele et al. (2014) and is recapitulated in Section 4.2.4, where it will also become clear that our chosen regularisation term is related to the *Projective Tensor Norm* (Bach et al., 2008; Haeffele et al., 2014).

A general matrix factorisation problem with squared Frobenius norm loss is given by

$$\min_{\boldsymbol{\Phi}, \mathbf{A}} \|\mathbf{X} - \boldsymbol{\Phi} \mathbf{A}\|_F^2 + \Omega(\boldsymbol{\Phi}, \mathbf{A}), \quad (4.3)$$

where the regulariser  $\Omega$  imposes certain properties upon  $\boldsymbol{\Phi}$  and  $\mathbf{A}$ . The optimisation is performed over *some* compact set (which we assume implicitly from here on). An obvious property of local support factors is sparsity. Moreover, it is desirable that neighbour vertices vary smoothly. Both properties together seem to be promising candidates to obtain local support factors, which we reflect in our regulariser. Our optimisation problem is given by

$$\min_{\substack{\boldsymbol{\Phi} \in \mathbb{R}^{3N \times M} \\ \mathbf{A} \in \mathbb{R}^{M \times K}}} \|\mathbf{X} - \boldsymbol{\Phi} \mathbf{A}\|_F^2 + \lambda \sum_{m=1}^M \|\boldsymbol{\Phi}_m\|_{\Phi} \|(\mathbf{A}_{m,:})^T\|_A, \quad (4.4)$$

where  $\|\cdot\|_{\Phi}$  and  $\|\cdot\|_A$  denote vector norms. For  $\mathbf{z}' \in \mathbb{R}^K$ ,  $\mathbf{z} \in \mathbb{R}^{3N}$ , we define

$$\|\mathbf{z}'\|_A = \lambda_A \|\mathbf{z}'\|_2, \text{ and} \quad (4.5)$$

$$\|\mathbf{z}\|_{\Phi} = \lambda_1 \|\mathbf{z}\|_1 + \lambda_2 \|\mathbf{z}\|_2 + \lambda_{\infty} \|\mathbf{z}\|_{1,\infty}^{\mathcal{H}} + \lambda_{\mathcal{G}} \|\mathbf{E}\mathbf{z}\|_2. \quad (4.6)$$

Both  $\ell_2$  norm terms will be discussed in Section 4.2.4. The  $\ell_1$  norm is used to obtain

sparsity in the factors. The (mixed)  $\ell_1/\ell_\infty$  norm is defined by

$$\|\mathbf{z}\|_{1,\infty}^{\mathcal{H}} = \sum_{g \in \mathcal{H}} \|\mathbf{z}_g\|_\infty, \quad (4.7)$$

where  $\mathbf{z}_g$  denotes a subvector of  $\mathbf{z}$  indexed by  $g \in \mathcal{H}$ . By using the collection  $\mathcal{H} = \{\{i, i + N, i + 2N\} : 1 \leq i \leq N\}$ , a grouping of the x, y and z components per vertex is achieved, i.e. within a group  $g$  only the component with largest magnitude is penalised and no extra cost is to be paid for the other components being non-zero.

The last term in eq. (4.6), the graph-based  $\ell_2$  (semi-)norm, imposes smoothness upon each factor, such that neighbour elements according to the graph  $\mathcal{G}$  vary smoothly. Based on the incidence matrix of  $\mathcal{G}$ , we choose  $\mathbf{E}$  such that

$$\|\mathbf{Ez}\|_2 = \sqrt{\sum_{d \in \{0, N, 2N\}} \sum_{(i,j)=e_p \in \mathcal{E}} \omega_{e_p} (\mathbf{z}_{d+i} - \mathbf{z}_{d+j})^2}. \quad (4.8)$$

As such,  $\mathbf{E}$  is a discrete (weighted) gradient operator and  $\|\mathbf{E} \cdot\|_2^2$  corresponds to Graph-Laplacian regularisation (Jiang et al., 2013).  $\mathbf{E}$  is specified in the supplementary material.

The structure of our problem formulation in eqs. (4.4), (4.5), (4.6) allows for various degrees-of-freedom in the form of the parameters. They allow to weigh the data term versus the regulariser ( $\lambda$ ), control the rank of the solution ( $\lambda_A$  and  $\lambda_2$  together, cf. last paragraph in Section 4.2.4), control the sparsity ( $\lambda_1$ ), control the amount of grouping of the x, y and z components ( $\lambda_\infty$ ) and control the smoothness  $\lambda_{\mathcal{G}}$ . The number of factors  $M$  has an impact on the size of the support regions (for small  $M$  the regions tend to be larger, whereas for large  $M$  they tend to be smaller).

#### 4.2.4 Theoretical Motivation

For a matrix  $\mathbf{X} \in \mathbb{R}^{3N \times K}$  and vector norms  $\|\cdot\|_\Phi$  and  $\|\cdot\|_A$ , let us define the function

$$\psi^M(\mathbf{X}) = \min_{\substack{\{\Phi \in \mathbb{R}^{3N \times M}, \\ \mathbf{A} \in \mathbb{R}^{M \times K}: \\ \Phi \mathbf{A} = \mathbf{X}\}}} \sum_{m=1}^M \|\Phi_m\|_\Phi \|(\mathbf{A}_{m,:})^T\|_A. \quad (4.9)$$

The function  $\psi(\cdot) = \lim_{M \rightarrow \infty} \psi^M(\cdot)$  defines a norm known as *Projective Tensor Norm* or *Decomposition Norm* (Bach et al., 2008; Haeffele et al., 2014).

**Lemma 1.** *For any  $\epsilon > 0$  there exists an  $M(\epsilon) \in \mathbb{N}$  such that  $\|\psi(\mathbf{X}) - \psi^{M(\epsilon)}(\mathbf{X})\| < \epsilon$ .*

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

---

*Proof.*

For  $\psi(\mathbf{X})$  there are sequences  $\{\Phi_i\}_{i=1}^{\infty}$  and  $\{\mathbf{A}_i^T\}_{i=1}^{\infty}$  such that  $\psi(\mathbf{X}) = \sum_{i=1}^{\infty} \|\Phi_i\|_{\Phi} \|\mathbf{A}_i^T\|_A$ . Let  $l_m = \sum_{i=1}^m \|\Phi_i\|_{\Phi} \|\mathbf{A}_i^T\|_A$ . The sequence  $l_m$  is monotone, bounded from above and convergent. Let  $l_{\infty} = \psi(\mathbf{X})$  denote its limit. Since the sequence is convergent, there is  $M(\epsilon)$  such that  $\|l_{\infty} - l_j\| < \epsilon$  for  $j \geq M(\epsilon)$ .  $\square$

We now proceed by introducing the optimisation problem

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{Z}\|_F^2 + \lambda \psi^M(\mathbf{Z}). \quad (4.10)$$

Next, we establish a connection between problem (4.10) and our problem (4.4). First, we assume that we are given a solution pair  $(\Phi, \mathbf{A})$  minimising problem (4.4). By defining  $\mathbf{Z} = \Phi \mathbf{A}$ ,  $\mathbf{Z}$  is a solution to problem (4.10). Secondly, assume we are given a solution  $\mathbf{Z}$  minimising problem (4.10). To find a solution pair  $(\Phi, \mathbf{A})$  minimising problem (4.4), one needs to compute the  $(\Phi, \mathbf{A})$  that achieves the minimum of the right-hand side of (4.9) for a given  $\mathbf{Z}$ .

This shows that given a solution to one of the problems, one can infer a solution to the other problem. Next we reformulate problem (4.10). Following Haeffele et al. (2014), we define the matrices  $\mathbf{Q} \in \mathbb{R}^{(3N+K) \times M}$ ,  $\mathbf{Y} \in \mathbb{R}^{(3N+K) \times (3N+K)}$  as

$$\mathbf{Q} = \begin{bmatrix} \Phi \\ \mathbf{A}^T \end{bmatrix}, \quad \mathbf{Y} = \mathbf{Q}\mathbf{Q}^T = \begin{pmatrix} \Phi\Phi^T & \Phi\mathbf{A} \\ \mathbf{A}^T\Phi^T & \mathbf{A}^T\mathbf{A} \end{pmatrix}, \quad (4.11)$$

and the function  $F: \mathbb{S}_{3N+K}^+ \rightarrow \mathbb{R}$  as

$$F(\mathbf{Y}) = F(\mathbf{Q}\mathbf{Q}^T) = \|\mathbf{X} - \Phi\mathbf{A}\|_F^2 + \lambda \psi^M(\Phi\mathbf{A}). \quad (4.12)$$

Let

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y} \in \mathbb{S}_{3N+K}^+} F(\mathbf{Y}). \quad (4.13)$$

For a given  $\mathbf{Y}^*$ , problem (4.10) is minimised by the upper-right block matrix of  $\mathbf{Y}^*$ . The difference between (4.10) and (4.13) is that the latter is over the set of symmetric positive semi-definite matrices, which, at first sight, does not present any gain. However, under certain conditions, the global solution for  $\mathbf{Q}$ , rather than the product  $\mathbf{Y} = \mathbf{Q}\mathbf{Q}^T$ , can be obtained directly (Burer and Monteiro, 2005). Bach et al. (2008) show that if  $\mathbf{Q}$  is a *rank deficient* local minimum of  $F(\mathbf{Q}\mathbf{Q}^T)$ , then it is also a global minimum. Whilst these results only hold for twice differentiable functions  $F$ , Haeffele et al. (2014) have presented analogous results for the case of  $F$  being a sum of a twice-differentiable term and a non-differentiable term, such as ours above.

As such, *any* (rank deficient) local optimum of problem (4.4) is also a global optimum. If in  $\psi(\cdot)$ , both  $\|\cdot\|_{\Phi}$  and  $\|\cdot\|_A$  are the  $\ell_2$  norm,  $\psi(\cdot)$  is equivalent to the nuclear norm, commonly used as convex relaxation of the matrix rank (Haeffele et al., 2014; Recht et al., 2010). In order to steer the solution towards being rank deficient, we include  $\ell_2$  norm terms in  $\|\cdot\|_{\Phi}$  and  $\|\cdot\|_A$  (see (4.5) and (4.6)). With that, part of the regularisation term in (4.4) is the nuclear norm that accounts for low-rank solutions.

### 4.2.5 Block Coordinate Descent

A solution to problem (4.4) is found by block coordinate descent (BCD) (Xu and Yin, 2013) (algorithm 1). It employs alternating proximal steps, which can be seen as generalisation of gradient steps for non-differentiable functions. Since computing the proximal mapping

```

1 repeat
2   // update  $\Phi$ 
3    $\Phi' \leftarrow \Phi - \epsilon_{\Phi} \nabla_{\Phi} \|\mathbf{X} - \Phi \mathbf{A}\|_F^2$  // gradient step (loss)
4   for  $m = 1, \dots, M$  do // proximal step  $\Phi$  (penalty)
5      $\Phi_m \leftarrow \text{prox}_{\lambda \|\cdot\|_{\Phi}} (\mathbf{A}_{m,:})^T (\Phi'_m)$ 
6   // update  $\mathbf{A}$ 
7    $\mathbf{A}' \leftarrow \mathbf{A} - \epsilon_A \nabla_A \|\mathbf{X} - \Phi \mathbf{A}\|_F^2$  // gradient step (loss)
8   for  $m = 1, \dots, M$  do // proximal step  $\mathbf{A}$  (penalty)
9      $\mathbf{A}_{m,:} \leftarrow \text{prox}_{\lambda \|\Phi_m\|_{\Phi} \|\cdot\|_A} ((\mathbf{A}'_{m,:})^T)^T$ 
10  until convergence

```

**Algorithm 1:** Simplified view of block coordinate descent. For details see (Haeffele et al., 2014; Xu and Yin, 2013).

is repeated in each iteration, an efficient computation is essential. The proximal mapping of  $\|\cdot\|_A$  in eq. (4.5) has a closed-form solution by *block soft thresholding* (Parikh and Boyd, 2013). The proximal mapping of  $\|\cdot\|_{\Phi}$  in eq. (4.6) is solved by dual forward-backward splitting (Combettes et al., 2010; Combettes and Pesquet, 2011) (see supplementary material). The benefit of BCD is that it scales well to large problems (cf. complexity analysis in the supplementary material). However, a downside is that by using the alternating updates one has only guaranteed convergence to a Nash equilibrium point (Haeffele et al., 2014; Xu and Yin, 2013).

### 4.2.6 Factor Splitting

Whilst solving problem (4.4) leads to smooth and sparse factors, there is no guarantee that the factors have only a *single* local support region. In fact, as motivated in Section 4.2.4, the solution of eq. (4.4) is steered towards being low-rank. However, the price to pay for a low-rank solution is that capturing multiple support regions in a *single* factor is preferred over having each support region in an individual factor (e.g. for  $M = 2$  and any  $\mathbf{a} \neq \mathbf{0}$ ,  $\mathbf{b} \neq \mathbf{0}$ , the matrix  $\Phi = [\Phi_1 \Phi_2]$  has a lower rank when  $\Phi_1 = [\mathbf{a}^T \mathbf{b}^T]^T$  and  $\Phi_2 = \mathbf{0}$ , compared to  $\Phi_1 = [\mathbf{a}^T \mathbf{0}^T]^T$  and  $\Phi_2 = [\mathbf{0}^T \mathbf{b}^T]^T$ ).

A simple yet effective way to deal with this issue is to split factors with multiple disconnected support regions into multiple (new) factors (see supplementary material). Since this is performed *after* the optimisation problem has been solved, it is preferable over pre- or intra-processing procedures (Tena et al., 2011; Neumann et al., 2013) since the optimisation remains unaffected and the data term in eq. (4.4) does not change due to the splitting.

### 4.3 Experimental Results

We compared the proposed method with *PCA* (Cootes and Taylor, 1992), *kernel PCA* (kPCA, cf. 4.3.2), *Varimax* (Harman, 1976), *ICA* (Hyvärinen et al., 2001), *SPCA* (Jenatton et al., 2010), *SSPCA* (Jenatton et al., 2010), and *SPLOCS* (Neumann et al., 2013) on two real datasets, brain structures and human body shapes. Only our method and the SPLOCS method explicitly aim to obtain local support factors, whereas the SPCA and SSPCA methods obtain sparse factors (for the latter the  $\ell_1/\ell_2$  norm with groups defined by  $\mathcal{H}$ , cf. eq. (4.7), is used). The methods kPCA, SPCA, SSPCA, SPLOCS and ours require to set various parameters, which we address by random sampling (see supplementary material).

For all evaluated methods a factorisation  $\Phi\mathbf{A}$  is obtained. W.l.o.g. we normalise the rows of  $\mathbf{A}$  to have standard deviation one (since  $\Phi\mathbf{A} = (\frac{1}{s}\Phi)(s\mathbf{A})$  for  $s \neq 0$ ). Then, the factors in  $\Phi$  are ordered descendingly according to their  $\ell_2$  norms.

In our method, the number of factors may be changed due to factor splitting, thus, in order to allow a fair comparison, we only retain the first  $M$  factors. Initially, the columns of  $\Phi$  are chosen to  $M$  (unique) columns selected randomly from  $\mathbf{I}_{3N}$ . This is in accordance with the observations by Haeffele et al. (2014), where empirically good results are obtained using trivial initialisations. Convergence plots for different initialisations can be found in the supplementary material.

#### 4.3.1 Quantitative Measures

For  $\mathbf{x} = \text{vec}(\mathbf{X})$  and  $\tilde{\mathbf{x}} = \text{vec}(\tilde{\mathbf{X}})$ , the *average error*

$$e_{\text{avg}}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{X}_{i,:} - \tilde{\mathbf{X}}_{i,:}\|_2 \quad (4.14)$$

and the *maximum error*

$$e_{\text{max}}(\mathbf{x}, \tilde{\mathbf{x}}) = \max_{i=1,\dots,N} \|\mathbf{X}_{i,:} - \tilde{\mathbf{X}}_{i,:}\|_2 \quad (4.15)$$

measure the agreement between shape  $\mathbf{X}$  and shape  $\tilde{\mathbf{X}}$ .

The *reconstruction error* for shape  $\mathbf{x}_k$  is measured by solving the overdetermined system  $\Phi\alpha_k = \mathbf{x}_k$  for  $\alpha_k$  in the least-squares sense, and then computing  $e_{\text{avg}}(\mathbf{x}_k, \Phi\alpha_k)$  and  $e_{\text{max}}(\mathbf{x}_k, \Phi\alpha_k)$ , respectively.

To measure the *specificity error*,  $n_s$  shape samples are drawn randomly ( $n_s = 1000$  for the brain shapes and  $n_s = 100$  for the body shapes). For each drawn shape, the average and maximum errors between the closest shape in the training set are denoted by  $s_{\text{avg}}$  and  $s_{\text{max}}$ , respectively. For simplicity, we assumed that the parameter vector  $\alpha$  follows a zero mean Gaussian distribution, where the covariance matrix  $\mathbf{C}_\alpha$  is estimated from the parameter vectors  $\alpha_k$  of the  $K$  training shapes. With that, a random shape sample  $\mathbf{x}_r$  is generated by drawing a sample of the vector  $\alpha_r$  from its distribution and setting  $\mathbf{x}_r = \Phi\alpha_r$ . The specificity can be interpreted as a score for assessing how realistic synthesized shapes are on a coarse level of detail (the contribution of errors on fine details to the specificity score is marginal due to the dominance of the errors on coarse scales).

For evaluating the *generalisation error*, a collection of index sets  $\mathcal{I} \subset 2^{\{1, \dots, K\}}$  is used, where each set  $\mathcal{J} \in \mathcal{I}$  denotes the set of indices of the *test shapes* for one run and  $|\mathcal{I}|$  is the number of runs. We used five-fold cross-validation, i.e.  $|\mathcal{I}| = 5$  and each set  $\mathcal{J}$  contains  $\text{round}(\frac{K}{5})$  random integers from  $\{1, \dots, K\}$ . In each run, the deformation factors  $\Phi^{\bar{\mathcal{J}}}$  are computed using only the shapes with indices  $\bar{\mathcal{J}} = \{1, \dots, K\} \setminus \mathcal{J}$ . For all test shapes  $\mathbf{x}_j$ , where  $j \in \mathcal{J}$ , the parameter vector  $\alpha_j$  is determined by solving  $\Phi^{\bar{\mathcal{J}}}\alpha_j = \mathbf{x}_j$  in the least-squares sense. Eventually, the average reconstruction error  $e_{\text{avg}}(\mathbf{x}_j, \Phi^{\bar{\mathcal{J}}}\alpha_j)$  and the maximum reconstruction error  $e_{\text{max}}(\mathbf{x}_j, \Phi^{\bar{\mathcal{J}}}\alpha_j)$  are computed for each test shape, which we denote as  $g_{\text{avg}}$  and  $g_{\text{max}}$ , respectively. Moreover, the *sparse reconstruction errors*  $g_{\text{avg}}^{0.05}$  and  $g_{\text{max}}^{0.05}$  are computed in a similar manner, with the difference that  $\alpha_j$  is now determined by using only 5% of the rows (selected randomly) of  $\Phi^{\bar{\mathcal{J}}}$  and  $\mathbf{x}_j$ , denoted by  $\tilde{\Phi}^{\bar{\mathcal{J}}}$  and  $\tilde{\mathbf{x}}_j$ . For that, we minimise  $\|\tilde{\Phi}^{\bar{\mathcal{J}}}\alpha_j - \tilde{\mathbf{x}}_j\|_2^2 + \|\Gamma\alpha_j\|_2^2$  with respect to  $\alpha_j$ , which is a least-squares problem with Tikhonov regularisation, where  $\Gamma$  is obtained by Cholesky factorisation of  $\mathbf{C}_\alpha = \Gamma^T\Gamma$ . The purpose of this measure is to evaluate how well a deformation model interpolates an unseen shape from a small subset of its points, which is relevant for shape model-based surface reconstruction (Bernard et al., 2015a, 2016b).

### 4.3.2 Brain Structures

The first evaluated dataset comprises 8 brain structure meshes of  $K = 17$  subjects (Bernard et al., 2016c). All 8 meshes together have a total number of  $N = 1792$  vertices that are in correspondence among all subjects. Moreover, all meshes have the same topology, i.e. seen as graphs they are isomorphic. A single deformation model is used to model the deformation of all 8 meshes in order to capture the interrelation between the brain structures. We fix the number of desired factors to  $M = 96$  to account for a sufficient amount of local details in the factors. Whilst the meshes are smooth and comparably

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

simple in shape (cf. Fig. 4.3), a particular challenge is that the training dataset comprises only  $K = 17$  shapes.

### Second-order Terms

Based on anatomical knowledge, we use the brain structure interrelation graph  $\mathcal{G}_{\text{bs}} = (\mathcal{V}_{\text{bs}}, \mathcal{E}_{\text{bs}})$  shown in Fig. 4.2, where an edge between two structures denotes that a deformation of one structure may have a direct effect on the deformation of the other structure. Using  $\mathcal{G}_{\text{bs}}$ , we now build a distance matrix that is then used in the SPLOCS method

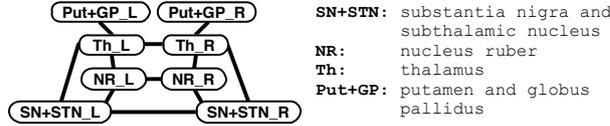


Figure 4.2: Brain structure interrelation graph.

and our method. For  $\mathfrak{o} \in \mathcal{V}_{\text{bs}}$ , let  $g_{\mathfrak{o}} \subset \{1, \dots, N\}$  denote the (ordered) indices of the vertices of brain structure  $\mathfrak{o}$ . Let  $\mathbf{D}_{\text{euc}} \in \mathbb{R}^{N \times N}$  be the Euclidean distance matrix, where  $(\mathbf{D}_{\text{euc}})_{ij} = \|\bar{\mathbf{X}}_{i,:} - \bar{\mathbf{X}}_{j,:}\|_2$  is the Euclidean distance between vertex  $i$  and  $j$  of the mean shape  $\bar{\mathbf{X}}$ . Moreover, let  $\mathbf{D}_{\text{geo}} \in \mathbb{R}^{N \times N}$  be the geodesic graph distance matrix of the mean shape  $\bar{\mathbf{X}}$  using the graph induced by the (shared) mesh topology. By  $\mathbf{D}_{\text{euc}}^{\mathfrak{o}} \in \mathbb{R}^{|g_{\mathfrak{o}}| \times |g_{\mathfrak{o}}|}$  and  $\mathbf{D}_{\text{geo}}^{\mathfrak{o}} \in \mathbb{R}^{|g_{\mathfrak{o}}| \times |g_{\mathfrak{o}}|}$  we denote the Euclidean distance matrix and the geodesic distance matrix of brain structure  $\mathfrak{o}$ , which are submatrices of  $\mathbf{D}_{\text{euc}}$  and  $\mathbf{D}_{\text{geo}}$ , respectively. Let  $\bar{d}^{\mathfrak{o}}$  denote the average vertex distance between neighbour vertices of brain structure  $\mathfrak{o}$ . We define the normalised geodesic graph distance matrix of brain structure  $\mathfrak{o}$  by  $\tilde{\mathbf{D}}_{\text{geo}}^{\mathfrak{o}} = \frac{1}{\bar{d}^{\mathfrak{o}}} \mathbf{D}_{\text{geo}}^{\mathfrak{o}}$  and the matrix  $\tilde{\mathbf{D}}_{\text{geo}} \in \mathbb{R}^{N \times N}$  is composed by the individual blocks  $\tilde{\mathbf{D}}_{\text{geo}}^{\mathfrak{o}}$ .

The normalised distance matrix between structure  $\mathfrak{o}$  and  $\tilde{\mathfrak{o}}$  is given by

$$\tilde{\mathbf{D}}_{\text{bs}}^{\mathfrak{o}, \tilde{\mathfrak{o}}} = \frac{2}{\bar{d}^{\mathfrak{o}} + \bar{d}^{\tilde{\mathfrak{o}}}} [(\mathbf{D}_{\text{euc}})_{g_{\mathfrak{o}}, g_{\tilde{\mathfrak{o}}}} - \mathbf{1}_{|g_{\mathfrak{o}}|} \mathbf{1}_{|g_{\tilde{\mathfrak{o}}}|}^T d_{\min}^{\mathfrak{o}, \tilde{\mathfrak{o}}}] \in \mathbb{R}^{|g_{\mathfrak{o}}| \times |g_{\tilde{\mathfrak{o}}}|}, \quad (4.16)$$

where  $d_{\min}^{\mathfrak{o}, \tilde{\mathfrak{o}}}$  is the smallest element of  $(\mathbf{D}_{\text{euc}})_{g_{\mathfrak{o}}, g_{\tilde{\mathfrak{o}}}}$ . The (symmetric) distance matrix  $\tilde{\mathbf{D}}_{\text{bs}} \in \mathbb{R}^{N \times N}$  between all structures is constructed by

$$(\tilde{\mathbf{D}}_{\text{bs}})_{g_{\mathfrak{o}}, g_{\tilde{\mathfrak{o}}}} = \begin{cases} \tilde{\mathbf{D}}_{\text{bs}}^{\mathfrak{o}, \tilde{\mathfrak{o}}} & \text{if } (\mathfrak{o}, \tilde{\mathfrak{o}}) \in \mathcal{E}_{\text{bs}} \\ \mathbf{0}_{|g_{\mathfrak{o}}| \times |g_{\tilde{\mathfrak{o}}}|} & \text{else} \end{cases}. \quad (4.17)$$

For the SPLOCS method we used the distance matrix  $\mathbf{D} = \alpha_D \tilde{\mathbf{D}}_{\text{geo}} + (1 - \alpha_D) \tilde{\mathbf{D}}_{\text{bs}}$ . For our method, we construct the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$  (cf. Section 4.2.2) by having an edge  $e = (i, j)$  in  $\mathcal{E}$  for each  $\omega_e = \alpha_D \exp(-(\tilde{\mathbf{D}}_{\text{geo}})_{ij}^2) + (1 - \alpha_D) \exp(-(\tilde{\mathbf{D}}_{\text{bs}})_{ij}^2)$  that is larger than the threshold  $\theta = 0.1$ . In both cases we set  $\alpha_D = 0.5$ .

### Dealing with the Small Training Set

For PCA, Varimax and ICA the number of factors cannot exceed the rank of  $\mathbf{X}$ , which is at most  $K - 1$  for  $K < 3N$ . For the used matrix factorisation framework, setting  $M$  to a value larger than the expected rank of the solution but smaller than full rank has empirically led to good results (Haeffele et al., 2014). However, since our expected rank is  $M = 96$  and the full rank is at most  $K - 1 = 16$ , this is not possible.

We compensate the insufficient amount of training data by assuming smoothness of the deformations. Based on concepts introduced by Cootes and Taylor (1995); Wang and Staib (1998, 2000), instead of the data matrix  $\mathbf{X}$ , we factorise the *kernelised covariance* matrix  $\mathbf{K}$ . The standard PCA method finds the  $M$  most dominant eigenvectors of the covariance matrix  $\mathbf{C}$  by the (exact) factorisation  $\mathbf{C} = \Phi \text{diag}(\lambda_1, \dots, \lambda_{K-1}) \Phi^T$ . The kernelised covariance  $\mathbf{K}$  allows to incorporate additional elasticity into the resulting deformation model. For example,  $\mathbf{K} = \mathbf{I}_{3N}$  results in independent vertex movements (Wang and Staib, 2000). A more interesting approach is to combine the (scaled) covariance matrix with a smooth vertex deformation. We define  $\mathbf{K} = \frac{1}{\|\text{vec}(\mathbf{X}\mathbf{X}^T)\|_\infty} \mathbf{X}\mathbf{X}^T + \mathbf{K}_{\text{euc}}$ , with  $\mathbf{K}_{\text{euc}} = \mathbf{I}_3 \otimes \mathbf{K}'_{\text{euc}} \in \mathbb{R}^{3N \times 3N}$ . Using the bandwidth  $\beta$ ,  $\mathbf{K}'_{\text{euc}}$  is given by

$$(\mathbf{K}'_{\text{euc}})_{ij} = \exp\left(-\left(\frac{(\mathbf{D}_{\text{euc}})_{ij}}{2\beta\|\text{vec}(\mathbf{D}_{\text{euc}})\|_\infty}\right)^2\right). \quad (4.18)$$

Setting  $\Phi$  to the  $M$  most dominant eigenvectors of the symmetric and positive semi-definite matrix  $\mathbf{K}$  gives the solution of kPCA (Bishop, 2006). In terms of our proposed matrix factorisation problem in eq. (4.4), we find a factorisation  $\Phi \tilde{\mathbf{A}}$  of  $\mathbf{K}$ , instead of factorising the data  $\mathbf{X}$ . Since the regularisation term remains unchanged, the factor matrix  $\Phi \in \mathbb{R}^{3N \times M}$  is still sparse and smooth (due to  $\|\cdot\|_\Phi$ ). Moreover, due to the nuclear norm term being contained in the product  $\|\cdot\|_\Phi \cdot \|\cdot\|_A$  (cf. Section 4.2.4), the resulting factorisation  $\Phi \tilde{\mathbf{A}}$  is steered towards being low-rank, in favour of the elaborations in Section 4.2.4. However, the resulting matrix  $\tilde{\mathbf{A}} \in \mathbb{R}^{M \times 3N}$  now contains the weights for approximating  $\mathbf{K}$  by a linear combination of the columns of  $\Phi$ , rather than approximating the data matrix  $\mathbf{X}$  by a linear combination of the columns of  $\Phi$ . For the known  $\Phi$ , the weights that best approximate the *data matrix*  $\mathbf{X}$  are found by solving the linear system  $\Phi \mathbf{A} = \mathbf{X}$  in the least-squares sense for  $\mathbf{A} \in \mathbb{R}^{M \times K}$ .

### Results

The magnitude of the deformation of the first three factors are shown in Fig. 4.3, where it can be seen that only SPCA, SSPCA, SPLOCS and our method obtain *sparse* deformation factors. The SPCA and SSPCA methods do not incorporate the spatial relation between vertices and as such the deformations are not spatially localised (see red arrows in Fig. 4.3, where more than a single region is active). The factors obtained by SPLOCS are non-smooth and do not exhibit local support, in contrast to our method, where smooth

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

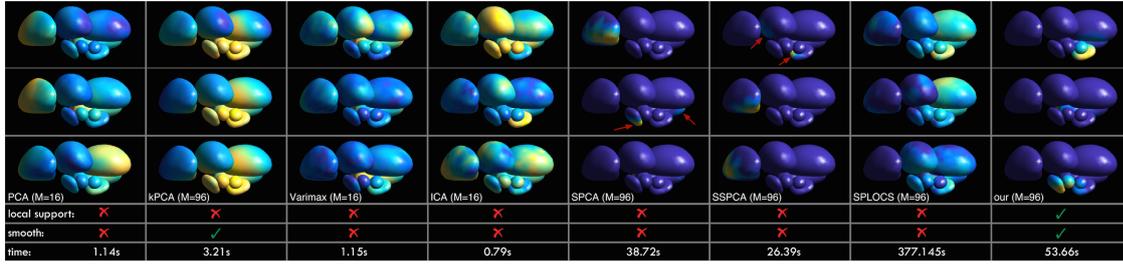


Figure 4.3: The colour-coded magnitude (blue corresponds to zero, yellow to the maximum deformation in each plot) for the three deformation factors with largest  $\ell_2$  norm is shown in the rows. The factors obtained by SPCA and SSPCA are sparse but not spatially localised (see red arrows). Our method is the only one that obtains local support factors.

deformation factors with local support are obtained.

The quantitative results presented in Fig. 4.4 reveal that our method has a larger reconstruction error. This can be explained by the fact that due to the sparsity and smoothness of the deformation factors a very large number of basis vectors is required in order to exactly span the subspace of the training data. Instead, our method finds a simple (sparse and smooth) representation that explains the data approximately, in favour of Occam’s razor. The average reconstruction error is around 1mm, which is low considering that the brain structures span approximately 6cm from left to right. Considering specificity, all methods are comparable. PCA, Varimax and ICA, which have the lowest reconstruction errors, have the highest generalisation errors, which underlines that these methods overfit the training data. The kPCA method is able to overcome this issue due to the smoothness assumption. SPCA and SSPCA have good generalisation scores but at the same time a very high maximum reconstruction error. Our method and SPLOCS are the only ones that explicitly strive for local support factors. Since our method outperforms SPLOCS with respect to generalisation and sparse reconstruction error, we claim that our method outperforms the state of the art.

### 4.3. Experimental Results

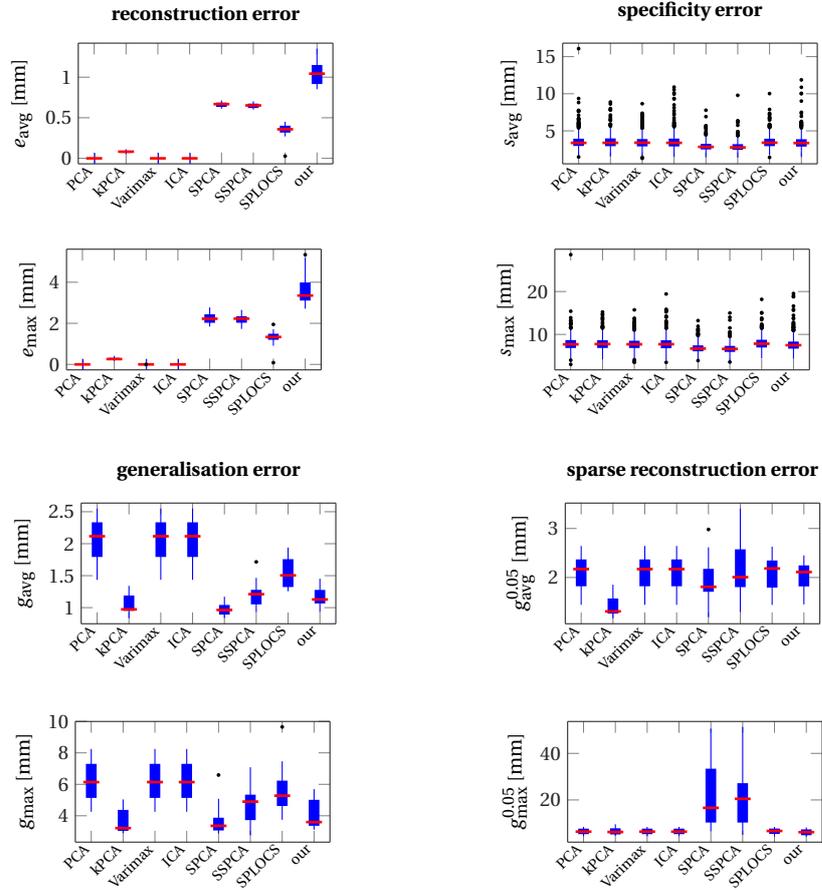


Figure 4.4: Boxplots of the quantitative measures for the brain shapes dataset. In each plot, the horizontal axis shows the individual methods and the vertical axis the error scores described in Section 4.3.1. Compared to SPLOCS, which is the only other method explicitly striving for local support deformation factors, our method has a smaller generalisation error and sparse reconstruction error. The sparse but not spatially localised factors obtained by SPCA and SSPCA (cf. Fig. 4.3) result in a large maximum sparse reconstruction error (bottom right).

### 4.3.3 Human Body Shapes

Our second experiment is based on 1531 female human body shapes (Yang et al., 2014), where each shape comprises 12500 vertices that are in correspondence among the training set. Due to the large number of training data and the high level of details in the meshes, we directly factorise the data matrix  $\mathbf{X}$ . The edge set  $\mathcal{E}$  now contains the edges of the triangle mesh topology and the weights for edge  $e = (i, j) \in \mathcal{E}$  are given by  $\omega_e = \exp(-(\frac{\mathbf{D}_{\text{euc}}}{\bar{d}})_{ij}^2)$ , where  $\bar{d}$  denotes the average vertex distance between neighbour vertices. Edges with weights lower than  $\theta = 0.1$  are ignored.

#### Results

Quantitatively the evaluated methods have comparable performance, with the exception that ICA has worse overall performance (Fig. 4.5). The most noticeable difference between the methods is the specificity error, where SPCA, SSPCA and our method perform best. Fig. 4.6 reveals that SPCA, SSPCA, SPLOCS and our method obtain factors with local support. Apparently, for large datasets, sparsity alone, as used in SPCA and SSPCA, is sufficient to obtain local support factors. However, our method is the only one that explicitly aims for smoothness of the factors, which leads to more realistic deformations, as shown in Fig. 4.7.

### 4.3. Experimental Results

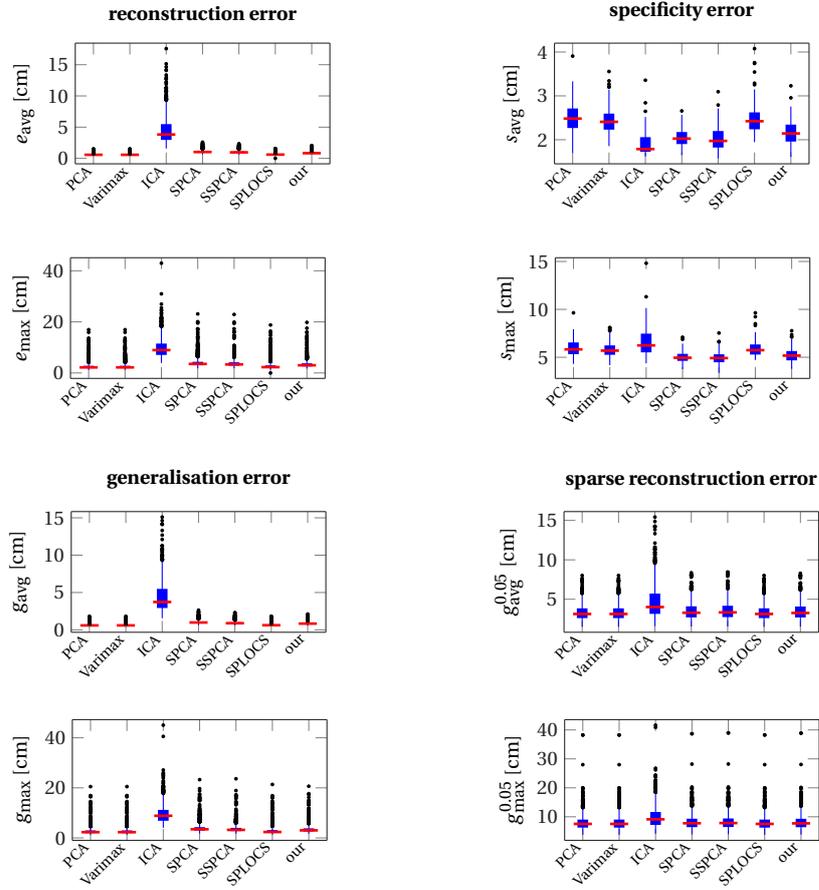


Figure 4.5: Boxplots of the quantitative measures in each column for the body shapes dataset. Quantitatively all methods have comparable performance, apart from ICA which performs worse.

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

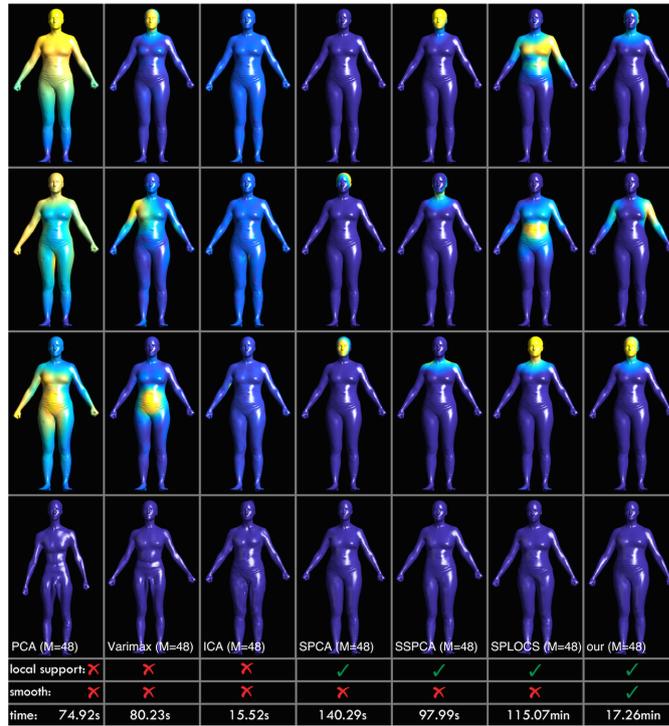


Figure 4.6: The deformation magnitudes reveal that SPCA, SSPCA, SPLOCS and our method obtain local support factors (in the second factor of our method the connection is at the back). The bottom row depicts randomly drawn shapes, where only the methods with local support deformation factors result in plausible shapes.

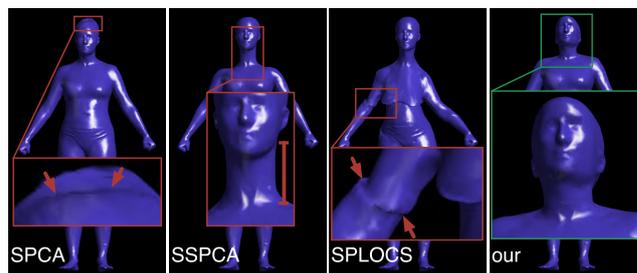


Figure 4.7: Shapes  $\bar{x} - 1.5\Phi_m$  for SPCA ( $m = 1$ ), SSPCA ( $m = 3$ ), SPLOCS ( $m = 1$ ) and our ( $m = 1$ ) method (cf. Fig. 4.6). Our method delivers the most realistic per-factor deformations.

## 4.4 Conclusion

We presented a novel approach for learning a linear deformation model from training shapes, where the resulting factors exhibit local support. By embedding sparsity and smoothness regularisers into a theoretically well-grounded matrix factorisation framework, we model local support regions *implicitly*, and thus get rid of the initialisation of the size and location of local support regions, which so far has been necessary in existing methods. On the small brain shapes dataset that contains relatively simple shapes, our method improves the state of the art with respect to generalisation and sparse reconstruction. For the large body shapes dataset containing more complex shapes, quantitatively our method is on par with existing methods, whilst it delivers more realistic per-factor deformations. Since articulated motions violate our smoothness assumption, our method cannot handle them. However, when smooth deformations are a reasonable assumption, our method offers a higher flexibility and better interpretability compared to existing methods, whilst at the same time delivering more realistic deformations.

### Acknowledgements

We thank Yipin Yang and colleagues for making the human body shapes dataset publicly available; Benjamin D. Haeffele and René Vidal for providing their code; Thomas Bühler and Daniel Cremers for helpful comments on the manuscript; Luis Salamanca for helping to improve our figures; and Michel Antunes and Djamila Aouada for pointing out relevant literature. The authors gratefully acknowledge the financial support by the Fonds National de la Recherche, Luxembourg (6538106, 8864515).

## 4.5 Supplementary Material

### 4.5.1 The matrix $\mathbf{E}$ in eq. (4.8)

The matrix  $\mathbf{E} \in \mathbb{R}^{3|\mathcal{E}| \times 3N}$  is defined as  $\mathbf{E} = \mathbf{I}_3 \otimes \mathbf{E}'$ , where  $\mathbf{E}' \in \mathbb{R}^{|\mathcal{E}| \times N}$  is the weighted incidence matrix of the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\omega})$  with elements

$$\mathbf{E}'_{pq} = \begin{cases} \sqrt{\boldsymbol{\omega}_{e_p}} & \text{if } q = i \text{ for } e_p = (i, j) \\ -\sqrt{\boldsymbol{\omega}_{e_p}} & \text{if } q = j \text{ for } e_p = (i, j) \\ 0 & \text{otherwise.} \end{cases} \quad (4.19)$$

For  $p = 1, \dots, |\mathcal{E}|$ ,  $e_p \in \mathcal{E}$  denotes the  $p$ -th edge.

### 4.5.2 Proximal Operators

**Definition 1.** The proximal operator (or proximal mapping)  $\text{prox}_{s\theta}(y) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  of a lower semicontinuous function  $\theta : \mathbb{R}^n \rightarrow \mathbb{R}$  scaled by  $s > 0$  is defined by

$$\text{prox}_{s\theta}(\mathbf{y}) = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2s} \|\mathbf{x} - \mathbf{y}\|_2^2 + \theta(\mathbf{x}) \right\}. \quad (4.20)$$

#### Proximal Operator of $\|\cdot\|_A$

The proximal operator  $\text{prox}_{\|\cdot\|_A}(\mathbf{y})$  of

$$\|\cdot\|_A = \lambda_A \|\cdot\|_2,$$

cf. eq. (4.5), can be computed by the so-called *block soft thresholding* (Parikh and Boyd, 2013), i.e.

$$\text{prox}_{\lambda_A \|\cdot\|_2}(\mathbf{y}) = (1 - \lambda_A / \|\mathbf{y}\|_2)_+ \mathbf{y}, \quad (4.21)$$

where for a vector  $\mathbf{x} \in \mathbb{R}^p$ ,  $(\mathbf{x})_+$  replaces each negative element in  $\mathbf{x}$  with 0. As such, a very efficient way for computing the proximal mapping of  $\|\cdot\|_A$  is available.

#### Proximal Operator of $\|\cdot\|_\Phi$

The proximal mapping of

$$\|\cdot\|_\Phi = \lambda_1 \|\cdot\|_1 + \lambda_2 \|\cdot\|_2 + \lambda_\infty \|\cdot\|_{1,\infty}^{\mathcal{H}} + \lambda_G \|\mathbf{E} \cdot\|_2,$$

cf. eq. (4.6), is given by

$$\text{prox}_{\|\cdot\|_\Phi}(\mathbf{z}) = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{x}\|_2 + \lambda_\infty \|\mathbf{x}\|_{1,\infty}^{\mathcal{H}} + \lambda_G \|\mathbf{E}\mathbf{x}\|_2 \right\}. \quad (4.22)$$

Due to the non-separability caused by the linear mapping  $\mathbf{E}$  inside the  $\ell_2$  norm, this case is more difficult compared to  $\text{prox}_{\|\cdot\|_A}(\mathbf{y})$ . However, by introducing

$$f(\mathbf{x}) = \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{x}\|_2 + \lambda_\infty \|\mathbf{x}\|_{1,\infty}^{\mathcal{H}} \quad (4.23)$$

and

$$g(\mathbf{x}) = \lambda_G \|\mathbf{x}\|_2, \quad (4.24)$$

eq. (4.22) can be rewritten as

$$\arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + f(\mathbf{x}) + g(\mathbf{E}\mathbf{x}) \right\}. \quad (4.25)$$

In this form, (4.25) can now be solved by a dual forward-backward splitting procedure (Combettes et al., 2010; Combettes and Pesquet, 2011) as shown in algorithm 2, which we will explain in the rest of this section. The efficient computability hinges on the efficient computation of the (individual) proximal mappings of  $f$  and  $g$ .

Since  $g$  is a (weighted)  $\ell_2$  norm, its proximal mapping is given by *block soft thresholding*

**Input:**  $\mathbf{z} \in \mathbb{R}^{3N}$   
**Output:**  $\mathbf{x} = \text{prox}_{\|\cdot\|_{\Phi}}(\mathbf{z})$   
**Parameters:**  $\lambda, \lambda_1, \lambda_{\infty}, \lambda_2, \lambda_{\mathcal{G}}$   
**Initialise:**  $\mathbf{v}, \epsilon \leftarrow 1e^{-4}, \gamma \leftarrow 1.999, \beta = \frac{1+\epsilon}{2}$   
 // Normalise  $\mathbf{E}$  (homogeneity of norm)  
 1  $\lambda_{\mathcal{G}} \leftarrow \lambda_{\mathcal{G}} \|\mathbf{E}\|_F; \quad \mathbf{E} \leftarrow \frac{\mathbf{E}}{\|\mathbf{E}\|_F}$   
 2 **repeat**  
 3      $\mathbf{x} \leftarrow \text{prox}_{\lambda_1 \|\cdot\|_1}(\mathbf{z} - \mathbf{E}^T \mathbf{v})$  //  $\ell_1$  prox, (4.26)  
 4      $\mathbf{x} \leftarrow \text{prox}_{\lambda_{\infty} \|\cdot\|_{1,\infty}^{\mathcal{H}}}(\mathbf{x})$  //  $\ell_1/\ell_{\infty}$ , (Bach et al., 2011; Duchi et al., 2008)  
 5      $\mathbf{x} \leftarrow \text{prox}_{\lambda_2 \|\cdot\|_2}(\mathbf{x})$  //  $\ell_2$  prox, (4.21)  
 6      $\mathbf{v} \leftarrow \mathbf{v} + \beta \gamma (\mathbf{E}\mathbf{x} - \text{prox}_{\lambda_{\mathcal{G}}/\gamma \|\cdot\|_2}(\frac{\mathbf{v} + \gamma \mathbf{E}\mathbf{x}}{\gamma}))$  //  $\dagger$   
 7 **until convergence**  
**Algorithm 2:** Dual forward-backward splitting algorithm to compute the proximal mapping of  $\|\cdot\|_{\Phi}$ .

presented in eq. (4.21).

In  $f$ , the sum of weighted  $\ell_1$  and  $\ell_1/\ell_{\infty}$  norms is a term that appears in the *sparse group lasso* and can be computed by applying the *soft thresholding* (Parikh and Boyd, 2013)

$$\text{prox}_{s \|\cdot\|_1}(\mathbf{y}) = (\mathbf{y} - s)_+ - (-\mathbf{y} - s)_+ \quad (4.26)$$

first, followed by *group soft thresholding* (Bach et al., 2011). As shown by Haeffele et al. (2014, Thm. 3), the  $\ell_2$  term can be additionally incorporated by composition, i.e. by subsequently applying *block soft thresholding* as presented in (4.21).

Now, to solve the *group soft thresholding* for the proximal mapping of the  $\ell_1/\ell_{\infty}$  norm, one can use the fact that for any norm  $\omega$  with dual norm  $\omega^*$ ,  $\text{prox}_{s\omega}(\mathbf{y}) = \mathbf{y} - \text{proj}_{\omega^* \leq s}(\mathbf{y})$ . By  $\text{proj}_{\omega^* \leq s}(\mathbf{y})$ , we denote the projection of  $\mathbf{y}$  onto the  $\omega^*$  norm ball with radius  $s$  (Bach et al., 2011). The dual norm of the  $\ell_1/\ell_{\infty}$  norm, eq. (4.7), is the  $\ell_{\infty}/\ell_1$  norm

$$\|\mathbf{z}\|_{\infty,1}^{\mathcal{H}} = \max_{g \in \mathcal{H}} \|\mathbf{z}_g\|_1. \quad (4.27)$$

The orthogonal projection  $\text{proj}_{\|\cdot\|_{\infty,1}^{\mathcal{H}} \leq s}$  onto the  $\ell_{\infty}/\ell_1$  ball is obtained by projecting separately each subvector  $\mathbf{z}_g$  onto the  $\ell_1$  ball in  $\mathbb{R}^{|g|}$  (Bach et al., 2011). This demands an efficient projection onto the  $\ell_1$  norm ball. Due to the special structure of our groups in  $\mathcal{H}$ , i.e. there are exactly  $N$  non-overlapping groups, each of which consisting of three elements, a vectorised Matlab implementation of the method by Duchi et al. (2008) can be employed.

**Definition 2.** (*Convex Conjugate*)

Let  $\theta^*(\mathbf{y}) = \sup_{\mathbf{x}} (\mathbf{y}^T \mathbf{x} - \theta(\mathbf{x}))$  be the convex conjugate of  $\theta$ .

The update of the dual variable  $\mathbf{v}$  in algorithm 2 (see the line marked with  $\dagger$ ) is based on the update

$$\mathbf{v} \leftarrow \mathbf{v} + \beta (\text{prox}_{\gamma(\lambda_{\mathcal{G}} \|\cdot\|_2)^*}(\mathbf{v} + \gamma \mathbf{E}\mathbf{x}) - \mathbf{v}), \quad (4.28)$$

presented by Combettes and Pesquet (2011); Combettes et al. (2010), where  $(\cdot)^*$  denotes the convex conjugate.

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

---

Let us introduce some tools first.

**Lemma 2.** (*Extended Moreau Decomposition*) (Parikh and Boyd, 2013)

It holds that

$$\forall \mathbf{y} : \mathbf{y} = \text{prox}_{s\theta}(\mathbf{y}) + s \text{prox}_{\theta^*/s}(\mathbf{y}/s). \quad (4.29)$$

**Lemma 3.** (*Conjugate of conjugate*) (Boyd and Vandenberghe, 2009)

For closed convex  $\theta$ , it holds that  $\theta^{**} = \theta$ .

**Corollary 1.** For convex and closed  $\theta$ , it holds that

$$\forall \mathbf{y} : \mathbf{y} = \text{prox}_{s\theta^*}(\mathbf{y}) + s \text{prox}_{\theta/s}(\mathbf{y}/s). \quad (4.30)$$

*Proof.* Define  $\theta' = \theta^*$  and apply Lemma 2 and Lemma 3 with  $\theta'$  in place of  $\theta$ .  $\square$

Since for our choice of  $\theta = \lambda_{\mathcal{G}} \|\cdot\|_2$ ,  $\theta$  is a closed convex function, by Corollary 1, one can write

$$\text{prox}_{\gamma(\lambda_{\mathcal{G}}\|\cdot\|_2)^*}(\mathbf{y}) = \mathbf{y} - \gamma \text{prox}_{(\lambda_{\mathcal{G}}/\gamma)\|\cdot\|_2}(\mathbf{y}/\gamma). \quad (4.31)$$

With that, the right-hand side of eq. (4.28) can be written as

$$\begin{aligned} \mathbf{v} + \beta(\mathbf{v} + \gamma \mathbf{E}\mathbf{x} - \gamma \text{prox}_{\lambda_{\mathcal{G}}/\gamma\|\cdot\|_2}(\frac{\mathbf{v} + \gamma \mathbf{E}\mathbf{x}}{\gamma}) - \mathbf{v}) = \\ \mathbf{v} + \beta\gamma(\mathbf{E}\mathbf{x} - \text{prox}_{\lambda_{\mathcal{G}}/\gamma\|\cdot\|_2}(\frac{\mathbf{v} + \gamma \mathbf{E}\mathbf{x}}{\gamma})). \end{aligned} \quad (4.32)$$

### 4.5.3 Computational Complexity

In practice it holds that  $N \gg \max(M, K)$ . The gradient steps for the updates of  $\Phi$  and  $\mathbf{A}$  have both time complexity  $\mathcal{O}(N \max(M^2, K^2))$ , the proximal step for  $\mathbf{A}$  has complexity  $\mathcal{O}(MK)$ , and the proximal step for  $\Phi$  has complexity  $\mathcal{O}(MN|\mathcal{E}|n_{\text{it}})$ , where  $n_{\text{it}}$  is the number of iterations for the dual forward-backward splitting procedure (we used a maximum of  $n_{\text{it}} = 20$ ). Thus, the total time complexity for one iteration in algorithm 1 is  $\mathcal{O}(N \max(M^2, K^2) + MN|\mathcal{E}|n_{\text{it}})$ . Since in practice the number of vertices  $N$  and the number of edges in the graph  $|\mathcal{E}|$  are larger than  $M$  and  $K$ , the runtime complexity is dominated by the proximal step for  $\Phi$ , which in our experiments takes around 60% of the time for the brain shapes dataset ( $N = 1792$ ,  $M = 96$ ,  $K = 17$ ,  $|\mathcal{E}| = 19182$ ), and uses more than 90% of the time for the human body shapes dataset ( $N = 12500$ ,  $M = 48$ ,  $K = 1531$ ,  $|\mathcal{E}| = 99894$ ).

### 4.5.4 Factor Splitting

The *factor splitting* procedure is presented in algorithm 3.

```

Input: factor  $\phi \in \mathbb{R}^{N \times 3}$  where  $\text{vec}(\phi) = \Phi_m$ , graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 
Output:  $J$  factors  $\Phi' \in \mathbb{R}^{3N \times J}$  with local support
Initialise:  $\mathcal{E}' = \emptyset$ ,  $\Phi' = []$ 
// build activation graph  $\mathcal{G}'$ 
1 foreach  $(i, j) = e \in \mathcal{E}$  do
2   | if  $\phi_{i,:} \neq \mathbf{0} \vee \phi_{j,:} \neq \mathbf{0}$  then // vertex  $i$  or  $j$  is active
3   |   |  $\mathcal{E}' = \mathcal{E}' \cup \{e\}$  // add edge
4  $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ 
// find connected components (Tarjan, 1972)
5  $\mathcal{C} = \text{connectedComponents}(\mathcal{G}')$  //  $\mathcal{C} \subset 2^{\mathcal{V}}$ 
6 foreach  $c \in \mathcal{C}$  do // add new factor for  $c \subset \mathcal{V}$ 
7   |  $\phi' = \mathbf{0}_{N \times 3}$ 
8   |  $\phi'_{c,:} = \phi_{c,:}$ 
9   |  $\Phi' = [\Phi', \text{vec}(\phi')]$ 
    
```

**Algorithm 3:** Factor splitting procedure.

### 4.5.5 Convergence Plots

For 100 different initialisations (cf. Section 4.3), the convergence plots are shown for both datasets in Fig. 4.8. It can be seen that the main convergence occurs after around 10 iterations. Moreover, for all 100 initialisations the objective value are near-congruent.

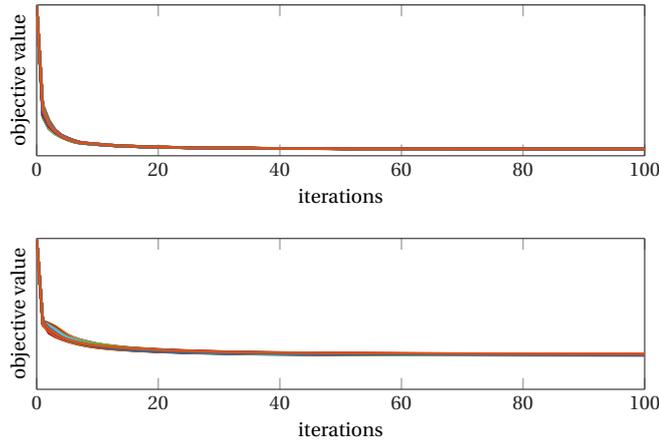


Figure 4.8: Convergence plots for the brain shapes dataset (top) and the human body shapes dataset (bottom). The iterations are shown on the horizontal axis and the (relative) objective value in eq. (4.4) is shown on the vertical axis. Note that in each subfigure all 100 lines are near-congruent.

### 4.5.6 Parameter Random Sampling

The methods kPCA, SPCA, SSPCA, SPLOCS and our method require various parameters to be set. In order to find a good parametrisation we conducted random sampling over the parameter space, where we determined reasonable ranges for the parameters experimentally.

In Table 4.1 the distributions and default values of the parameters for each method are

## Chapter 4. Linear Shape Deformation Models with Local Support using Graph-based Structured Matrix Factorisation

---

given.  $\mathcal{U}(a, b)$  is the uniform distribution with the open interval  $(a, b)$  as support,  $10^{\mathcal{U}(\cdot, \cdot)}$  is a distribution of the random variable  $y = 10^x$ , where  $x \sim \mathcal{U}(\cdot, \cdot)$ , and  $\bar{d}_{\max}$  is the largest distance between all pairs of vertices of the mean shape  $\bar{X}$ .

Table 4.1: Assumed distributions of the parameters interpreted as random variables.

Method	Parameter	Distribution / Default Value
kPCA	$\beta$	$\beta \sim (\mathcal{U}(1, 10))^{-1}$
SPCA	$\lambda$	$\lambda \sim \frac{1}{3N} 10^{\mathcal{U}(-4, -3)}$ (see eq. (2), Jenatton et al. (2010))
SSPCA	$\lambda$	$\lambda \sim \frac{1}{N} 10^{\mathcal{U}(-4, -3)}$ (see eq. (2), Jenatton et al. (2010))
SPLOCS	$\lambda$	$\lambda \sim (3K) \cdot 10^{\mathcal{U}(-4, -3)}$ ; $d_{\min} \sim c - w$ ; $d_{\max} \sim c + w$ (see eq. (6) by Neumann et al. (2013)), where $c \sim \mathcal{U}(0.1, \bar{d}_{\max} - 0.1)$ and $w \sim \mathcal{U}(0, \min( c - 0.1 ,  \bar{d}_{\max} - 0.1 - c ))$
our	$\beta$	$\beta \sim (\mathcal{U}(1, 10))^{-1}$ ; $\lambda = 64 \cdot \frac{3NK}{M}$ ; $\lambda_{\mathcal{G}} = \frac{1}{\sqrt{3 \mathcal{E} }}$ ; $\lambda_1 = \frac{1}{\sqrt{3N}}$ ; $\lambda_2 = \frac{1}{\sqrt{3N}}$ ; $\lambda_{\infty} = 2 \cdot \frac{1}{\sqrt{N}}$ ; $\lambda_A = 10^{-4} \cdot \frac{1}{\sqrt{K}}$

For each of the  $n_r$  random samples (we set  $n_r = 500$  for the brain shapes dataset, and  $n_r = 50$  for the human body shapes, due to the large size of the dataset) we compute the 8 scores described in Section 4.3.1 and store them in the score matrix  $\mathbf{S} \in \mathbb{R}^{n_r \times 8}$ . After (linearly) mapping the elements of each  $n_r$ -dimensional column vector in  $\mathbf{S}$  onto the interval  $[0, 1]$ , the best parametrisation is determined by finding the index of the smallest value of the vector  $\mathbf{S}\mathbf{1}_8 \in \mathbb{R}^{n_r}$ .

For the lambda parameters of the proposed method we identified default values that we used for the evaluation of both datasets. Moreover, a normalisation of the parameters is conducted for all methods. For kPCA, SPCA, SSPCA and our method the random sampling was conducted only for a single parameter, whereas for the SPLOCS method three parameters had to be set, two of them related to the size of the local support region.

# 5 Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds

## Abstract

The reconstruction of an object's shape or surface from a set of 3D points is a fundamental topic in material and life sciences, computationally handled in computer graphics. Such points usually stem from optical or tactile 3D coordinate measuring equipment. Surface reconstruction also appears in medical image analysis, e.g. in anatomy reconstruction from tomographic measurements or the alignment of intra-operative navigation and preoperative planning data. In such scenarios, one usually has to deal with *sparse* data, which significantly aggravates the problem. However, medical applications often provide contextual information on the 3D point data that can be used to adopt prior information on the shape that is to be reconstructed from the measurements. We propose the use of a statistical shape model (SSM) as a prior for surface reconstruction. The SSM is represented by a point distribution model (PDM), which is associated with a surface mesh. Using the shape distribution that is modelled by the PDM, we reformulate the problem of surface reconstruction from a probabilistic perspective based on a Gaussian Mixture Model (GMM). In order to do so, the given measurements are interpreted as samples of the GMM. By using mixture components with anisotropic covariances, that are oriented according to the surface normals at the PDM points, a surface-based fitting is accomplished. Estimating the parameters of the GMM in a maximum a posteriori manner yields the reconstruction of the surface from the given measurements. We compare our method to the current standard approach (Iterative Closest Points) on several different anatomical datasets/SSMs (brain, femur, tibia, hip, liver) and demonstrate superior accuracy and robustness on sparse data.

## 5.1 Introduction

The reconstruction of an object’s shape or surface from a set of 3D points is a common topic in materials and life sciences, typically addressed in computer graphics (Berger et al., 2014). The points are usually produced by optical or tactile 3D coordinate measuring equipment, where non-contact measurements are often disturbed by artefacts like reflections, noise, or outliers, making a proper geometric reconstruction even more difficult.

Surface reconstruction also appears in medical image analysis, e.g. in anatomy reconstruction from tomographic measurements via image segmentation. Conceptually speaking, images provide implicit information on the location of anatomical structures via intensity levels. This information is – in many approaches – converted into geometric information via some kind of feature extraction method, either automatically (atlas- or model-based approaches, learning methods, landmark detection, etc.) or interactively (e.g. via semi-automated or manual labelling of image regions). In both cases, the uncertainty due to noise or artefacts in the input image data is carried over to the geometric data during the transformation process. In those approaches the segmentation problem is effectively re-phrased as a surface reconstruction problem. Another example can be found in computer-assisted surgery where a pre-operative therapy plan is transferred to the operating room (OR) by means of a navigation system (or imaging – however, this can be seen again as a segmentation problem). Here, optical or electromagnetic tracking devices generate 3D positions of a patient’s surface intra-operatively. Matching a therapy plan to those measurements amounts to a surface reconstruction and registration problem.

In contrast to mere 3D point-clouds that may represent virtually any object, medical image data yield additional contextual information that can be used to adopt prior information on the anatomical structures to be reconstructed. Heckel et al. (2011) make use of the variational interpolation method (Turk and O’Brien, 1999), which essentially uses a general prior on the surface smoothness. Going one step further, Pauly et al. (2005) and Gal et al. (2007) have considered templates for 3D scan completion that are matched to the measurements. However, these methods are limited since the available measurements are assumed to be sufficiently dense (Berger et al., 2014). To tackle this limitation, Bernard et al. (2015a) suggested the use of a Statistical Shape Model (SSM) for surface reconstruction. Because the class of anatomical structures is known for clinical routine tasks such as segmentation, registration, or intra-operative navigation, it is possible to use their shapes as geometric priors.

With SSMs being rather strong and specific priors, a central question is how robustly and accurately they can be used for surface reconstruction from a sparse set of surface point measurements. Here we present a novel shape-aware surface reconstruction approach,

which we evaluate accordingly. Using a Point Distribution Model (PDM) (Cootes and Taylor, 1992), we incorporate a prior into our reconstruction framework that captures the likely shape of an object to be extracted. In doing so, we reformulate the problem of surface reconstruction from a probabilistic perspective, embedding the prior distribution of the SSM parameters that explain plausible shapes into the objective function.

Our evaluation will consider several different anatomical structures and SSMs (brain, femur, tibia, hip, liver), and sparse input point scenarios which may occur in different applications, ranging from interactive segmentation to intra-operative registration for navigation. Rather than restricting ourselves to particular applications by investigating application-specific aspects in an exhaustive manner, our goal is to demonstrate the *general applicability* of the proposed approach in order to emphasize that the method may be useful in a wide range of settings.

The remainder of this article is organised as follows: Section 5.2 summarises previous research relevant to our methodology, followed by a concise summary of our key contributions in Section 5.3. In Sections 5.4 and 5.5, we familiarise the reader with the theoretical background and define the scope of our work. In Section 5.6, we present several variants of the shape-aware surface reconstruction method, including a time complexity and convergence analysis. Section 5.7 contains experiments conducted using the proposed methods. We draw our conclusions in Section 5.8.

## 5.2 Related Work

A plethora of methods for general surface reconstruction have been presented in the literature so far (see e.g. the works by Raya and Udupa (1990); Bolle and Vemuri (1991); Herman et al. (1992); Hoppe et al. (1992); Edelsbrunner and Mücke (1994); Bajaj et al. (1995); Amenta et al. (1998); Bernardini et al. (1999); Treece et al. (2000); Kazhdan et al. (2006); Schroers et al. (2014)). Many of them are summarized and described in the State-of-the-Art Report (STAR) by Berger et al. (2014). In the remainder of the present section, only those surface reconstruction methods will be discussed that go beyond pure smoothness assumptions and make use of more explicit shape priors.

For the completion of 2D shapes, Guo et al. (2012, 2013) incorporate templates from a shape database as (geometric) priors into a Bayesian framework. Similarly, a database of 3D shapes is used by Pauly et al. (2005) for completing 3D surface scans. For increased flexibility compared to static priors, Gal et al. (2007) use a context-specific database of *local* shape priors, where the input data is matched by (dynamically) combining multiple local shape priors into a global prior. As pointed out by Berger et al. (2014), both approaches described by Pauly et al. (2005) and Gal et al. (2007) are limited by the assumption that the point-clouds are assumed to be sufficiently dense. A unified

framework for repairing the geometry and texture of meshes has been presented by Park et al. (2006). They employ a context-based geometry filling procedure for filling holes in the surface, where available local patches of the mesh are used to fill its missing parts. The task of obtaining high-resolution 3D meshes from low-quality input is tackled by Shen et al. (2012). Their approach is to dynamically assemble object templates from a database of object parts. As the name suggests, the 3D shape completion methods discussed above have a common focus on completing (mostly small) missing parts of meshes obtained from 3D scans. However, our interest lies primarily in methods that go beyond patching or improving low-resolution input.

Blanz et al. (2004) have presented a closed-form solution for SSM-based 3D surface reconstruction from a sparse set of points, which relies, however, on the assumption that such a set of points is already in correspondence to the model. Albrecht et al. (2013) introduced posterior shape models that have the objective to model the distribution of a whole shape given only partial information. This method assumes that the corresponding model points of the available partial data are known. In their experiments this issue is either solved manually or using the Iterative Closest Point (ICP) method (Besl and McKay, 1992). Similarly, for shape prediction from sparse observations, Blanc and Székely (2012) use a variant of ICP that evaluates multiple initialisations. Anguelov et al. (2005) introduced the shape completion and animation of people (SCAPE) method, where one model for pose deformations and one model for shape variations are learned separately. The main objective of this method is the completion of body shapes based on a small number of known positions for some of the model points. Applied to bone models, Rajamani et al. (2007) fit an SSM to a small number of anatomical landmarks that correspond to some of the model points. Baka et al. (2010) fit an SSM to sparse data points that are in correspondence with the model, applied to 2D heart datasets. By producing confidence intervals as output, their method is able to incorporate uncertainties in the input data. Instead of using a trained SSM, Lu et al. (2011) formulate a low-rank matrix recovery problem for restoring missing parts of objects in archaeological studies. Considering that a set of (incomplete) objects of the same class is available, and that correspondences between common parts are known, their approach is based on the assumption that all shapes are approximately linearly correlated. *A shortcoming of the methods discussed so far is that they all assume known correspondences.* However, if the objects do not exhibit a sufficient amount of distinct features, the identification of exact correspondences is very difficult or even infeasible in practice.

In the literature, there are several methods published addressing this difficulty in (automatically) detecting the correspondence between sparse data points and a model. Due to its simplicity, the ICP algorithm, where correspondences and transformations are estimated in an alternating manner, is a very popular method for the registration of two shapes represented as point sets. Numerous variants of the originally-proposed

method have since been developed, e.g. by Rusinkiewicz and Levoy (2001); Segal et al. (2009); Maier-Hein et al. (2012); Bouaziz et al. (2013); Billings et al. (2015). The surface reconstruction method by Stoll et al. (2006), for instance, deforms a given template to fit point-cloud data. To do so, the user defines initial correspondences between the template and a set of points, which are then refined iteratively in an ICP-like manner. Along the lines of Stoll et al. (2006), for knee surgery, Fleute and Lavallée (1998); Fleute et al. (1999) have presented a methodology to find pose and shape deformation parameters in order to fit an SSM to very sparse data points. Their approach resembles ICP due to the alternating closest point estimation and pose/deformation model parameter updates. Similarly, Chan et al. (2003) reconstruct 3D models of bones in orthopaedic surgery based on a sparse set of points obtained from ultrasound. Here shapes are repeatedly instantiated using a PDM, where each shape instance is used as input to ICP in order to (rigidly) fit the sparse points. This work has been extended by Barratt et al. (2008), who use a PDM defined on a regular grid instead of the shape surface. Zheng et al. (2006) have proposed a three-stage procedure for sparse shape reconstruction in computer-assisted orthopaedic surgery. In the first stage, the pose of the sparse points is adapted using ICP so that they best fit the mean shape. In the second stage, the shape deformation parameters are estimated for the given correspondences, which are eventually refined using a further deformation based on thin-plate splines.

Alternatively, in the work of Chang and Zwicker (2009), the registration of articulated shapes in two range scans is based on a reduced deformation model defined on a regular grid. In this approach, the deformations are modelled by a convex combination of rigid transformations, where the weights are spatially varying. The registration is performed by alternately updating the rigid transformations and their weights, where closest point correspondences are recomputed after each step.

A promising approach of fitting a hand pose model to points that are sufficiently densely sampled from depth images is presented by Taylor et al. (2014, 2016). The authors formulate a continuous optimisation problem that solves simultaneously for surface-based correspondences and model parameters using a nonlinear sum of squares objective.

Despite the immense amount of literature on applications and variations of Statistical Shape Models (for an overview the interested reader is referred to the works by Heimann and Meinzer (2009); Cootes and Taylor (1992); van Ginneken et al. (2002)), there have been a limited number of investigations into the use of SSMs for interactive segmentation. In the work of van Ginneken et al. (2003), a user directly manipulates points of the PDM, which has the disadvantage that the user needs to estimate the (unknown) corresponding position of the considered model point in the image. This is particularly difficult if the object does not exhibit distinct features. In their slice-wise SSM-based segmentation of abdominal aortic aneurysms, de Bruijne et al. (2004) initialise the current slice's PDM

fitting from the segmentation result of the previous slice, with the option of manually correcting segmentations on a per-slice basis. Liu and Udupa (2009) present Oriented Active Shape Models where the semi-automatic live-wire technique is coupled with an Active Shape Model. Other authors present tools for the posterior correction of model-based segmentations, such as Timinger et al. (2003); Schwarz et al. (2008); Tan and Acharya (2014). van den Hengel et al. (2007) present an interactive procedure based on a set of rules and 2D sketches for completing partial 3D models in structure from motion.

To summarise, existing approaches performing surface reconstruction using SSMs either assume known correspondences or estimate discrete correspondences in an ICP-like manner, where the surface reconstruction is achieved by fitting the SSM in a point-wise manner to the given data points. One exception is the work by Taylor et al. (2016), where, for sufficiently dense data, a surface-based hand pose model fitting is performed.

### 5.3 Main Contributions

Our main contribution is the introduction of a *surface-based* SSM fitting procedure in order to reconstruct a surface from a sparse 3D point-cloud. We evaluate our approach on a broad range of datasets (brain, femur, tibia, hip, liver) and show superior accuracy and robustness compared to the standard approach of using ICP. The surface-based fitting procedure is achieved by the following methodological contributions:

- By extending powerful existing point-set registration procedures (Myronenko et al., 2007; Myronenko and Song, 2010; Horaud et al., 2011) such that anisotropic covariances are used in combination with a PDM as transformation model, we obtain a shape-aware surface reconstruction method that we show to be superior to ICP with respect to robustness and accuracy.
- In earlier work, we used a spherical (or isotropic) GMM that accounted for a *point-based* matching (Bernard et al., 2015a). We now complement this work by presenting a formulation that is based on anisotropic GMMs that are oriented by the surface normals, which in turn accounts for a *surface-based* fitting.
- A *rigorous and self-contained derivation* of the surface-based fitting method is presented, leading to an Expected Conditional Maximisation (ECM) algorithm (Meng and Rubin, 1993). ECM shares the same convergence properties as the Expectation Maximisation (EM) method while being more general.
- We develop a fast approximation of the ECM-based fitting method. The approximation method has the same computational complexity as the spherical GMM-based method and is empirically less prone to unwanted local optima compared to the

ECM-based method.

## 5.4 Background

In this section we introduce our notation and we provide a definition of PDMs as they are used throughout the subsequent sections.

### 5.4.1 Notation

$\mathbf{1}_p$  and  $\mathbf{0}_p$  denote the  $p$ -dimensional column vector containing ones and zeros, respectively.  $\mathbf{I}_p$  denotes the  $p \times p$  identity matrix and  $\text{diag}(\mathbf{x})$  is the  $p \times p$  matrix with the elements of the vector  $\mathbf{x} \in \mathbb{R}^p$  on its diagonal. For a matrix  $\mathbf{A}$ , by  $\mathbf{A}_{i,j}$  we denote the (scalar) element in row  $i$  and column  $j$ . The colon-notation is used to denote all rows or columns, e.g.  $\mathbf{A}_{:,j}$  denotes the column vector formed by the  $j$ -th column of  $\mathbf{A}$ .

By  $\mathbf{p}(x)$  we denote the *probability density function* (p.d.f.), or *probability mass function* in the discrete case, where  $x$  can indicate both a random variable and a realisation of it, depending on the context.  $\mathcal{N}(x|\mu, \Sigma)$  denotes the Gaussian p.d.f. with mean  $\mu$  and covariance matrix  $\Sigma$ .

### 5.4.2 Point Distribution Models

Statistical Shape Models based on Point Distribution Models (Cootes and Taylor, 1992) are an established technique to capture nonlinear shape deformations in  $\mathbb{R}^3$  from training data by using a linear model in the higher-dimensional shape space. Let  $\{\mathbf{X}_k : 1 \leq k \leq K\}$  be the set of  $K$  training shapes, where each shape is represented by  $N$  points (or vertices) in 3 dimensions written as the matrix  $\mathbf{X}_k \in \mathbb{R}^{N \times 3}$ . In order to process multiple shapes in a meaningful way, it is essential that the rows for all  $\mathbf{X}_k$  are corresponding, i.e. the  $N$  vertices of all  $K$  shapes are in correspondence. The PDM is obtained by finding an (affine) subspace close to the subspace spanned by the training shapes, which is commonly performed by Principal Components Analysis (PCA).

We first define  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K] \in \mathbb{R}^{3N \times K}$ , with  $\mathbf{x}_k = \text{vec}(\mathbf{X}_k) \in \mathbb{R}^{3N}$ . This allows to compute the modes of shape variability as the eigenvectors of the sample covariance matrix  $\mathbf{C} = \frac{1}{K-1}(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_K^T)(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_K^T)^T$ , where  $\bar{\mathbf{x}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k$  denotes the mean of all corresponding vertices for all shapes in  $3N$ -dimensional shape space. Let  $\Phi \in \mathbb{R}^{3N \times M}$  be the matrix of the first  $M$  eigenvectors of  $\mathbf{C}$  with largest eigenvalues. The PDM  $\mathbf{y}(\alpha) : \mathbb{R}^M \rightarrow \mathbb{R}^{3N}$  is a vector-valued function defined as

$$\mathbf{y}(\alpha) = \bar{\mathbf{x}} + \Phi\alpha, \quad (5.1)$$

where  $\boldsymbol{\alpha} \in \mathbb{R}^M$  is the shape deformation parameter. The deformation of vertex  $i$  through  $\boldsymbol{\alpha}$  is denoted by

$$y_i(\boldsymbol{\alpha}) = \bar{x}_i + \Phi_i \boldsymbol{\alpha}, \quad (5.2)$$

where the three rows of vertex  $i$  are selected appropriately from  $\mathbf{y}(\boldsymbol{\alpha})$ ,  $\bar{\mathbf{x}}$  and  $\Phi$  to obtain  $y_i(\boldsymbol{\alpha}) \in \mathbb{R}^3$ ,  $\bar{x}_i \in \mathbb{R}^3$  and  $\Phi_i \in \mathbb{R}^{3 \times M}$ .

A common assumption is that  $\boldsymbol{\alpha}$  follows a zero-mean Gaussian distribution, i.e.  $\mathbf{p}(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha} | \mathbf{0}_M, \Lambda)$ , where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_M)$  with  $\lambda_m$  being the  $m$ -th eigenvalue of  $\mathbf{C}$ . Therefore, thanks to imposing a distribution upon  $\boldsymbol{\alpha}$ , we obtain a distribution over shapes (Albrecht et al., 2013).

Usually, in addition to the PDM accounting for shape deformations, a rigid transformation is employed in order to account for the absolute pose of the shape  $\mathbf{y}(\boldsymbol{\alpha})$  with respect to some reference (e.g. the image coordinate system). In the following we assume that the predominant part of the pose has already been normalised. Furthermore, minor pose variations can be (approximately) captured implicitly in the PDM.

## 5.5 Problem Formulation

Given is a PDM that serves as prior for plausible shapes. Also, we assume that (for fixed  $\boldsymbol{\alpha}$ ) the PDM points are vertices of an oriented triangular surface mesh  $\mathcal{M}$ . Additionally, we are given the set  $\mathcal{P} = \{p_j : 1 \leq j \leq P\}$  of  $P$  surface points of the shape that is to be reconstructed, where  $\mathcal{P}$  is sparse in the sense that it only contains few measurements  $p_j$  of the object's surface. The objective is to find the deformation parameter  $\boldsymbol{\alpha}$  such that  $\mathbf{y}(\boldsymbol{\alpha})$  "best agrees" with the points in  $\mathcal{P}$ .

### 5.5.1 A Generative Model

Motivated by the widely used Coherent Point Drift (CPD) approach for general point-set registration (Myronenko et al., 2007; Myronenko and Song, 2010), we now introduce a probabilistic model based on imposing a distribution over each vertex  $i$  by using a GMM.

Given the set  $\mathcal{P}$  and a PDM, we consider the following assumptions:

1. For  $i = 1, \dots, N$ , each PDM vertex position  $y_i(\boldsymbol{\alpha}) \in \mathbb{R}^3$  is considered as the mean of a 3D Gaussian distribution with covariance  $\Sigma_i$ .
2. Each point  $p_j \in \mathcal{P}$  can be uniquely mapped to one specific vertex index  $i$ , its *generating component*, from whose distribution it is drawn.

As such, each point  $p_j$  follows the distribution

$$\mathbf{p}(p_j|i, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_i) = \mathcal{N}(p_j|y_i(\boldsymbol{\alpha}), \boldsymbol{\Sigma}_i), \quad (5.3)$$

where all  $p_j$  for  $j = 1, \dots, P$  are independent. The corresponding graphical model is depicted in Fig. 5.1. Note that all of the  $N$  Gaussian components are parameterised by  $\boldsymbol{\alpha}$  and the covariances  $\hat{\boldsymbol{\Sigma}} := \{\boldsymbol{\Sigma}_i\}$ . We assume that each component with index  $i$  is chosen with equal probability, i.e.  $\mathbf{p}(i) = \frac{1}{N}$ .

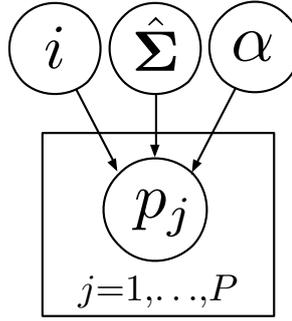


Figure 5.1: Graphical model of the generating process of  $\mathcal{P}$ .

Our objective is to find the parameters  $\boldsymbol{\alpha}$  and  $\hat{\boldsymbol{\Sigma}}$  that are most likely to have generated the data points  $\mathcal{P}$ . Since the generating component  $i_j$  of  $p_j$  is unknown, the indices of the generating components are treated as *latent* variables. Therefore, to incorporate the uncertainty about the generating component, we consider a GMM for the distribution of  $p_j$ , leading to

$$\mathbf{p}(p_j|\boldsymbol{\alpha}, \hat{\boldsymbol{\Sigma}}) = \sum_{i=1}^N \mathbf{p}(i) \mathbf{p}(p_j|i, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_i) \quad (5.4)$$

$$= \frac{1}{N} \sum_{i=1}^N \mathcal{N}(p_j|y_i(\boldsymbol{\alpha}), \boldsymbol{\Sigma}_i). \quad (5.5)$$

Using Bayes' theorem, one can derive the probability that the  $i$ -th mixture component has generated the point  $p_j$ , given also  $\boldsymbol{\alpha}$  and  $\hat{\boldsymbol{\Sigma}}$ , as

$$\mathbf{p}(i|p_j, \boldsymbol{\alpha}, \hat{\boldsymbol{\Sigma}}) = \frac{\mathbf{p}(p_j|i, \boldsymbol{\alpha}, \hat{\boldsymbol{\Sigma}})\mathbf{p}(i)\mathbf{p}(\boldsymbol{\alpha})\mathbf{p}(\hat{\boldsymbol{\Sigma}})}{\sum_{i'} \mathbf{p}(p_j|i', \boldsymbol{\alpha}, \hat{\boldsymbol{\Sigma}})\mathbf{p}(i')\mathbf{p}(\boldsymbol{\alpha})\mathbf{p}(\hat{\boldsymbol{\Sigma}})} \quad (5.6)$$

$$= \frac{\mathbf{p}(p_j|i, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_i)\mathbf{p}(i)}{\sum_{i'} \mathbf{p}(p_j|i', \boldsymbol{\alpha}, \boldsymbol{\Sigma}_{i'})\mathbf{p}(i')} \quad (5.7)$$

$$= \frac{\mathcal{N}(p_j|y_i(\boldsymbol{\alpha}), \boldsymbol{\Sigma}_i)}{\sum_{i'} \mathcal{N}(p_j|y_{i'}(\boldsymbol{\alpha}), \boldsymbol{\Sigma}_{i'})}. \quad (5.8)$$

### 5.5.2 Optimisation using EM

If the generating component  $i_j$  of  $p_j$  is unknown, according to eq. (5.5) all points  $p_j \in \mathcal{P}$  are independent and identically distributed (i.i.d.). Thus, the log-likelihood as a function of the model parameters  $\boldsymbol{\theta} := (\boldsymbol{\alpha}, \hat{\boldsymbol{\Sigma}})$  reads

$$\begin{aligned} L(\boldsymbol{\theta}) &= \ln \mathfrak{p}(\mathcal{P}|\boldsymbol{\theta}) = \ln \prod_{j=1}^P \sum_{i=1}^N \mathfrak{p}(i)\mathfrak{p}(p_j|i, \boldsymbol{\theta}) \\ &= \sum_{j=1}^P \ln \sum_{i=1}^N \mathfrak{p}(i)\mathfrak{p}(p_j|i, \boldsymbol{\theta}). \end{aligned} \quad (5.9)$$

The maximisation of  $L$  w.r.t.  $\boldsymbol{\theta}$  cannot be solved readily due to the sum appearing inside the logarithm. Therefore, a common approach is to employ an iterative method for the maximisation, where by  $\boldsymbol{\theta}^{(n)}$  we denote the estimate of  $\boldsymbol{\theta}$  at iteration  $n$ . Rewriting eq. (5.9) as

$$L(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \sum_{j=1}^P \ln \sum_{i=1}^N \mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)}) \frac{\mathfrak{p}(i)\mathfrak{p}(p_j|i, \boldsymbol{\theta})}{\mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)})} \quad (5.10)$$

allows applying Jensen's inequality (Boyd and Vandenberghe, 2009), leading to

$$L(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) \geq \sum_{j=1}^P \sum_{i=1}^N \mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)}) \ln \frac{\mathfrak{p}(i)\mathfrak{p}(p_j|i, \boldsymbol{\theta})}{\mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)})} =: Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}). \quad (5.11)$$

As such, the right-hand side of eq. (5.11), which we denote by  $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$ , is a lower bound for  $L(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$ . Maximising this lower bound is the key idea of the well-established EM algorithm (Dempster et al., 1977). In the E-step, the probabilities  $\mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)})$  are evaluated for fixed  $\boldsymbol{\theta}^{(n)}$  by using eq. (5.8). Then, in the M-step,  $Q$  in eq. (5.11) is maximised w.r.t.  $\boldsymbol{\theta}$  for the fixed  $\mathfrak{p}(i|p_j, \boldsymbol{\theta}^{(n)})$  computed before. An appealing characteristic of EM is that in each iteration the log-likelihood  $L$  is guaranteed to not decrease, which guarantees convergence.

## 5.6 Methods

We start by introducing an isotropic GMM, in analogy to the CPD algorithm (Myronenko et al., 2007; Myronenko and Song, 2010). The difference to CPD is that we use a PDM as transformation model. Subsequently, an anisotropic GMM approach is described, where the covariance matrices are oriented according to the surface normals at the PDM points. With that, we move from a purely point-based matching to a more surface-oriented fitting approach. Eventually, we describe a fast approximation of the anisotropic GMM method. By using an extension to this approximation, one can ensure that it is an instance of the Generalised Expectation Maximisation (GEM) method and thus the convergence is guaranteed.

### 5.6.1 Surface Reconstruction using an Isotropic GMM

Assuming that all  $N$  Gaussian components share the same (spherical) covariance  $\Sigma_i = \sigma^2 \mathbf{I}_3$ , the generative model of eq. (5.3) reads

$$\mathfrak{p}(p_j|i, \boldsymbol{\alpha}, \sigma) = \mathcal{N}(p_j|y_i(\boldsymbol{\alpha}), \sigma^2 \mathbf{I}). \quad (5.12)$$

With that, the update in the E-step in eq. (5.8) becomes

$$\mathfrak{p}(i|p_j, \boldsymbol{\alpha}, \sigma) = \frac{\exp(-\frac{1}{2\sigma^2} \|p_j - y_i(\boldsymbol{\alpha})\|^2)}{\sum_{i'} \exp(-\frac{1}{2\sigma^2} \|p_j - y_{i'}(\boldsymbol{\alpha})\|^2)}. \quad (5.13)$$

The M-step comprises maximising

$$Q(\boldsymbol{\alpha}, \sigma, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) = \text{const} + \sum_{j=1}^P \sum_{i=1}^N \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \ln[\mathfrak{p}(i)\mathfrak{p}(p_j|i, \boldsymbol{\alpha}, \sigma)] \quad (5.14)$$

w.r.t.  $\boldsymbol{\alpha}$  and  $\sigma$ . Setting the gradient of  $Q$  w.r.t.  $\boldsymbol{\alpha}$  to zero reveals that  $\boldsymbol{\alpha}$  can be found by solving the linear system  $\mathbf{A}\boldsymbol{\alpha} = \mathbf{b}$ , with

$$\mathbf{A} = \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \quad \text{and} \quad (5.15)$$

$$\mathbf{b} = \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \boldsymbol{\Phi}_i^T (p_j - \bar{x}_i). \quad (5.16)$$

Similarly,  $\sigma$  is computed by

$$\sigma^2 = \frac{1}{3P} \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \|p_j - y_i(\boldsymbol{\alpha})\|^2. \quad (5.17)$$

Since the maximisation of  $Q$  w.r.t.  $\boldsymbol{\alpha}$  does not depend on  $\sigma$ , we first compute  $\boldsymbol{\alpha}$  and then compute  $\sigma$  with the obtained  $\boldsymbol{\alpha}$ , which results in the global maximiser of  $Q$  in each M-step. As such, this case is an instance of the ordinary EM algorithm (Dempster et al., 1977). The pseudocode is presented in Algorithm 4.

```

Input:  $\bar{\mathbf{x}}, \boldsymbol{\Phi}, \mathcal{P}$ 
Output:  $\boldsymbol{\alpha}, \sigma^2$ 
Initialise:  $\boldsymbol{\alpha} = \mathbf{0}$ ,  $\sigma^2 = \frac{1}{3NP} \sum_{i,j} \|p_j - \bar{x}_i\|^2$ ,  $\mathbf{P} \in \mathbb{R}^{P \times N}$ ,  $\mathbf{y} = \bar{\mathbf{x}} + \boldsymbol{\Phi}\boldsymbol{\alpha}$ 
1 repeat
   // E-step
2   foreach  $j = 1, \dots, P$  do
3      $z = 0$ 
4     foreach  $i = 1, \dots, N$  do
5        $\mathbf{P}_{ji} = \exp(-\frac{1}{2\sigma^2} \|p_j - y_i\|^2)$ 
6        $z = z + \mathbf{P}_{ji}$ 
7      $\mathbf{P}_{j,:} = \frac{1}{z} \mathbf{P}_{j,:}$ 
   // M-step
8    $\boldsymbol{\alpha} = (\sum_i \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \sum_j \mathbf{P}_{ji})^{-1} (\sum_i \boldsymbol{\Phi}_i^T \sum_j \mathbf{P}_{ji} (p_j - \bar{x}_i))$ 
9    $\mathbf{y} = \bar{\mathbf{x}} + \boldsymbol{\Phi}\boldsymbol{\alpha}$ 
10   $\sigma^2 = \frac{1}{3P} \sum_{i,j} \mathbf{P}_{ji} \|p_j - y_i\|^2$ 
11 until convergence

```

**Algorithm 4:** Pseudocode of the isotropic GMM fitting method.

### Maximum A Posteriori (MAP) Solution

We can cast the log-likelihood from eq. (5.9) into a Bayesian view, leading to

$$L^{\text{posterior}}(\boldsymbol{\theta}) = \ln[\mathbf{p}(\mathcal{P}|\boldsymbol{\theta})\mathbf{p}(\boldsymbol{\theta})], \quad (5.18)$$

where additional knowledge in the form of the prior distribution  $\mathbf{p}(\boldsymbol{\theta})$  is incorporated. By assuming  $\mathbf{p}(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\alpha}|\mathbf{0}_M, \Lambda)$  as described in Section 5.4.2, and choosing a uniform prior for  $\mathbf{p}(\sigma)$ , the prior distribution results in  $\mathbf{p}(\boldsymbol{\theta}) \propto \mathbf{p}(\boldsymbol{\alpha})$ . When estimating the parameters in the M-step,  $\boldsymbol{\alpha}$  is now found by solving the system  $\hat{\mathbf{A}}\boldsymbol{\alpha} = \mathbf{b}$  where  $\hat{\mathbf{A}} = \mathbf{A} + \sigma^2\Lambda^{-1}$ . The additional term  $\sigma^2\Lambda^{-1}$  corresponds to a Tikhonov regulariser. However, now both updates of  $\boldsymbol{\alpha}$  and  $\sigma$  depend on each other, leading to an instance of the ECM algorithm, which has similar convergence properties as the ordinary EM. To obtain the MAP solution, we have to change the expression in line 8 in Algorithm 4, which uses now the value of  $\sigma$  from the previous iteration to update  $\boldsymbol{\alpha}$ . We refer to the MAP solution of the surface reconstruction using an isotropic GMM as ISO.

#### 5.6.2 Surface Reconstruction using an Anisotropic GMM

The previous approaches use the same (spherical) covariance for each of the  $N$  Gaussian components, which can be seen as a purely point-based fitting. However, in a vast amount of medical applications of SSMs, the points of the PDM represent the vertices of a *surface mesh*. This surface mesh is in general only an approximation of a *continuous surface*. Whilst the sparse points  $\mathcal{P}$  are assumed to lie on this continuous surface, in general they do not coincide with the PDM vertices. Hence, matching the *surface*, depending on the PDM deformation parameter  $\boldsymbol{\alpha}$ , is more appropriate. A didactic 2D example is presented in Fig. 5.2.

#### Surface-aligned Covariance Matrices

We now formalise this surface-based fitting method using a GMM with anisotropic covariance matrices that are oriented according to the surface normals. In the GMM, the covariance matrix of each component  $i$ , i.e. each vertex of the PDM, is defined as  $\Sigma_i(\sigma, \boldsymbol{\alpha}) := \sigma^2\mathbf{C}_i(\boldsymbol{\alpha})$ . The scalar parameter  $\sigma^2$  can be seen as a global scaling factor, whereas the matrix  $\mathbf{C}_i(\boldsymbol{\alpha})$  models the anisotropy of the surface structure locally by using a larger variance in the direction of vectors lying in the tangent plane of the PDM surface, compared to the variance along the PDM normal direction (cf. Fig. 5.2 (c)).

Assuming that the surface mesh  $\mathcal{M}$  of the underlying shape of the PDM is given in the form of oriented triangles (cf. Section 5.5), with  $i_2$  and  $i_3$  we denote the index of the “left” and “right” neighbour vertex of  $i$ , respectively. With that, the surface normal at vertex  $i$

is given by

$$n_i(\boldsymbol{\alpha}) = \frac{(y_{i_2}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha})) \times (y_{i_3}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha}))}{\|(y_{i_2}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha})) \times (y_{i_3}(\boldsymbol{\alpha}) - y_i(\boldsymbol{\alpha}))\|}. \quad (5.19)$$

The covariance matrix of vertex  $i$  is defined as

$$\mathbf{C}_i(\boldsymbol{\alpha}) = \left(\frac{1}{\eta} - 1\right)n_i(\boldsymbol{\alpha})n_i^T(\boldsymbol{\alpha}) + \mathbf{I}_3, \quad (5.20)$$

where the parameter  $\eta \geq 1$  weights the variances of vectors along the normal direction compared to the tangential direction. For the covariance matrix  $\Sigma_i(\sigma, \boldsymbol{\alpha})$ , the variance along the normal  $n_i(\boldsymbol{\alpha})$  is given by  $\frac{\sigma^2}{\eta}$ , and the variance in the direction of any vector in the tangent plane is  $\sigma^2$ . As such, for  $\eta = 1$  one obtains the isotropic GMM described in Section 5.6.1, and choosing  $\eta > 1$  achieves the desired behaviour of modelling a larger variance in the tangent plane. A derivation for eq. (5.20) can for example be found in the work of Hill et al. (1995). Note that for  $\eta > 0$ , the matrix  $\mathbf{C}_i(\boldsymbol{\alpha})$  is symmetric and positive definite with the eigenvalues  $\{\frac{1}{\eta}, 1, 1\}$ . The inverse of  $\mathbf{C}_i$ , the *precision matrix*, is given by

$$\mathbf{W}_i(\boldsymbol{\alpha}) := \mathbf{C}_i^{-1}(\boldsymbol{\alpha}) = (\eta - 1)n_i(\boldsymbol{\alpha})n_i^T(\boldsymbol{\alpha}) + \mathbf{I}_3. \quad (5.21)$$

### MAP Solution

The  $Q$  function for the MAP solution now reads

$$Q(\boldsymbol{\alpha}, \sigma, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) = \text{const} - \frac{1}{2}\boldsymbol{\alpha}^T \Lambda^{-1} \boldsymbol{\alpha} - \frac{3P}{2} \ln \sigma^2 \quad (5.22)$$

$$- \frac{1}{2\sigma^2} \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) (p_j - y_i(\boldsymbol{\alpha}))^T \mathbf{W}_i(\boldsymbol{\alpha}) (p_j - y_i(\boldsymbol{\alpha})).$$

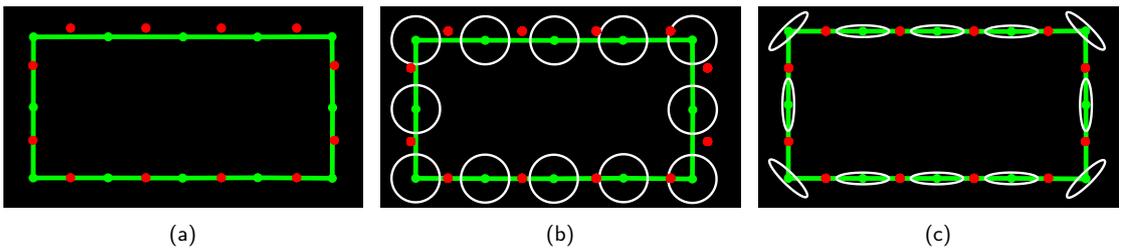


Figure 5.2: Anisotropic covariance matrices to achieve a surface-based fitting. The sparse points  $\mathcal{P}$  are shown in red, the PDM of a rectangle is shown in green, where the green points define the PDM vertices ( $N = 12, M = 2, P = 12$ ). The orientation of the covariance matrices is shown as white ellipse. The objective is to deform the rectangle PDM such that it fits the red points by adjusting  $\boldsymbol{\alpha}$ . The initialisation is shown in (a). Since the red points are sampled between the PDM points (cf. *shape approximation problem* (Hill et al., 1995)), using spherical covariance matrices results in a fit that is even worse than the initialisation (b), whereas using anisotropic covariances results in a more accurate fit (c).

## Chapter 5. Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds

---

As already described, the E-step is solved by evaluating eq. (5.8). Then, the M-step comprises maximising  $Q$  in eq. (5.22) w.r.t.  $\alpha$  and  $\sigma$ . Due to the dependence of  $\mathbf{W}_i$  on  $\alpha$ , finding  $\alpha$  that maximises  $Q$  does not admit a closed-form solution anymore. Instead,  $\alpha$  is now obtained using the BFGS quasi-Newton method (Nocedal and Wright, 2006). The idea is to start with the old value  $\alpha^{(n)}$ , and then iteratively move along directions that increase  $Q$ . Whilst the ordinary Newton method requires the gradient of  $Q$  as well as its Hessian, the BFGS quasi-Newton method uses an approximation of the Hessian that is cheap to compute. We now derive the gradient of  $Q$  w.r.t.  $\alpha$ , i.e.

$$\nabla_{\alpha} Q = \left[ \frac{\partial Q}{\partial \alpha_m} \right]_m. \quad (5.23)$$

For brevity, we write  $\partial \cdot$  to denote the partial derivative  $\frac{\partial \cdot}{\partial \alpha_m}$  w.r.t.  $\alpha_m$ , where the dependence on  $m$  is implicit. First, we note that the cross-product  $u \times v$  of two vectors  $u, v \in \mathbb{R}^3$  can be written as the matrix multiplication  $[u]_{\times} v$ , where the operator  $[\cdot]_{\times} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$  creates a skew-symmetric matrix from its input vector by

$$\left[ \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \right]_{\times} := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \quad (5.24)$$

Introducing

$$b_i(\alpha) := (y_{i_2}(\alpha) - y_i(\alpha)) \times (y_{i_3}(\alpha) - y_i(\alpha)), \quad (5.25)$$

we can write  $n_i(\alpha) = \frac{b_i(\alpha)}{\|b_i(\alpha)\|}$ . Now, by representing the cross product in (5.25) as a matrix product with the notation from (5.24), and by using the product rule, the partial derivative of  $b_i(\alpha)$  is given by

$$\partial b_i(\alpha) = \{\partial [y_{i_2}(\alpha) - y_i(\alpha)]_{\times}\} (y_{i_3}(\alpha) - y_i(\alpha)) \quad (5.26)$$

$$\begin{aligned} &+ [y_{i_2}(\alpha) - y_i(\alpha)]_{\times} \{\partial (y_{i_3}(\alpha) - y_i(\alpha))\} \\ &= [\Phi_{i_2, m} - \Phi_{i, m}]_{\times} (y_{i_3}(\alpha) - y_i(\alpha)) \quad (5.27) \\ &+ [y_{i_2}(\alpha) - y_i(\alpha)]_{\times} (\Phi_{i_3, m} - \Phi_{i, m}). \end{aligned}$$

Moreover,

$$\partial \|b_i(\alpha)\| = \frac{b_i^T(\alpha) \{\partial b_i(\alpha)\}}{\|b_i(\alpha)\|}. \quad (5.28)$$

By using the quotient rule, the partial derivative of  $n_i(\alpha)$  is given by

$$\partial n_i(\alpha) = \frac{\|b_i(\alpha)\| \{\partial b_i(\alpha)\} - b_i(\alpha) \{\partial \|b_i(\alpha)\|\}}{\|b_i(\alpha)\|^2}. \quad (5.29)$$

Using  $\partial n_i(\alpha)$ , we can write

$$\partial \mathbf{W}_i(\alpha) = (\eta - 1) (\{\partial n_i(\alpha)\} n_i^T(\alpha) + n_i(\alpha) \{\partial n_i^T(\alpha)\}). \quad (5.30)$$

Now, given the expression for  $\partial \mathbf{W}_i(\alpha)$ , we can finally compute the partial derivative of  $Q$  w.r.t.  $\alpha_m$ , which is

$$\partial Q(\alpha, \sigma, \alpha^{(n)}, \sigma^{(n)}) = -(\Lambda^{-1})_{m, :} \alpha - \frac{1}{2\sigma^2} \sum_{i, j} \mathfrak{p}(i|p_j, \alpha^{(n)}, \sigma^{(n)}) \quad (5.31)$$

$$[(p_j - y_i(\alpha))^T \{\partial \mathbf{W}_i(\alpha)\} (p_j - y_i(\alpha)) - 2\Phi_{i, m}^T \mathbf{W}_i(\alpha) (p_j - y_i(\alpha))].$$

In order to obtain  $\alpha$ , one option is to run the quasi-Newton procedure until convergence, where one obtains an  $\alpha$  that (locally) maximises  $Q$ . With that, the updates of  $\sigma$  on  $\alpha$  depend on each other and the procedure reverts to the ECM algorithm (Meng and Rubin, 1993), as already stated for the MAP solution in Section 5.6.1. An alternative is to run only a single quasi-Newton step in each M-step. With that, the obtained  $\alpha$  is not a local maximiser of  $Q$ ; however, one has still the guarantee that  $Q$  is non-decreasing. As such, this procedure reverts to the GEM algorithm (Dempster et al., 1977).

Finally, the  $\sigma$ -update for fixed  $\alpha$  is given by

$$\sigma^2 = \frac{1}{3P} \sum_{i,j} \mathbf{p}(i|p_j, \alpha^{(n)}, \sigma^{(n)}) (p_j - y_i(\alpha))^T \mathbf{W}_i(\alpha) (p_j - y_i(\alpha)). \quad (5.32)$$

The pseudocode of the anisotropic GMM fitting procedure is presented in Algorithm 5.

```

Input:  $\bar{\mathbf{x}}, \Phi, \mathcal{P}, \eta, \mathcal{M}$ 
Output:  $\alpha, \sigma^2$ 
Initialise:  $\alpha = \mathbf{0}, \sigma^2 = \frac{1}{3NP} \sum_{i,j} \|p_j - \bar{x}_i\|^2, \mathbf{P} \in \mathbb{R}^{P \times N}, \mathbf{y} = \bar{\mathbf{x}} + \Phi \alpha$ 
1 foreach  $i = 1, \dots, N$  do
   | // compute  $\mathbf{W}_i$ 
2   |  $n_i = \frac{(y_{i2} - y_i) \times (y_{i3} - y_i)}{\|(y_{i2} - y_i) \times (y_{i3} - y_i)\|}$ 
3   |  $\mathbf{W}_i = (\eta - 1)n_i n_i^T + \mathbf{I}_3$ 
4 repeat
   | // E-step
5   | foreach  $j = 1, \dots, P$  do
6   |   |  $t = 0$ 
7   |   | foreach  $i = 1, \dots, N$  do
8   |   |   |  $\mathbf{P}_{ji} = \exp(-\frac{1}{2\sigma^2} (p_j - y_i)^T \mathbf{W}_i (p_j - y_i))$ 
9   |   |   |  $t = t + \mathbf{P}_{ji}$ 
10  |   |  $\mathbf{P}_{j,:} = \frac{1}{t} \mathbf{P}_{j,:}$ 
   | // M-step
11  |  $\alpha = \text{quasi-Newton}(Q, \nabla Q, \alpha)$ 
12  |  $\mathbf{y} = \bar{\mathbf{x}} + \Phi \alpha$ 
13  | foreach  $i = 1, \dots, N$  do
   |   | // compute  $\mathbf{W}_i$ 
14  |   |  $n_i = \frac{(y_{i2} - y_i) \times (y_{i3} - y_i)}{\|(y_{i2} - y_i) \times (y_{i3} - y_i)\|}$ 
15  |   |  $\mathbf{W}_i = (\eta - 1)n_i n_i^T + \mathbf{I}_3$ 
16  |   |  $\sigma^2 = \frac{1}{3P} \sum_{i,j} \mathbf{P}_{ji} (p_j - y_i)^T \mathbf{W}_i (p_j - y_i)$ 
17 until convergence
    
```

**Algorithm 5:** Pseudocode of the anisotropic GMM fitting method. The notation “quasi-Newton( $Q, \nabla Q, \alpha$ )” denotes running the quasi-Newton method for maximising  $Q$  w.r.t.  $\alpha$ , where  $\nabla Q$  is its gradient and the third argument is the initial value of  $\alpha$ . If the GEM approach is used, the quasi-Newton method is run only for a single iteration. Note that the surface mesh  $\mathcal{M}$  is used for the normal computations.

In the next section we introduce an approximation of the  $\alpha$ -update that is a much faster alternative to the quasi-Newton method.

### Fast Approximate Anisotropic GMM

The main idea of our proposed approximation is to use the previous value  $\boldsymbol{\alpha}^{(n)}$  instead of  $\boldsymbol{\alpha}$  for computing the anisotropic covariance matrices  $\mathbf{C}_i(\boldsymbol{\alpha}^{(n)})$  during the  $\boldsymbol{\alpha}$ -update in the M-step. Our key assumption is that the PDM is *well-behaved* in the sense that neighbouring vertices vary smoothly during deformation; thus, *locally* the deformation of an individual triangle is nearly a translation. Since surface normals are *invariant* to translations it follows that  $\|n_i(\boldsymbol{\alpha}) - n_i(\boldsymbol{\alpha}^{(n)})\|$  is small, which implies that  $\|W_i(\boldsymbol{\alpha}) - W_i(\boldsymbol{\alpha}^{(n)})\|$  is also small.

The resulting  $Q$  function using the proposed approximation is now given by

$$\begin{aligned} \tilde{Q}(\boldsymbol{\alpha}, \sigma, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) = & \text{const} - \frac{1}{2} \boldsymbol{\alpha}^T \Lambda^{-1} \boldsymbol{\alpha} - \frac{3P}{2} \ln \sigma^2 \\ & - \frac{1}{2\sigma^2} \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) (p_j - y_i(\boldsymbol{\alpha}))^T \mathbf{W}_i(\boldsymbol{\alpha}^{(n)}) (p_j - y_i(\boldsymbol{\alpha})), \end{aligned} \quad (5.33)$$

where the difference to  $Q$  in eq. (5.22) is that the constant  $\mathbf{W}_i(\boldsymbol{\alpha}^{(n)})$  is now used in place of the function  $\mathbf{W}_i(\boldsymbol{\alpha})$ . As such, the  $\boldsymbol{\alpha}$ -update in the M-step is a quadratic concave problem that can be solved efficiently. The solution for  $\boldsymbol{\alpha}$  is found by solving the linear system  $\tilde{\mathbf{A}}\boldsymbol{\alpha} = \tilde{\mathbf{b}}$ , where  $\tilde{\mathbf{A}} \in \mathbb{R}^{M \times M}$  is given by

$$\tilde{\mathbf{A}} = \sigma^2 \Lambda^{-1} + \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \boldsymbol{\Phi}_i^T \mathbf{W}_i(\boldsymbol{\alpha}^{(n)}) \boldsymbol{\Phi}_i \quad (5.34)$$

and  $\tilde{\mathbf{b}} \in \mathbb{R}^M$  by

$$\tilde{\mathbf{b}} = \sum_{i,j} \mathfrak{p}(i|p_j, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \boldsymbol{\Phi}_i^T \mathbf{W}_i(\boldsymbol{\alpha}^{(n)}) (p_j - \bar{x}_i). \quad (5.35)$$

The pseudocode for this approximate method is similar to Algorithm 5, except for line 11, where the quasi-Newton method is replaced by solving a linear system for the update of  $\boldsymbol{\alpha}$ .

In order to guarantee that the approximate method converges, it is necessary that in the M-step the value of the *exact*  $Q$  in eq. (5.22) is non-decreasing, i.e. the new  $\boldsymbol{\alpha} = \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{b}}$  obtained using  $\tilde{Q}$  must fulfil

$$Q(\tilde{\mathbf{A}}^{-1} \tilde{\mathbf{b}}, \sigma^{(n)}, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}) \geq Q(\boldsymbol{\alpha}^{(n)}, \sigma^{(n)}, \boldsymbol{\alpha}^{(n)}, \sigma^{(n)}). \quad (5.36)$$

For  $\eta = 1$  the method reverts to the isotropic method and condition (5.36) vacuously holds. However, for  $\eta > 1$  this is not true in general. One way to ensure that  $Q$  is non-decreasing is to evaluate the condition in eq. (5.36) in each iteration, and in the case of a violation revert to one of the quasi-Newton methods for the  $\boldsymbol{\alpha}$ -update. Specifically, we consider a single quasi-Newton step for updating  $\boldsymbol{\alpha}$ . We denote the approximate method without this convergence check as *ANISO*, and the method with the convergence check and the quasi-Newton step as fallback as *ANISOc*.

In Fig. 5.3 we illustrate the behaviour of  $Q$  and compare it with  $\tilde{Q}$  for various choices of  $\eta$ .

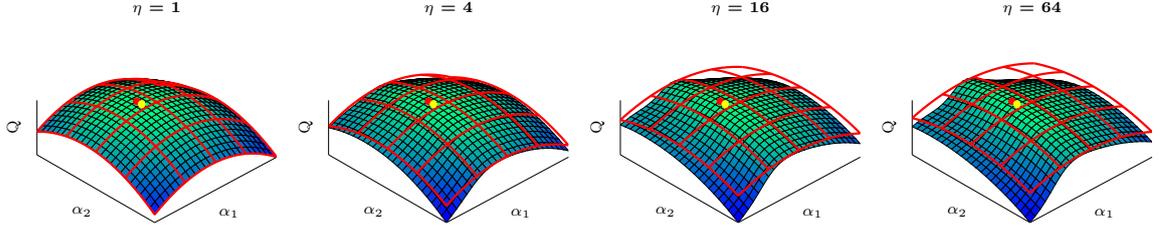


Figure 5.3: Illustration of the behaviour of  $Q$  for various  $\eta$ . The height and the colour of the surface both show the value of  $Q$ , eq. (5.22), depending on  $\alpha_1$  and  $\alpha_2$ . The red grid shows its concave approximation  $\tilde{Q}$  as presented in eq. (5.33). The red dot denotes the value of  $Q$  at  $\alpha^{(n)}$ ; at this position  $Q = \tilde{Q}$ . The yellow dot indicates the maximum of  $Q$ . For the trivial case of  $\eta = 1$  it can be seen that  $Q = \tilde{Q}$  everywhere, whereas an increasing  $\eta$  leads to a larger discrepancy between  $Q$  and  $\tilde{Q}$  as well as to an “increased non-concavity” of  $Q$ .

Note that for visualisation purposes we have chosen  $M = 2$ , whereas in higher-dimensional cases the effect of an increasing  $\eta$  on the non-concavity can be expected to be more severe.

### 5.6.3 Performance Analysis

Table 5.1 summarises the computational complexity of the individual subroutines involved in the presented methods.

Table 5.1: Computational complexity table. The complexity of the  $\alpha$ -update for one iteration of the BFGS quasi-Newton methods is  $\mathcal{O}(M^2)$  plus the complexity of the evaluation of  $Q$  and  $\nabla Q$  (Nocedal and Wright, 2006) (we use  $n$  to denote the number of iterations of the quasi-Newton method). The complexity of the  $\alpha$ -update of the remaining methods comprises the computation of  $\mathbf{A}/\tilde{\mathbf{A}}$  and  $\mathbf{b}/\tilde{\mathbf{b}}$ , as well as solving a linear system of equations of size  $M \times M$ , for which we present the complexity  $\mathcal{O}(M^3)$  due to the matrix inversion involved. Note that in \* we present the complexity for general  $\Lambda$ , for diagonal  $\Lambda$  the quadratic time complexity in  $M$  reduces to linear complexity.

		ECM	GEM	anisotropic ANISOc	ANISO	ISO
update $\mathbf{y}$		$\mathcal{O}(MN)$				
compute $\{\mathbf{W}_i\}$		$\mathcal{O}(N)$				-
E-step		$\mathcal{O}(NP)$				
$\alpha$ -update	evaluate $Q$ , eq. (5.22)	$\mathcal{O}(MN + M^2 + NP)^*$				-
	evaluate $\nabla Q$ , eq. (5.26)	$\mathcal{O}(M^2 + MNP)^*$				-
	construct $\mathbf{A}/\tilde{\mathbf{A}}$	-		$\mathcal{O}(M^2N + NP)$		
	construct $\mathbf{b}/\tilde{\mathbf{b}}$	-		$\mathcal{O}(MN + NP)$		
	total $\alpha$ -update	$\mathcal{O}(n(M^2 + MNP))$	$\mathcal{O}(M^2 + MNP)$	$\mathcal{O}(M^3 + MNP + M^2N)$	$\mathcal{O}(M^3 + M^2N + NP)$	
$\sigma$ -update		$\mathcal{O}(NP)$				
total	(per outer iteration)	$\mathcal{O}(n(M^2 + MNP))$	$\mathcal{O}(M^2 + MNP)$	$\mathcal{O}(M^3 + MNP + M^2N)$	$\mathcal{O}(M^3 + M^2N + NP)$	

In Fig. 5.4 we plot the mean of the normalised value of  $Q$  as a function of the processing time for all four anisotropic fitting methods. For each random run we sample a shape instance by drawing  $\alpha$  (cf. Section 5.4.2), select  $P$  points randomly from the mesh surface (cf. Section 5.7), and run the four methods. The obtained  $Q$  for the four methods are then normalised such that in each run the smallest  $Q$  corresponds to 0 and the largest  $Q$  corresponds to 1 (normalisation w.r.t to all four methods simultaneously). We have found that the single-step quasi-Newton method (GEM) is faster compared to the full quasi-Newton procedure (ECM). Moreover, compared to both quasi-Newton methods, the approximate methods are much faster. Since the ANISOc method makes use of elements both of the GEM and the ANISO method, the total time complexity of the ANISOc method is the combined time complexity for GEM and ANISO (cf. Table 5.1). Nevertheless, in our simulations the ANISOc method comes close to the ANISO method in terms of convergence speed. This is because in the early stages of the iterative procedure the ANISOc method in most cases satisfies condition (5.36) by using approximate M-steps. A violation of condition (5.36) happens more frequently in the later stages of the iterative procedure, where additionally the slower quasi-Newton M-step update has to be performed.

Since the results shown in Fig. 5.4 suggest that the ANISO method is faster whilst being on par with the other methods, for further experiments we chose to use the ANISO method as representative for the anisotropic fitting methods, which allowed us to evaluate more extensive configurations due to its computational efficiency.

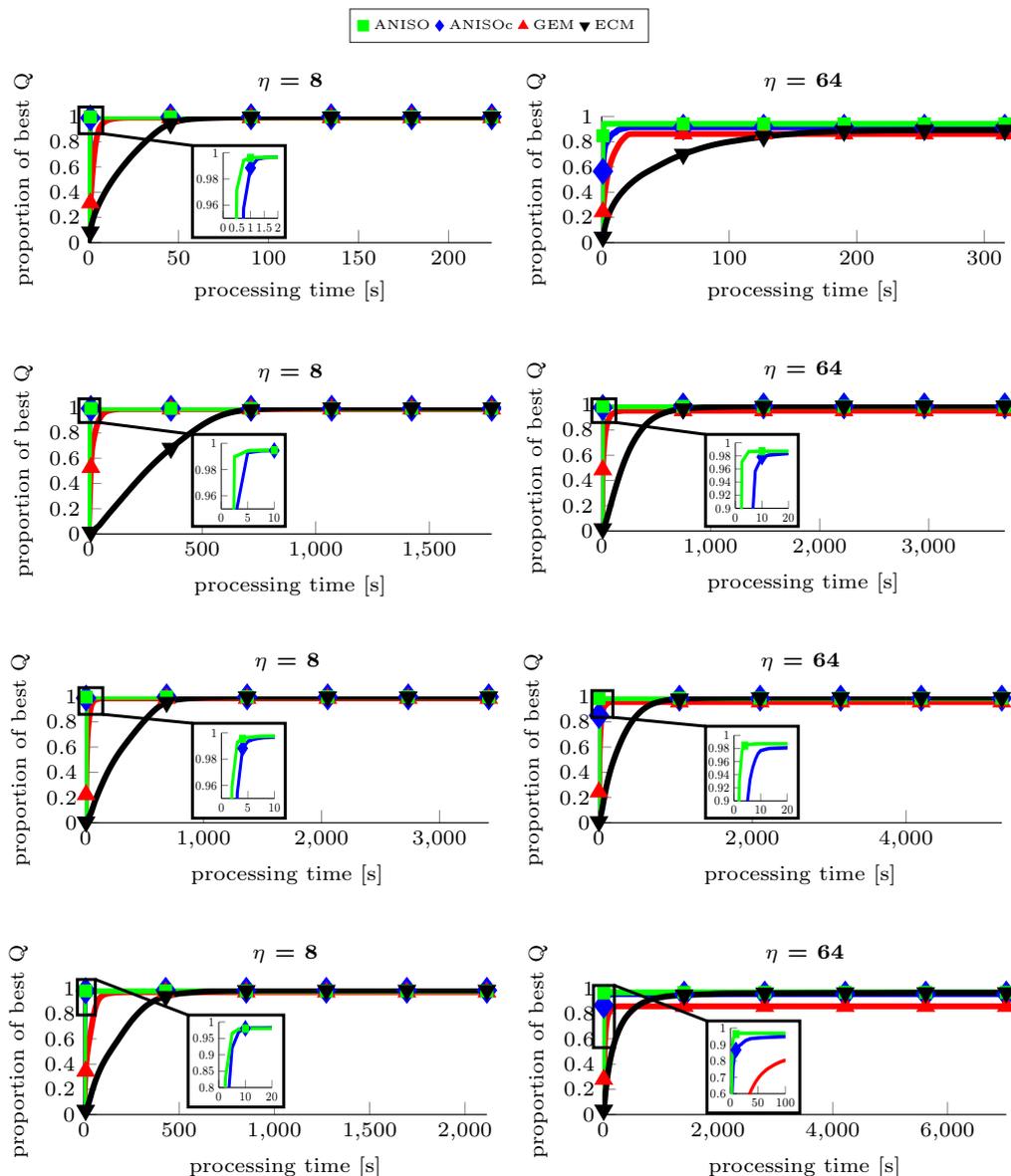


Figure 5.4: Proportion of best value of  $Q$  versus processing time averaged over 100 random runs for the four anisotropic methods for two choices of  $\eta$  (in the columns). In each row a different dataset has been used to produced the results, from top to bottom we show results produced by the brain shapes dataset ( $N = 1792, M = 16, P = 30$ , cf. Section 5.7.2), the femur dataset ( $N = 3800, M = 59, P = 30$ , cf. Section 5.7.3), the tibia dataset ( $N = 4071, M = 59, P = 30$ , cf. Section 5.7.3), and the hip dataset ( $N = 5603, M = 47, P = 30$ , cf. Section 5.7.5).

## 5.7 Experiments

In this section we evaluate the proposed fitting procedures on five datasets, which are summarised in Table 5.2. For the generation of the set of sparse points  $\mathcal{P}$ , we sample sparse points randomly on the shape surfaces. For sampling the random points on the shape *surface*, we first select a triangle from the surface mesh with a probability proportional to its area. Then, we uniformly sample a point lying within the triangle according to the procedure presented by Osada et al. (2002). Moreover, we evaluate noisy versions of these points by adding Gaussian noise with standard deviation  $\tilde{\sigma}$  to each point individually.

In addition to the probabilistic fitting methods presented in this paper, we also use a regularised ICP method as baseline for fitting the PDM to the sparse points, which is outlined in Algorithm 6. Moreover, in our evaluation we compare the ground truth data to the mean shape, i.e. in this setting we do not run any fitting procedure at all, which amounts to setting  $\alpha = \mathbf{0}$ .

The anisotropic method requires to set the parameter  $\eta$  accounting for the amount of anisotropy. We evaluate various choices of  $\eta = 2, 4, 8, 16$  and report results for the value of  $\eta$  that leads to the highest average Dice Similarity Coefficient (cf. Section 5.7.1).

We consider leave-all-in (LAI) and leave-one-out (LOO) experiments. The LAI experiments measure the performance of our method given a perfect model, whereas the LOO experiments evaluate the generalisation ability to unseen data.

### 5.7.1 Evaluation Metrics

Let  $\mathcal{X}$  be the set of points representing the vertices of a high-resolution mesh of the ground truth shape, and let  $\mathcal{Y}$  be the set of the points of the PDM after fitting. We use

$$\mathcal{N}_x = \arg \min_{y \in \mathcal{Y}} \|x - y\|_2 \quad \text{and} \quad (5.37)$$

$$\mathcal{N}_y = \arg \min_{x \in \mathcal{X}} \|x - y\|_2 \quad (5.38)$$

to denote the nearest neighbour of  $x$  in  $\mathcal{Y}$  and of  $y$  in  $\mathcal{X}$ , respectively.

Table 5.2: Size of datasets.

	$N$	$N$ (ds)	$K$	$M$ (LAI)	$M$ (LOO)
brain shapes	1792	371	17	16	96 (kPCA)
femur	3800	759	60	59	58
tibia	4701	814	60	59	58
hip	5603	1120	48	47	46
liver	4542	908	112	111	110

---

```

Input:  $\bar{\mathbf{x}}, \Phi, \mathcal{P}, \Lambda$ 
Output:  $\alpha$ 
Initialise:  $\alpha = \mathbf{0}, \mathbf{p} = \text{vec}([p_1, \dots, p_P]) \in \mathbb{R}^{3P}$ 
1 repeat
2    $\mathbf{y} = \bar{\mathbf{x}} + \Phi \alpha$ 
   // Find nearest neighbours
3    $\mathcal{N} = \text{findNearestNeighbourIndices}(\mathbf{y}, \mathcal{P})$ 
   // Solve linear system for  $\alpha$ 
4    $\mathbf{A} = \Phi_{\mathcal{N},:}$ 
5    $\mathbf{b} = \mathbf{p} - \bar{\mathbf{x}}_{\mathcal{N}}$ 
6    $\alpha = (\mathbf{A}^T \mathbf{A} + \Lambda^{-1})^{-1} \mathbf{A}^T \mathbf{b}$  // Tikhonov regularisation
7 until convergence

```

**Algorithm 6:** Pseudocode of the ICP baseline method. The notation  $\Phi_{\mathcal{N},:}$  and  $\bar{\mathbf{x}}_{\mathcal{N}}$  means selecting the appropriate rows from  $\Phi$  and  $\bar{\mathbf{x}}$  according to the indices of the nearest neighbours  $\mathcal{N}$ .

The (symmetric) Hausdorff distance  $S_{\max}$  between  $\mathcal{X}$  and  $\mathcal{Y}$  is given by

$$S_{\max}(\mathcal{X}, \mathcal{Y}) = \max\{\max_{x \in \mathcal{X}} \|x - \mathcal{N}_x\|_2, \max_{y \in \mathcal{Y}} \|y - \mathcal{N}_y\|_2\}. \quad (5.39)$$

The (symmetric) average distance  $S_{\text{avg}}$  is given by

$$S_{\text{avg}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{\|\mathcal{X}\| + \|\mathcal{Y}\|} \left( \sum_{x \in \mathcal{X}} \|x - \mathcal{N}_x\|_2 + \sum_{y \in \mathcal{Y}} \|y - \mathcal{N}_y\|_2 \right). \quad (5.40)$$

In addition, we use the Dice Similarity Coefficient (DSC) as volumetric metric, which is defined as

$$DSC(\mathcal{V}_x, \mathcal{V}_y) = \frac{2|\mathcal{V}_x \cap \mathcal{V}_y|}{|\mathcal{V}_x| + |\mathcal{V}_y|} \quad (5.41)$$

for the volumetric segmentations  $\mathcal{V}_x$  and  $\mathcal{V}_y$ .

### 5.7.2 Brain Shapes

For deep brain stimulation (DBS) surgery the availability of patient-specific 3D models of relevant brain structures would be highly beneficial both for surgery planning and for intra-operative navigation (Bernard et al., 2012). However, a manual segmentation is too costly for the daily clinical routine use and the clinical acceptance of fully computer-generated 3D models for such highly individualised surgeries stems a problem for automated segmentation methods. One interesting application of our presented shape-aware surface reconstruction method is interactive segmentation. This could be implemented by alternating between the user annotating brain structure boundaries and running our fitting method in order to reconstruct a surface from the user-input.

We consider a multi-object PDM that captures the inter-relation between the brain structures *Substantia Nigra* & *Subthalamic Nucleus* (SN+STN, as compound object), *Nucleus Ruber* (NR), *Thalamus* (Th) and *Putamen* & *Globus Pallidus* (Put+GP, as compound object), where all structures are considered bilaterally. The mean of the PDM is shown in Fig. 5.5 (a). The motivation for using a *multi-object* PDM is to be able to

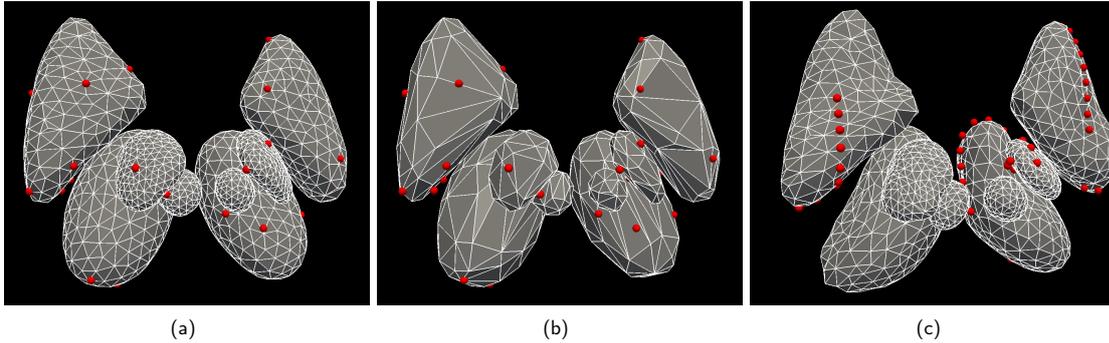


Figure 5.5: Brain shapes dataset. (a) Mean shape with  $N = 1792$  vertices. (b) Downsampled mean shape with 371 vertices. In (a,b)  $P = 36$  sparse points have been randomly drawn from the original surface according to the procedure described in Section 5.7. (c) A shape instance from the training set with partial contours.

reconstruct surfaces of objects, even if for some of the structures no annotation has been provided, e.g. because the image contrast is too low for identifying the boundaries of these particular structures with high confidence.

In Fig. 5.5 (a), random sparse points that are generated according to the procedure described in Section 5.7 are shown. In addition, we use a set  $\mathcal{P}$  that comprises (partial) contours, as shown in Fig. 5.5 (c). These partial contours can for example be drawn by a user in an interactive segmentation setting. For the partial contours we considered two settings,  $\mathbf{c}_1$  and  $\mathbf{c}_2$ . For  $\mathbf{c}_1$ , we have two contours in four of the eight brain structures, as shown in Fig. 5.5 (c). For  $\mathbf{c}_2$ , we have a single contour for each of the eight brain structures. Note that when considering partial contours, for each  $p_j \in \mathcal{P}$  we assume that it is known to which of the eight brain structures it belongs, which is used to constrain the E-step in our fitting methods (and the nearest-neighbour routine for the ICP method). In order to evaluate the robustness of our presented method with respect to noisy input, we also created noisy contours. For that, each partial contour is translated by a random vector that has a zero-mean Gaussian distribution with covariance  $\tilde{\sigma}\mathbf{I}_2$ .

### Pose Normalisation

The PDM is learned from multi-label segmentations that are all represented in a common coordinate system, the ICBM 152 (Fonov et al., 2009) template space (MNI) in our case (more details on the manual annotation and the establishment of correspondences can be found in our previous work (Bernard et al., 2014, 2016c)). The alignment of the patient images into the MNI template space is conducted using the rigid image registration method FLIRT (Jenkinson and Smith, 2001). Hence, thanks to this alignment, the orientation and position are already approximately normalised. Nevertheless, minor pose

variations may still be present between the individual images. Whilst the translational part of these pose differences can be captured precisely by the PDM, the rotational part can only be captured approximately. However, due to the normalisation to the MNI space it can be expected that the rotational part is small and as such the first-order approximation that is captured by the PDM is of sufficient accuracy. Consequently, for a new patient image that is to be segmented, a normalisation to the MNI template space is sufficient.

## Results

For each of the  $K = 17$  training shapes we sample 20 instances of sparse points  $\mathcal{P}$ , leading to  $17 \cdot 20 = 340$  runs per method. As such, the total number of computed surface reconstructions is  $2 \cdot 6 \cdot 6 \cdot 340 = 24,480$  (LOO and LAI, 6 method/noise settings, 6 settings of  $\mathcal{P}$ , where 4 settings of random points on the shape surface and 2 settings of partial contours were considered). For the LAI simulations we used a PCA-based PDM with  $M = 16$  modes. For the LOO experiments we used kernel PCA (kPCA) with  $M = 96$  modes to learn the PDM, which is able to improve the generalisation ability of the PDM from our small training dataset comprising  $K = 17$  shapes (Bernard et al., 2016a). First, we compute all metrics for each of the eight brain structure objects individually, which are then summarised by computing the average for DSC and  $S_{\text{avg}}$ , and by computing the maximum for  $S_{\text{max}}$ .

The results of the ICP method, the ANISO method, and both the ISO and ANISO method with a downsampled PDM are shown in Figs. 5.6 and 5.7. In each column a different metric is shown. In each figure, the first row shows the mean and the second row shows the standard deviation of the respective score. Each subplot shows the value of the metric depending on the considered setting of  $\mathcal{P}$ , as described above. The solid lines indicate the noise-free case, the dotted lines the noisy case with  $\tilde{\sigma} = 2$ , and the dashed line the mean shape baseline (cf. Section 5.7).

As anticipated, the plots confirm that with respect to fitting accuracy an increasing number of measurements  $P$  improves the results. Moreover, it can be seen that the ANISO method (green) outperforms the ICP method (blue) in all cases, where the standard deviation of ICP is much larger. Moreover, the ANISO-ds method (black) outperforms the ISO-ds method (red), which confirms our elaborations in Fig. 5.2 on real data. In all cases, running *any* fitting method is superior compared to simply using the mean shape. Due to the simplicity of the ICP method, its runtime is much lower compared to the proposed fitting methods. However, by using the ANISO method with a downsampled PDM, the runtime can be reduced compared to the original ANISO method, whilst still having superior fitting accuracy compared to ICP.

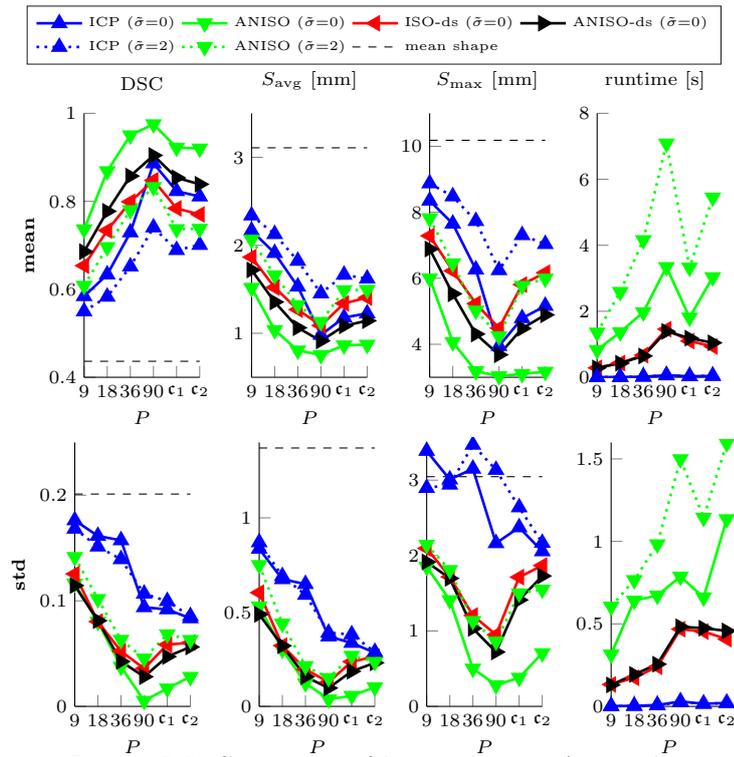


Figure 5.6: Summary of brain shape LAI results.

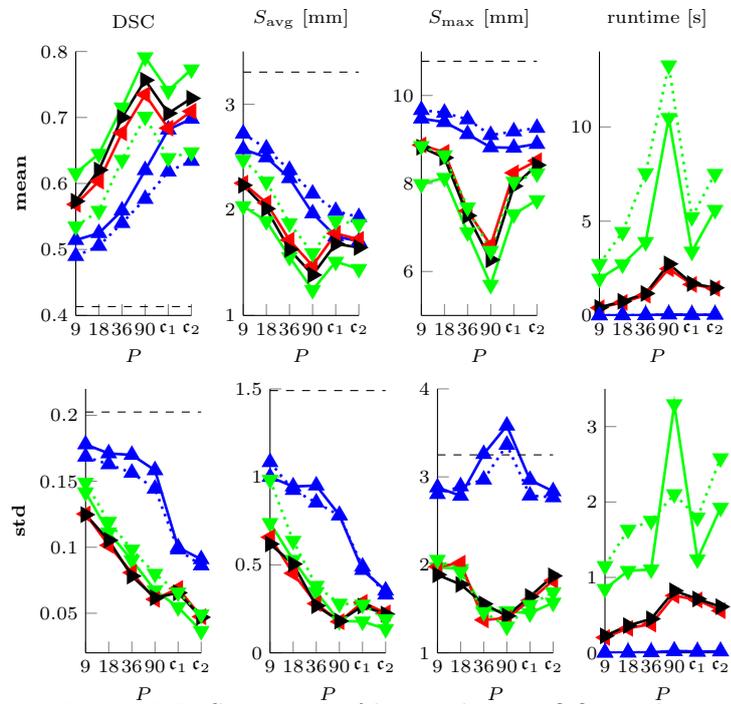


Figure 5.7: Summary of brain shape LOO results.

### 5.7.3 Knee Bones: Femur and Tibia

Knee malalignment is assumed to be associated with knee osteoarthritis (OA), where both *Varus* and *Valgus* malalignment are a potential risk factor. In late stages of knee OA a total knee replacement is often the ultima ratio. In total knee prosthesis, malalignment has a direct impact on the load situation. Knowledge of the accurate bone shape aids implant design as well as patient-specific planning and could also predict OA progression.

However, the manual segmentation of knee bones is tedious, time consuming and highly subjective. Thus, in recent years many semi-automatic and fully automated segmentation methods have been proposed for CT and MRI data of the human knee (Heimann and Meinzer, 2009; Arovitola and Gallo, 2016). In this section we illustrate the potential benefit of our method for knee bone segmentation for both, a femur and a tibia SSM. Given points that were defined in an image on the respective bone’s surface, the proposed shape-aware surface reconstruction method can be used for the task of SSM-based knee bone segmentation. These points could be manually defined or automatically detected by algorithms like SIFT (Lowe, 1999), SURF (Bay et al., 2008) or machine learning techniques in general (Yang et al., 2015; Xue et al., 2015).

#### Pose Normalisation

For the femur and tibia dataset we assumed that the pose has already been normalised and we directly worked in the space of the SSM. In practice, this can for example be tackled in a similar manner as by Seim et al. (2010), who proposed an automated SSM-based knee bone segmentation, where initially the model is positioned inside the three dimensional CT or MR image via Generalised Hough Transform (Ballard, 1981).

#### Results

We present experiments for a PDM of the femur with  $N = 3800$  points (cf. Fig. 5.8 (a)) and a PDM of the tibia with  $N = 4701$  points (cf. Fig. 5.8 (c)). Additionally, we evaluated the ANISO method using the downsampled PDMs, denoted ANISO-ds, where only a subset of the original PDM vertices are used (cf. Fig. 5.8 (b) and Fig. 5.8 (d)). Random sparse points are generated according to the procedure described in Section 5.7, where for each training shape 10 instances of sparse points  $\mathcal{P}$  are sampled. In Fig. 5.8 such random instances of  $\mathcal{P}$  are shown for the mean shapes of both bones.

LAI and LOO experiments have been carried out for the femur and tibia model evaluating the same metrics as before. For both models our PCA-based PDM has  $K = 60$  training shapes with  $M = 59$  modes for the LAI experiments, and has  $K = 59$  training shapes with  $M = 58$  modes for the LOO experiments.

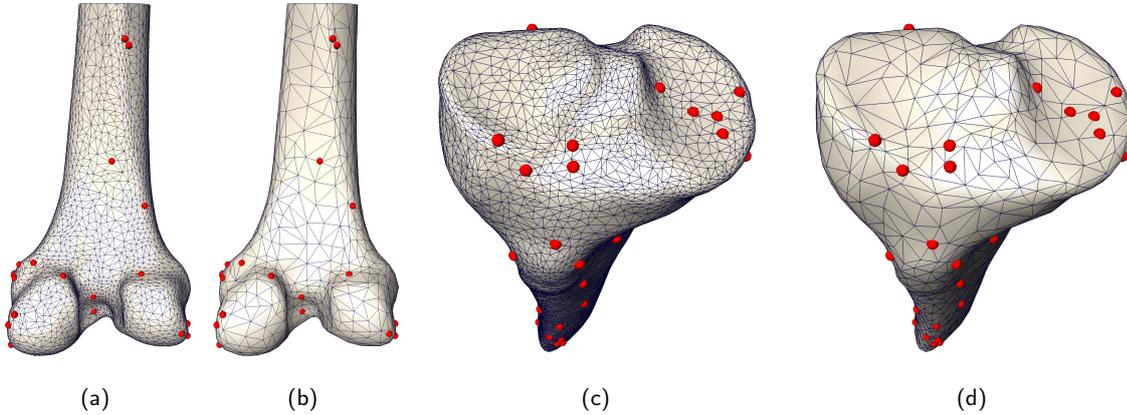


Figure 5.8: Femur and tibia datasets. (a) Femur mean shape with  $N = 3800$  vertices. (b) Downsampled mean shape with 759 vertices. (c) Tibia mean shape with  $N = 4701$  vertices. (d) Downsampled mean shape with 814 vertices. For both bone models  $P = 36$  sparse points have been randomly drawn from the original surface according to the procedure described in Section 5.7.

A summary of the results comparing the ICP method, the ANISO method and the ANISO-ds method are shown in Figs. 5.9 and 5.10 for the femur and in Figs. 5.11 and 5.12 for the tibia. In each of the first three columns a different metric is shown. The last column shows the runtime. The upper row depicts the mean and the lower row the standard deviation. Each subplot shows the value of the metric as a function of the number of sparse points  $P$ .

It can be seen that for both bone PDMs if only  $P = 9$  points are available, the ICP method's results appear slightly better than the ANISO method's results. Once more points become available, both anisotropic methods outperform the ICP method. Surprisingly, the ANISO-ds method, which uses a downsampled PDM, outperforms the ANISO method for  $P = 9$ . We assume that this is because the original PDMs with  $N = 3800$  vertices for the femur and  $N = 4701$  vertices for the tibia contain fine local details that lead to an overfitting when reconstructing the surface from only  $P = 9$  points. In contrast, the downsampled PDM contains less details that may impede the surface reconstruction.

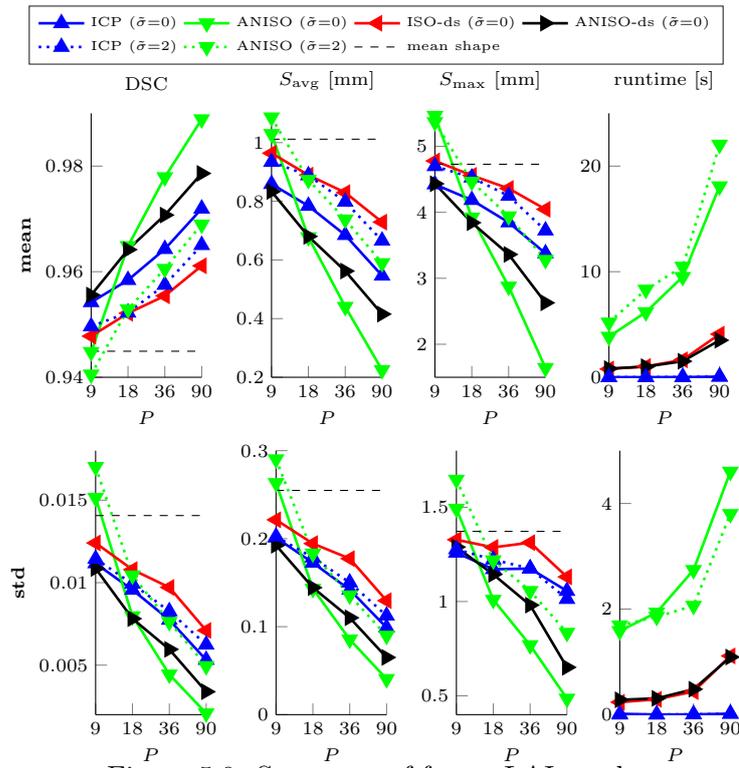


Figure 5.9: Summary of femur LAI results.

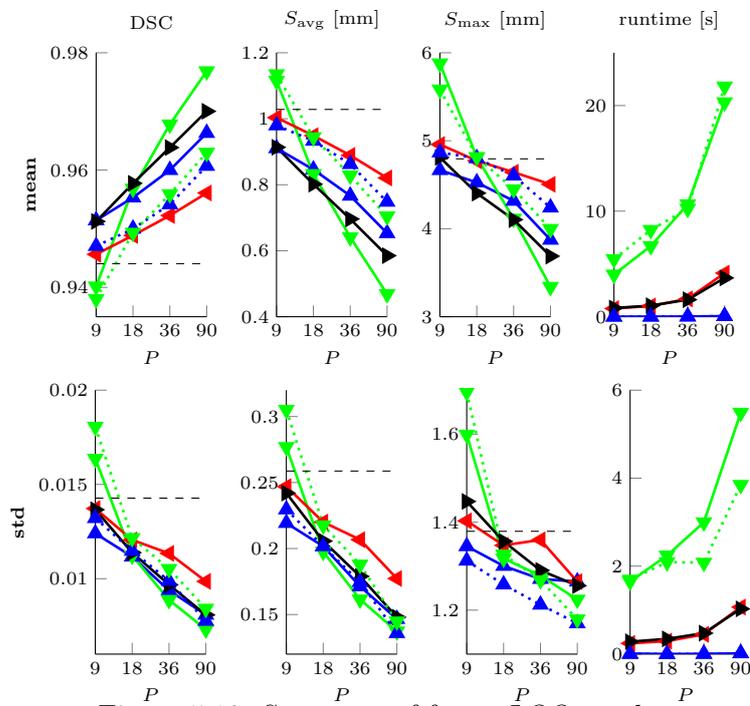


Figure 5.10: Summary of femur LOO results.

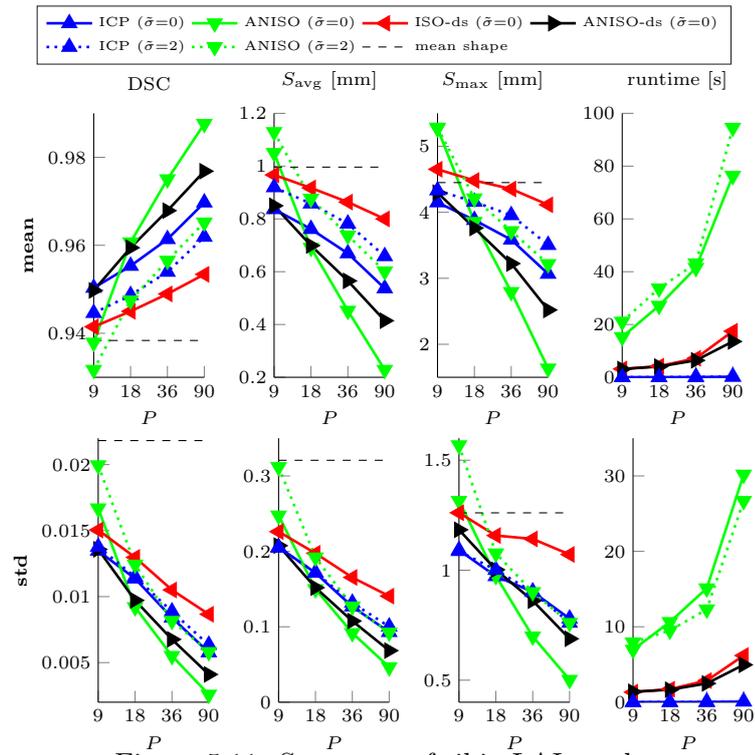


Figure 5.11: Summary of tibia LAI results.

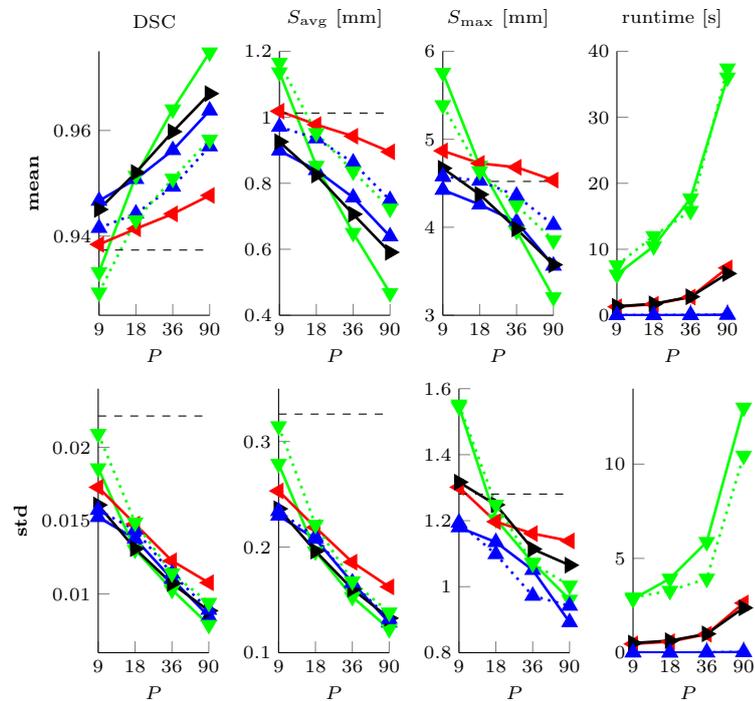


Figure 5.12: Summary of tibia LOO results.

### 5.7.4 Liver

Automated liver segmentation is beneficial for many applications, such as surgery planning (e.g. prior to hepatic resection (Gao et al., 1996)), lesion detection, automated detection of liver cancer, and the development of radiation treatment programs (Campadelli et al., 2009). However, for up to 1000 axial CT slices this is tedious work (Zhou et al., 2006) that can require more than one hour (Campadelli et al., 2009). Similar to the previous section on knee bone segmentation, our proposed method could improve the accuracy of the automated liver volume segmentation. Again, we assume, that the pose has already been normalised.

### Results

We carried out LAI and LOO experiments using a liver PDM with  $N = 4542$  points (cf. Fig. 5.13 (a)). The PDM has  $K = 112$  training shapes with  $M = 111$  modes for the LAI experiments, and has  $K = 111$  training shapes with  $M = 110$  modes for the LOO experiments. Figure 5.13 (b) shows the downsampled PDM used for the ANISO-ds experiments and again random instances of  $\mathcal{P}$  are shown for the original and downsampled mean shape. For each training shape 10 instances of sparse points  $\mathcal{P}$  are sampled.

A summary of the results comparing the ICP method, the ANISO method and the ANISO-ds method are shown in Figs. 5.14 and 5.15. In each of the first three columns a different metric is shown. The last column shows the runtime. The upper row depicts the mean and the lower row the standard deviation. Each subplot shows the value of the metric as a function of the number of sparse points  $P$ .

For both settings, LAI and LOO, without any Gaussian disturbance ( $\tilde{\sigma} = 0$ ) the anisotropic method is outperforming the ICP method with respect to the mean DSC regardless of how many random points have been sampled. Especially for 36 and 90 points the accuracy of the anisotropic method becomes increasingly superior compared to ICP. Considering the random points disturbed by Gaussian noise with  $\tilde{\sigma} = 2\text{cm}$ , the ICP method yields better results for 9 points and comparable results for 18 points. With this disturbance, for the LAI and the LOO experiments at least 36 points seem to be necessary for the anisotropic method to achieve better results than the ICP method with respect to the DSC.

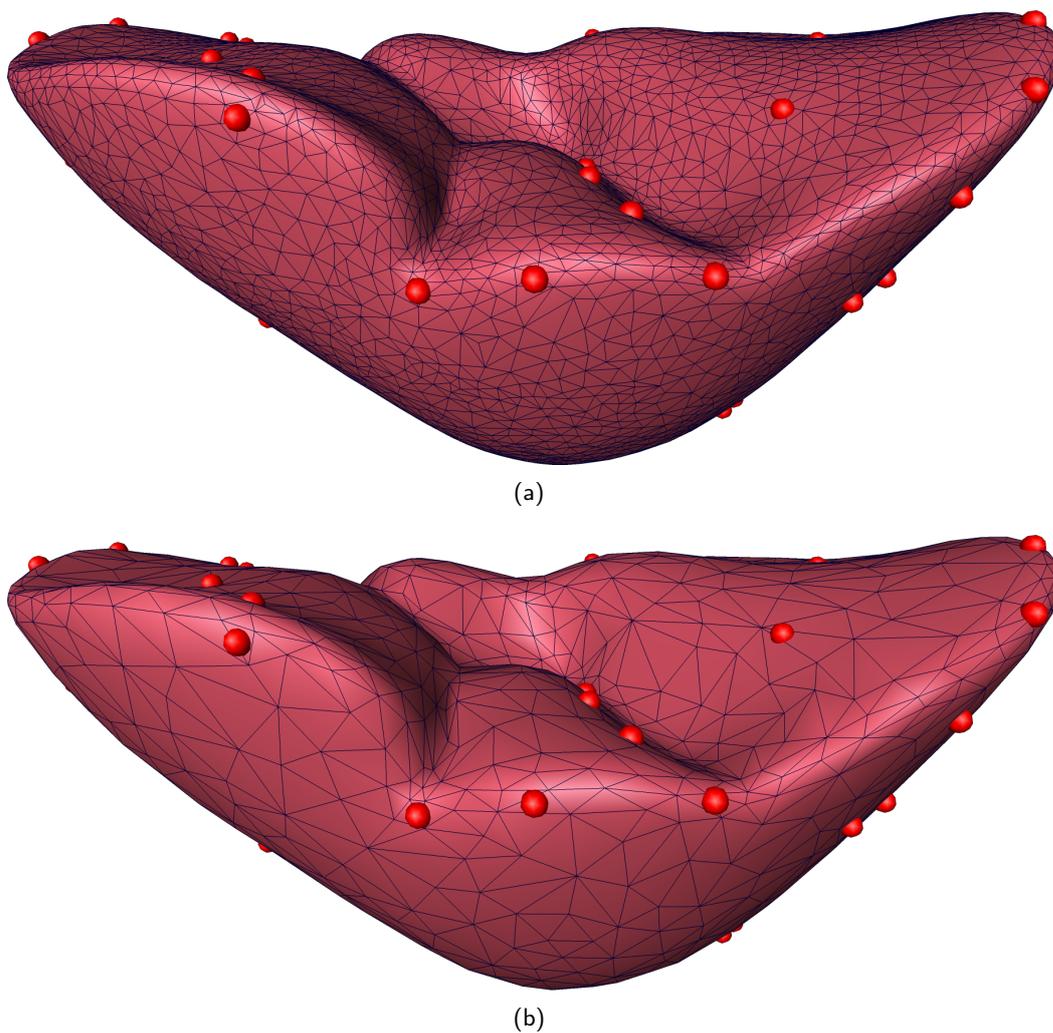


Figure 5.13: Liver dataset. (a) Mean shape with  $N = 4542$  vertices. (b) Downsampled mean shape with 908 vertices.  $P = 36$  sparse points have been randomly drawn from the original surface according to the procedure described in Section 5.7.

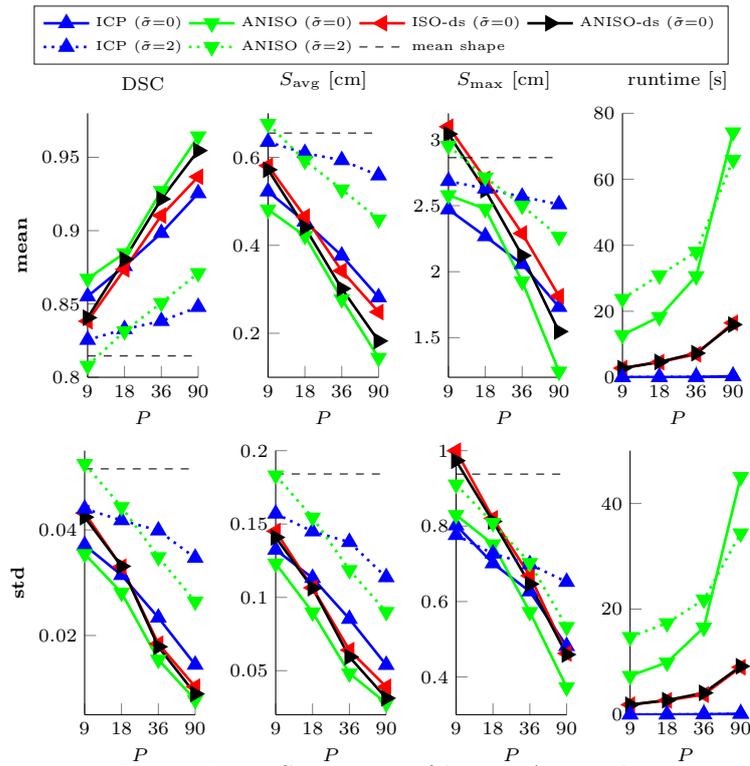


Figure 5.14: Summary of liver LAI results.

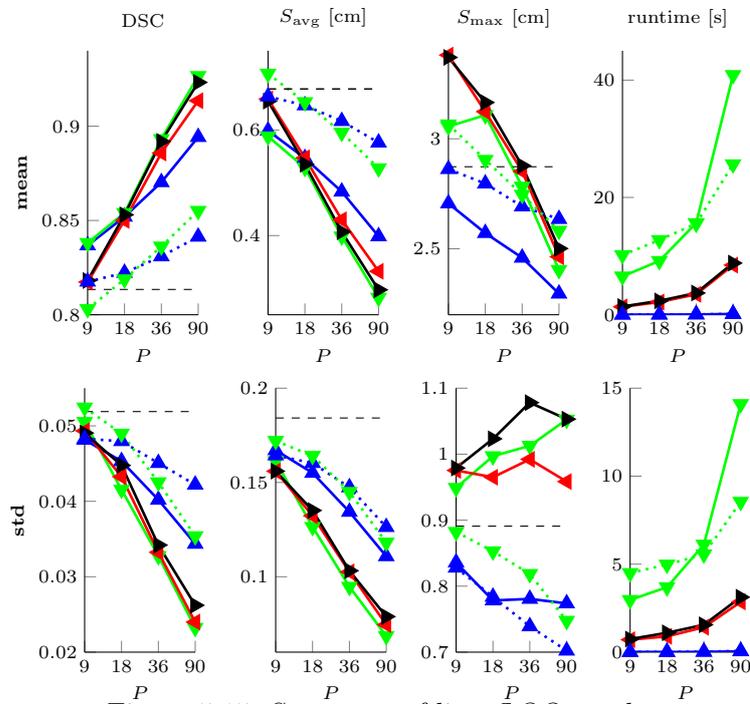


Figure 5.15: Summary of liver LOO results.

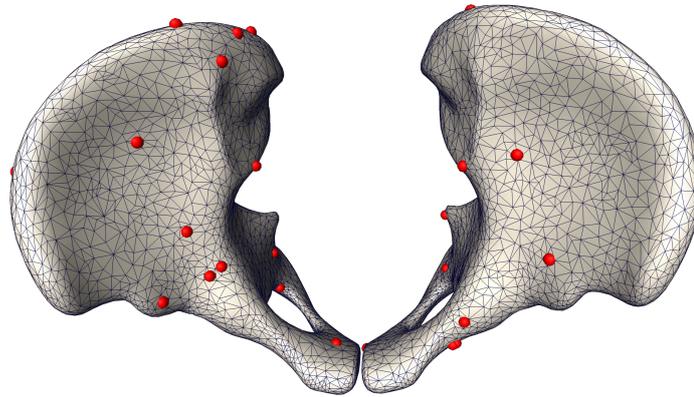
### 5.7.5 Hip

For total hip replacement computer-assisted pre-operative planning systems are already used to improve the positioning of the acetabular cup to avoid dislocation and to minimize wear. For these systems a prerequisite is often the 3D segmentation of the pelvis. Differentiation between the acetabulum and the (arthritic) femoral head is a challenging task. Our approach might be used to improve segmentation accuracy not only for the knee bone and liver segmentation, but also for the pelvis segmentation.

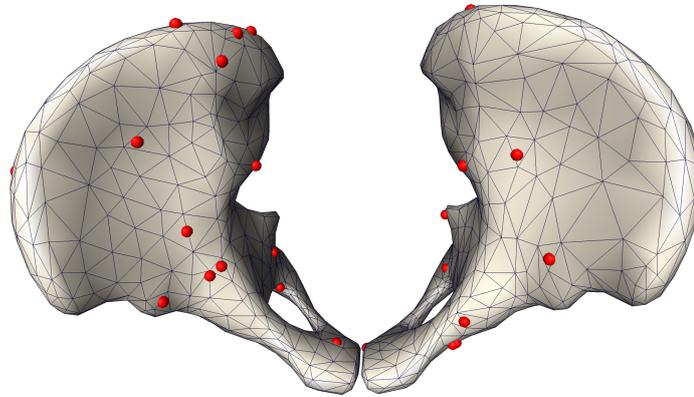
#### Results

We carried out LAI and LOO experiments using a hip PDM with  $N = 5603$  points (cf. Fig. 5.16 (a)). The PDM has  $K = 48$  training shapes with  $M = 47$  modes for the LAI experiments, and has  $K = 47$  training shapes with  $M = 46$  modes for the LOO experiments. Figure 5.16 (b) shows the downsampled PDM used for the ANISO-ds experiments and again random instances of  $\mathcal{P}$  are shown for the original and downsampled mean shape. For each training shape 10 instances of sparse points  $\mathcal{P}$  are sampled.

A summary of the results comparing the ICP method, the ANISO method and the ANISO-ds method are shown in Figs. 5.17 and 5.18. For both settings, LAI and LOO, without any Gaussian disturbance ( $\tilde{\sigma} = 0$ ) the anisotropic method outperforms the ICP method with respect to the mean Dice Similarity Coefficient for more than 9 points. Again, the accuracy is increasing with more sampled points and the difference between the anisotropic method and ICP becomes more clearly visible for 36 and 90 points. Considering the random points disturbed by a Gaussian of 2 mm ( $\tilde{\sigma} = 2$ ), the ICP method yields again better results for 9 points and comparable results for 18 points for both experimental settings.



(a)



(b)

Figure 5.16: Hip dataset. (a) Mean shape with  $N = 5603$  vertices. (b) Downsampled mean shape with 1120 vertices.  $P = 36$  sparse points have been randomly drawn from the original surface according to the procedure described in Section 5.7.

## Chapter 5. Shape-aware Surface Reconstruction from Sparse 3D Point-Clouds

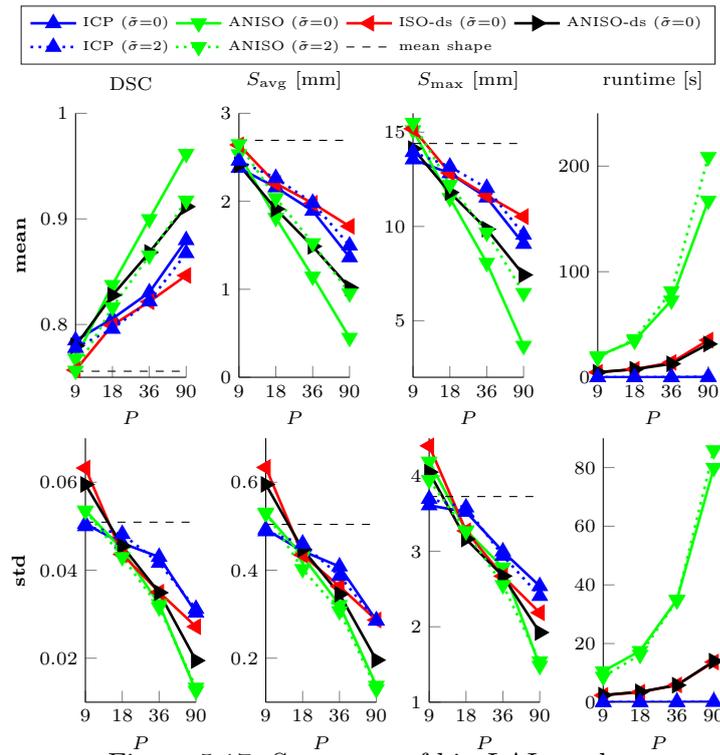


Figure 5.17: Summary of hip LAI results.

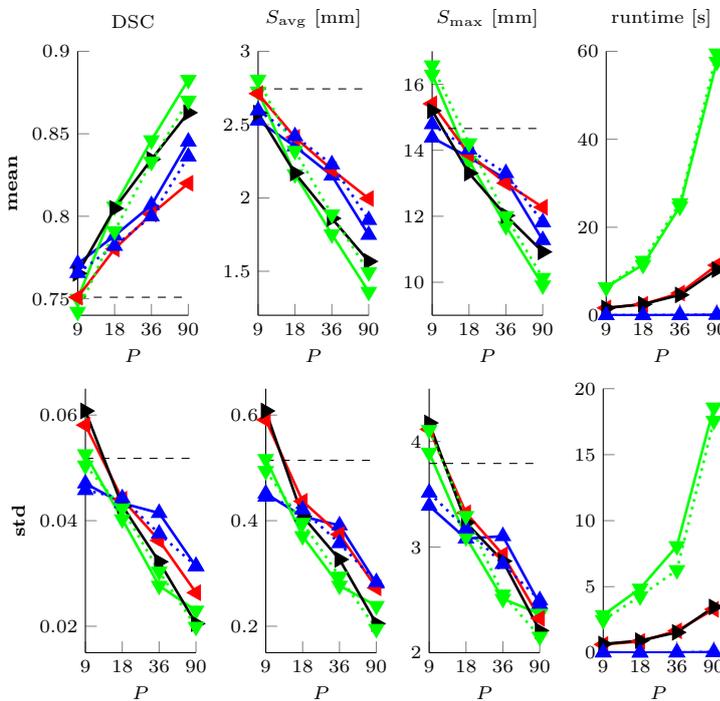


Figure 5.18: Summary of hip LOO results.

## 5.8 Conclusion and Outlook

In this paper we have presented a methodology for a shape-aware surface reconstruction from sparse surface points. The proposed methodology is superior compared to the standard approach of ICP with respect to accuracy and robustness on a wide range of datasets. In this method, the likely shape of the object that is to be reconstructed is captured by a PDM associated with a surface mesh. By interpreting the available sparse surface points as samples of a GMM, the surface reconstruction task is cast as maximisation of the posterior likelihood, which we tackle by variants of the EM algorithm. In order to achieve a surface-based fitting, we use a GMM with anisotropic covariance matrices, which are oriented by the surface normals at the PDM points. However, this results in a non-concave optimisation problem that is to be solved in each M-step. We deal with this by maximising a concave approximation that considers the surface normals of the PDM computed from the previous value of the shape deformation parameter. We have presented an intuitive explanation why this approximation makes sense with the assumption that neighbour PDM vertices vary smoothly and the fact that surface normals are invariant to translations. We empirically demonstrated that finding a global maximum of this approximation leads to better results compared to finding a local optimum during the exact (non-concave) M-step. Moreover, our proposed concave approximation results in an algorithm that has the same time complexity as the isotropic fitting procedure.

The proposed surface reconstruction method deals exclusively with shape deformations. Thus, the normalisation of the pose must be solved a-priori in an application-dependent manner. In the example of the multi-object brain shape reconstruction we dealt with this issue using rigid image registration in order to align the data into a common coordinate system. Dealing with the limitation of not explicitly considering a rigid transformation in order to model the pose of the object is the next step in order to achieve an even broader applicability. Whilst in principal one can formulate an analogous problem that considers the pose, the resulting problem is much more difficult to solve. This is because a simultaneous maximisation must be performed with respect to the rigid transformation and the shape deformation parameter. This is usually done iteratively, as in Active Shape Model search (Cootes and Taylor, 1992). With that, particular challenges to be dealt with are that the resulting surface reconstruction procedure would be much more sensitive to unwanted local optima as well as much slower.

In our evaluation we considered a wide range of datasets in order to demonstrate its general applicability, where dealing with certain application-specific aspects is out of scope of this work. One relevant question to be answered when applying the proposed methodology to medical imaging problems is how to obtain the 3D surface points. The analysis of which points are most useful for the reconstruction is one interesting direction for future work.

## **Acknowledgements**

The authors gratefully acknowledge the financial support by the Fonds National de la Recherche, Luxembourg (6538106, 8864515, 9169303), by the German federal ministry of education and research (BMBF), grant no. 01EC1408B, and by the Einstein Center for Mathematics (ECMath), Berlin.

# Transformation Synchronisation **Part II**



# 6 A Solution for Multi-Alignment by Transformation Synchronisation

## Abstract

The alignment of a set of objects by means of transformations plays an important role in computer vision. Whilst the case for only two objects can be solved globally, when multiple objects are considered usually iterative methods are used. In practice the iterative methods perform well if the relative transformations between any pair of objects are free of noise. However, if only noisy relative transformations are available (e.g. due to missing data or wrong correspondences) the iterative methods may fail. Based on the observation that the underlying noise-free transformations can be retrieved from the null space of a matrix that can directly be obtained from pairwise alignments, this paper presents a novel method for the synchronisation of pairwise transformations such that they are transitively consistent. Simulations demonstrate that for noisy transformations, a large proportion of missing data and even for wrong correspondence assignments the method delivers encouraging results.

## 6.1 Introduction

The alignment of a set of objects by means of transformations plays an important role in the field of computer vision and recognition. For instance, for the creation of Statistical Shape Models (SSMs) (Cootes and Taylor, 1992) training shapes are initially aligned for removing pose differences in order to only model shape variability.

The most common way of shape representation is by encoding each shape as a point-cloud. In order to be able to process a set of shapes it is necessary that correspondences between all shapes are established. Whilst there is a vast amount of research in the field of shape correspondences (for an overview see the survey papers by (Heimann and Meinzer, 2009; van Kaick et al., 2011)), in this paper we focus on the alignment of shapes and we assume

## Chapter 6. A Solution for Multi-Alignment by Transformation Synchronisation

---

that correspondences have already been established.

The alignment of two objects by removing location, scale and rotation is known as *Absolute Orientation Problem (AOP)* (Horn et al., 1988) or *Procrustes Analysis* (Gower and Dijksterhuis, 2004). For the AOP there are various closed-form solutions, among them methods based on Singular Value Decomposition (SVD) (Arun et al., 1987; Schönemann, 1966); based on eigenvalue decomposition (Horn et al., 1988); based on unit quaternions (Horn, 1987) or based on dual quaternions (Walker et al., 1991). A comparison of these methods (Eggert et al., 1997) has revealed that the accuracy and the robustness of all methods are comparable.

The alignment of more than two objects is known as *Generalised Procrustes Analysis (GPA)*. Whilst a computationally expensive global solution for GPA in two and three dimensions has been presented by Pizarro and Bartoli (2011), the most common way for solving the GPA is to align the objects with a reference object. However, fixing any of the objects as reference induces a bias. An unbiased alternative is to align all objects with the adaptive mean object as reference. An iterative algorithm then alternately updates the reference object and estimates the transformations aligning the objects. The iterative nature of these methods constitutes a problem if the relative transformation between any pair of objects is noisy. This is for example the case if data is missing, correspondences are wrong or if the transformations are observed by independent sensors (e.g. non-communicating robots observe each other). Noisy relative transformations can be characterised by transitive inconsistency, i.e. transforming  $A$  to  $B$  and  $B$  to  $C$  might lead to a different result than transforming  $A$  directly to  $C$ .

This paper presents a novel method for synchronising the set of all pairwise transformations in such a way that they globally exhibit transitive consistency. Experiments demonstrate the effectiveness of this method in denoising noisy pairwise transformations. Furthermore, using this novel method the GPA is solved in an unbiased manner in closed-form, i.e. non-iterative. Transformation synchronisation is applied to solve the GPA with missing data as well as with wrong correspondence assignments and results in superior performance compared to existing methods.

Our main contribution is a generalisation of the techniques presented by Chaudhury et al. (2013); Hadani and Singer (2011a,b); Singer and Shkolnisky (2011), who have introduced a method for minimising global self-consistency errors between pairwise orthogonal transformations based on eigenvalue decomposition and semidefinite programming. With permutation transformations being a subset of orthogonal transformations, Pachauri et al. (2013) demonstrate that the method by Singer et al. is also able to effectively synchronise permutation transformations for globally consistent matchings.

In our case, rather than considering the special case of orthogonal matrices, we present a

synchronisation method for invertible linear transformations. Furthermore, it is demonstrated how this method can be applied for the synchronisation of similarity, euclidean and rigid transformations, which are of special interest for the groupwise alignment of shapes.

Whilst the proposed synchronisation method is applicable in many other fields where noisy pairwise transformations are to be denoised (e.g. groupwise image registration or multi-view registration), in this paper GPA is used as illustrating example.

## 6.2 Methods

For the presentation of our novel transformation synchronisation method the notation and some foundations are introduced first. Subsequently, a formulation for the case of perfect information is given. Motivated by these elaborations, a straightforward extension to handle noisy pairwise transformations is presented. Finally, various types of transformations are discussed.

### 6.2.1 Notation and Foundations

$\mathbf{X}_i, \mathbf{X}_j \in \mathbb{R}^{n \times d}$  are matrices representing point-clouds with  $n$  points in  $d$  dimensions where in the following all  $\mathbf{X}_i$  are simply referred to as point-clouds. Let  $\mathbf{I}$  be the identity matrix and  $\mathbf{0}$  be the vector containing only zeros, both having appropriate dimensions according to their context. The Frobenius norm is denoted by  $\|\cdot\|_F$ . Let  $T_{ij} \in \mathbb{R}^{d \times d}$  be an invertible transformation matrix aligning point-cloud  $\mathbf{X}_i$  with  $\mathbf{X}_j$  (for all  $i, j = 1, \dots, k$ ), where  $T_{ij} = T_{ji}^{-1}$ . Furthermore,  $\mathcal{T} = \{T_{ij}\}_{i=1, j=1}^k$  is the set of all  $k^2$  pairwise transformations.

A desirable property of the set of transformations  $\mathcal{T}$  is that it complies with the following transitive consistency condition:

**Definition 3.** *The set of relative transformations  $\mathcal{T}$  is said to be transitively consistent if*

$$T_{ij}T_{jl} = T_{il} \quad \text{for all } i, j, l = 1, \dots, k.$$

Definition 3 states that the transformation from  $i$  to  $j$  followed by the transformation from  $j$  to  $l$  must be the same as directly transforming from  $i$  to  $l$ .

**Lemma 4.** *The set of relative transformations  $\mathcal{T}$  is transitively consistent if and only if there is a set of invertible transformations  $\{\bar{T}_i\}_{i=1}^k$  such that*

$$T_{ij} = \bar{T}_i \bar{T}_j^{-1} \quad \text{for all } i, j = 1, \dots, k.$$

*Proof.* For the sake of completeness a proof is provided.

## Chapter 6. A Solution for Multi-Alignment by Transformation Synchronisation

---

“ $\Leftarrow$ ”: Transitive consistency of  $\mathcal{T}$  follows directly from the definition  $T_{ij} = \bar{T}_i \bar{T}_j^{-1}$ , since for all  $i, j, l = 1, \dots, k$  it holds that

$$T_{il} = \bar{T}_i \bar{T}_l^{-1} = \bar{T}_i \mathbf{I} \bar{T}_l^{-1} \quad (6.1)$$

$$= \bar{T}_i (\bar{T}_j^{-1} \bar{T}_j) \bar{T}_l^{-1} \quad (6.2)$$

$$= (\bar{T}_i \bar{T}_j^{-1}) (\bar{T}_j \bar{T}_l^{-1}) \quad (6.3)$$

$$= T_{ij} T_{jl}. \quad (6.4)$$

“ $\Rightarrow$ ”: In the rest of the proof we direct our attention towards the necessity of the existence of the  $\bar{T}_i$  transformations.

First of all, if the transformations in  $\mathcal{T}$  are transitively consistent

$$T_{ii} = \mathbf{I} \quad \text{for all } i = 1, \dots, k. \quad (6.5)$$

This follows by the fact that the  $T_{ii}$  needs to be invertible while satisfying, by Definition 3, that  $T_{ii} T_{ii} = T_{ii}$ .

Let  $\bar{T}'_i = T_{i1}$  for all  $i$ . Now we show that  $\bar{T}'_i$  are such  $\bar{T}_i$  matrices we seek. Since, by using (6.5),  $\bar{T}'_1 = \mathbf{I}$ , we can write  $T_{i1} = \bar{T}'_i = \bar{T}'_i (\bar{T}'_1)^{-1}$ .

Now for any  $T_{ij}$ , we can use that

$$T_{1i} T_{ij} = T_{1j}. \quad (6.6)$$

Thus,

$$T_{ij} = T_{1i}^{-1} T_{1j} = T_{i1} T_{j1}^{-1} = \bar{T}'_i (\bar{T}'_j)^{-1}. \quad (6.7)$$

□

### 6.2.2 Perfect Information

Due to Lemma 4, there is a reference coordinate frame, denoted by  $\star$ , from which there are  $T_{i\star}$  transformations such that  $T_{ij} = T_{i\star} T_{\star j}$  for all  $i, j$ . Note that the reference coordinate frame is merely used as a tool for deriving our method and it is irrelevant what the actual reference frame is. Let us introduce

$$\mathbf{W} = [T_{ij}] = \begin{bmatrix} T_{11} & \cdots & T_{1k} \\ \vdots & \ddots & \vdots \\ T_{k1} & \cdots & T_{kk} \end{bmatrix} \quad (6.8)$$

$$= \begin{bmatrix} T_{1\star} T_{\star 1} & \cdots & T_{1\star} T_{\star k} \\ \vdots & \ddots & \vdots \\ T_{k\star} T_{\star 1} & \cdots & T_{k\star} T_{\star k} \end{bmatrix} \quad (6.9)$$

$$= \begin{bmatrix} T_{1\star}T_{1\star}^{-1} & \cdots & T_{1\star}T_{k\star}^{-1} \\ \vdots & \ddots & \vdots \\ T_{k\star}T_{1\star}^{-1} & \cdots & T_{k\star}T_{k\star}^{-1} \end{bmatrix} \quad (6.10)$$

$$= \mathbf{U}_1 \mathbf{U}_2, \quad (6.11)$$

where

$$\mathbf{U}_1 = \begin{bmatrix} T_{1\star} \\ T_{2\star} \\ \vdots \\ T_{k\star} \end{bmatrix} \quad \text{and} \quad \mathbf{U}_2 = [T_{1\star}^{-1}, T_{2\star}^{-1}, \dots, T_{k\star}^{-1}].$$

Using this notation, finding either  $\mathbf{U}_1$  or  $\mathbf{U}_2$  (up to an invertible linear transformation) gives the transitively consistent transformations.

**Definition 4.** Let  $\mathbf{A} \in \mathbb{R}^{p \times q}$ . The set

$$\text{im}(\mathbf{A}) = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^q\}$$

is the column space of  $\mathbf{A}$  and the set

$$\text{ker}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^q \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$$

is the null space of  $\mathbf{A}$ .

Note that due to the invertability of  $T_{i\star}$  (for all  $i = 1, \dots, k$ ) it holds that the matrix  $\mathbf{U}_1$  has rank  $d$  and thus the dimensionality of the column space  $\text{im}(\mathbf{U}_1)$  of  $\mathbf{U}_1$  is exactly  $d$ .

**Proposition 1.** Let  $\mathbf{Z} = \mathbf{W} - k\mathbf{I}$ . The linear subspace  $\text{ker}(\mathbf{Z})$  has dimension  $d$  and is equal to  $\text{im}(\mathbf{U}_1)$ .

*Proof.* First it is shown that the columns of  $\mathbf{U}_1$  are contained in the null space of  $\mathbf{Z}$ , i.e.  $\text{im}(\mathbf{U}_1) \subseteq \text{ker}(\mathbf{Z})$ , and then it is shown that the null space of  $\mathbf{Z}$  has exactly dimension  $d$ .

Note that  $\mathbf{W}\mathbf{U}_1 = k\mathbf{I}$ , which we will make use of shortly. Multiplication of  $\mathbf{U}_1$  on the right to  $\mathbf{W} = \mathbf{U}_1\mathbf{U}_2$  gives

$$\mathbf{W}\mathbf{U}_1 = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_1 = \mathbf{U}_1k\mathbf{I} \quad (6.12)$$

$$\Leftrightarrow \mathbf{W}\mathbf{U}_1 = k\mathbf{U}_1 \quad (6.13)$$

$$\Leftrightarrow \mathbf{W}\mathbf{U}_1 - k\mathbf{U}_1 = \mathbf{0} \quad (6.14)$$

$$\Leftrightarrow (\mathbf{W} - k\mathbf{I})\mathbf{U}_1 = \mathbf{0} \quad (6.15)$$

$$\Leftrightarrow \mathbf{Z}\mathbf{U}_1 = \mathbf{0} \quad \text{with } \mathbf{Z} = \mathbf{W} - k\mathbf{I}. \quad (6.16)$$

From (6.16) it can be seen that all the columns of  $\mathbf{U}_1$  are contained in the null space of  $\mathbf{Z}$ , so  $\text{im}(\mathbf{U}_1) \subseteq \text{ker}(\mathbf{Z})$ .

## Chapter 6. A Solution for Multi-Alignment by Transformation Synchronisation

---

However, it still remains to be shown that the dimensionality of  $\ker(\mathbf{Z})$  is exactly  $d$ , i.e.  $\text{im}(\mathbf{U}_1)$  spans the entire null space of  $\mathbf{Z}$  and not just a part of it. This is done by showing that there are no non-zero vectors  $\mathbf{x}$  that are not contained in  $\text{im}(\mathbf{U}_1)$  but are contained in  $\ker(\mathbf{Z})$ .

Formally this is expressed by the requirement that the set

$$A = \{\mathbf{x} \in \mathbb{R}^{kd} \mid \mathbf{x} \neq \mathbf{0}, \mathbf{x} \notin \text{im}(\mathbf{U}_1), \mathbf{x} \in \ker(\mathbf{Z})\}$$

is empty. Suppose now that  $A$  is not empty, so it contains the element  $\mathbf{x} \in \mathbb{R}^{kd}$ . Using the orthogonal decomposition theorem, the vector  $\mathbf{x}$  can be rewritten as  $\mathbf{x} = \mathbf{x}_{\ker} + \mathbf{x}_{\text{im}}$ , where  $\mathbf{x}_{\ker} \in \ker(\mathbf{U}_1^T)$  and  $\mathbf{x}_{\text{im}} \in \text{im}(\mathbf{U}_1)$ . The definition of  $A$  states that  $\mathbf{x} \notin \text{im}(\mathbf{U}_1)$ , which implies that  $\mathbf{x}_{\ker} \neq \mathbf{0}$ . Further, the definition of  $A$  states that

$$\mathbf{x} \in \ker(\mathbf{Z}) \tag{6.17}$$

$$\Leftrightarrow \mathbf{Z}\mathbf{x} = \mathbf{0} \tag{6.18}$$

$$\Leftrightarrow \mathbf{Z}(\mathbf{x}_{\ker} + \mathbf{x}_{\text{im}}) = \mathbf{0} \tag{6.19}$$

$$\Leftrightarrow \mathbf{Z}\mathbf{x}_{\ker} + \mathbf{Z}\mathbf{x}_{\text{im}} = \mathbf{0}. \tag{6.20}$$

Per definition  $\mathbf{x}_{\text{im}} \in \text{im}(\mathbf{U}_1) \subseteq \ker(\mathbf{Z})$ , so it follows that  $\mathbf{Z}\mathbf{x}_{\text{im}} = \mathbf{0}$ , leading to

$$\mathbf{Z}\mathbf{x}_{\ker} = \mathbf{0}. \tag{6.21}$$

Multiplication of  $\mathbf{x}_{\ker}^T$  on the left gives

$$\mathbf{x}_{\ker}^T \mathbf{Z}\mathbf{x}_{\ker} = 0 \tag{6.22}$$

$$\Leftrightarrow \mathbf{x}_{\ker}^T (\mathbf{U}_1 \mathbf{U}_2 - k\mathbf{I})\mathbf{x}_{\ker} = 0 \tag{6.23}$$

$$\Leftrightarrow \mathbf{x}_{\ker}^T \mathbf{U}_1 \mathbf{U}_2 \mathbf{x}_{\ker} - k\mathbf{x}_{\ker}^T \mathbf{x}_{\ker} = 0 \tag{6.24}$$

$$\Leftrightarrow \mathbf{x}_{\ker}^T \mathbf{U}_2^T \mathbf{U}_1^T \mathbf{x}_{\ker} - k\mathbf{x}_{\ker}^T \mathbf{x}_{\ker} = 0. \tag{6.25}$$

Per definition  $\mathbf{x}_{\ker} \in \ker(\mathbf{U}_1^T)$ , so  $\mathbf{U}_1^T \mathbf{x}_{\ker} = \mathbf{0}$ , leading to

$$\mathbf{x}_{\ker}^T \mathbf{x}_{\ker} = 0 \tag{6.26}$$

$$\Leftrightarrow \mathbf{x}_{\ker} = \mathbf{0}. \tag{6.27}$$

Equation (6.27) is a contradiction to  $\mathbf{x}_{\ker} \neq \mathbf{0}$ , thus, the set  $A$  is empty.  $\square$

Proposition 1 states that  $\mathbf{U}_1$  in (6.16) can, up to an invertible linear transformation, be retrieved by finding the  $d$ -dimensional null space of  $\mathbf{Z}$ . Let  $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  be the Singular Value Decomposition (SVD) of  $\mathbf{Z}$ . The  $d$  columns of  $\mathbf{V}$  corresponding to the zero singular values span  $\ker(\mathbf{Z})$  and give a solution to (6.16).

As we are only able to retrieve the transformations  $T_{i\star}$  in the blocks of  $\mathbf{U}_1$  up to invertible linear transformations, w.l.o.g. we create a new version of  $\mathbf{U}_1$ , call it  $\mathbf{U}'_1$ , with the first

$d \times d$  block being equal to the identity, as

$$\mathbf{U}'_1 = \mathbf{U}_1 T_{1\star}^{-1} = \begin{bmatrix} T_{1\star} T_{1\star}^{-1} \\ T_{2\star} T_{1\star}^{-1} \\ \vdots \\ T_{k\star} T_{1\star}^{-1} \end{bmatrix}. \quad (6.28)$$

### 6.2.3 Noisy Pairwise Transformations

Up until this point, the matrix  $\mathbf{U}_1$  is obtained under perfect information, i.e. the transitivity condition in Definition 3 holds for all  $T_{ij}$  transformations contained in the blocks of  $\mathbf{W}$ . However, we are interested in the case when the transitivity condition does not hold due to measurement noise. Assume now that we have a noisy observation of  $\mathbf{W}$ , denoted as  $\tilde{\mathbf{W}}$ . Also, let the noisy version of  $\mathbf{Z}$  be  $\tilde{\mathbf{Z}} = \tilde{\mathbf{W}} - k\mathbf{I}$ . Now, in general it is not the case that the null space of  $\tilde{\mathbf{Z}}$  is  $d$ -dimensional. Instead, the least-squares approximation of the  $d$ -dimensional null space is considered, which leads to the following optimisation problem:

**Problem 1.** *Least-squares Transformation Synchronisation*

$$\begin{aligned} \underset{\hat{T}_{1\star}, \dots, \hat{T}_{k\star}}{\text{minimise}} \quad & \|\tilde{\mathbf{Z}} \hat{\mathbf{U}}_1\|_F^2 \\ \text{subject to} \quad & \mathbf{u}_i^T \mathbf{u}_j = 0 \quad \text{for all } i \neq j \\ & \|\mathbf{u}_i\| = 1 \quad \text{for all } i \\ & \hat{\mathbf{U}}_1 = [\mathbf{u}_1, \dots, \mathbf{u}_d] \in \mathbb{R}^{kd \times d}. \end{aligned}$$

The rank- $d$  approximation of the null space of  $\tilde{\mathbf{Z}}$  can be retrieved using the SVD of  $\tilde{\mathbf{Z}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ . In this case the columns of  $\mathbf{V}$  corresponding to the  $d$  smallest singular values span the rank- $d$  approximation of the null space of  $\tilde{\mathbf{Z}}$ , giving  $\hat{\mathbf{U}}_1$ , the estimate for  $\mathbf{U}_1$ . By using (6.28),  $\hat{\mathbf{U}}'_1$  can be retrieved from  $\hat{\mathbf{U}}_1$ .

### 6.2.4 Affine Transformations in Homogeneous Coordinates

In this section it is shown that the method is also applicable for invertible affine transformations, rather than invertible linear transformations. This is done by representing the  $d$ -dimensional affine transformations  $T_{ij}$  by using  $(d+1) \times (d+1)$  homogeneous matrices.

## Chapter 6. A Solution for Multi-Alignment by Transformation Synchronisation

---

Each affine transformation  $T_{ij}$  can be written as

$$T_{ij} = \begin{bmatrix} \mathbf{A}_{ij} & \mathbf{0} \\ \mathbf{t}_{ij} & 1 \end{bmatrix}, \quad (6.29)$$

where  $\mathbf{A}_{ij}$  is the (invertible) linear  $d \times d$  transformation matrix and  $\mathbf{t}_{ij}$  is the  $d$ -dimensional row vector representing the translation. The inverse of  $T_{ij}$  is given by

$$T_{ij}^{-1} = \begin{bmatrix} \mathbf{A}_{ij}^{-1} & \mathbf{0} \\ -\mathbf{t}_{ij}\mathbf{A}_{ij}^{-1} & 1 \end{bmatrix}. \quad (6.30)$$

Similar to the linear case described in (6.8), the matrix  $\mathbf{W}$  is constructed from all  $T_{ij}$ . It is assumed that the matrix  $\tilde{\mathbf{W}}$ , corresponding to the noisy observation of  $\mathbf{W}$ , contains blocks that are proper affine transformations, i.e. the last column of each block is  $\begin{bmatrix} \mathbf{0} & 1 \end{bmatrix}^T$ .

A simple way to ensure that the synchronised transformations are affine transformations in homogeneous coordinates is to add the row vector  $\mathbf{z} = \begin{bmatrix} z & z & \dots & z \end{bmatrix} \in \mathbb{R}^{k(d+1)}$ , with  $z = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{d+1}$ , to the matrix  $\tilde{\mathbf{Z}}$ . By adding the vector  $\mathbf{z}$  to  $\tilde{\mathbf{Z}}$ , the vector  $\mathbf{z}^T$  is removed from the null space of  $\tilde{\mathbf{Z}}$ . Using this approach, a solution is then found by solving Problem (1) with the updated  $\tilde{\mathbf{Z}}$ . Then the resulting  $\hat{\mathbf{U}}'_1$  gives an estimate of the first  $d$  columns of  $\hat{T}_{i\star}$  ( $i = 1, \dots, k$ ) and these are the columns we seek.

### 6.2.5 Similarity Transformations

Similarity transformations are transformations that allow for translations, isotropic scaling, rotations and reflections. To retrieve similarity transformations, the estimates of the synchronised affine transformations  $\hat{T}_{i\star}$  ( $i = 1, \dots, k$ ) are determined first. The translation component  $\hat{\mathbf{t}}_{i\star}$  of  $\hat{T}_{i\star}$  can directly be extracted from  $\hat{T}_{i\star}$  since it has the structure presented in (6.29). To obtain the scaling factor and the orthogonal transformation, the linear component  $\hat{\mathbf{A}}_{i\star}$  is factorised using SVD, resulting in  $\hat{\mathbf{A}}_{i\star} = \mathbf{U}_{i\star}\mathbf{\Sigma}_{i\star}\mathbf{V}_{i\star}^T$ . The orthogonal component  $\hat{\mathbf{Q}}_{i\star}$  is then given by

$$\hat{\mathbf{Q}}_{i\star} = \mathbf{U}_{i\star}\mathbf{V}_{i\star}^T, \quad (6.31)$$

and the isotropic scaling factor  $\hat{s}_{i\star}$  is given by

$$\hat{s}_{i\star} = \left( \prod_{j=1}^d |(\sigma_{i\star})_{jj}| \right)^{\frac{1}{d}}, \quad (6.32)$$

where  $(\sigma_{i\star})_{jj}$  is the  $j$ -th element on the diagonal of  $\mathbf{\Sigma}_{i\star}$ .

**Remark 1.** *It can be shown that retrieving the orthogonal component as presented in eq. (6.31) is the least-squares solution to the projection onto the set of orthogonal matrices. However, in eq. (6.32) the isotropic scaling factor is retrieved as the geometric mean of the individual axis-aligned scaling factors. The least-squares solution to the projection onto the set of similarity transformations is given by the arithmetic mean, i.e.  $\hat{s}_{i\star}^{lsq} = \frac{1}{d} \sum_{j=1}^d |(\sigma_{i\star})_{jj}|$ .*

### 6.2.6 Euclidean Transformations

Similarity transformations without isotropic scaling are called euclidean transformations. To obtain euclidean transformations, the similarity transformations are extracted and the scaling factors  $\hat{s}_{i\star}$  (for all  $i = 1, \dots, k$ ) are set to 1.

### 6.2.7 Rigid Transformations

Euclidean transformations without reflections are called rigid transformations. Rigid transformations can be obtained by ensuring that the determinant of the rotational component  $\hat{\mathbf{Q}}_{i\star}$  described in (6.31) equals 1. This can be achieved by setting

$$\hat{\mathbf{Q}}_{i\star} = \mathbf{U}_{i\star} \mathbf{D}_{i\star} \mathbf{V}_{i\star}^T, \text{ with} \quad (6.33)$$

$$\mathbf{D}_{i\star} = \text{diag}(1, \dots, 1, \det(\mathbf{V}_{i\star}^T \mathbf{U}_{i\star})). \quad (6.34)$$

## 6.3 Experiments

By generating ground truth data and adding Gaussian noise to it, we first compare the error of the synchronised transformations using our method to the error of the unsynchronised transformations. Furthermore, the transformation synchronisation method is applied for solving the Generalised Procrustes Problem with missing points and with wrong correspondence assignments.

### 6.3.1 Noisy Transformations

In this section it is described how the ground truth transformations are generated, how noisy versions thereof are generated and eventually results of the transformation synchronisation method are presented.

#### Ground Truth Transformations

For the analysis of the performance of our method we generate a set of random transformations  $\mathcal{T}_\star = \{T_{i\star}\}_{i=1}^k$ , that are used in turn to generate the transitively consistent set of pairwise transformations  $\mathcal{T} = \{T_{ij} = T_{i\star} T_{j\star}^{-1}\}_{i=1, j=1}^k$ , serving as ground truth for the evaluation. The generation of  $\mathcal{T}_\star$  is described in the following.

The dot-notation is used to illustrate that  $\dot{x}$  is a random variable with a particular probability distribution. For generating the set  $\mathcal{T}_\star$ , we assume that the point-clouds that lead to the transformations have some structural similarity, i.e. the transformations are not entirely random. In particular, the scaling factors, the translation components and

## Chapter 6. A Solution for Multi-Alignment by Transformation Synchronisation

---

the linear part of the transformation are restricted in the sense that they cannot be arbitrary. However, arbitrary orientations in  $d$ -dimensional space are allowed for.

The set  $\mathcal{T}_\star$  contains the elements  $T_{i_\star}$  ( $i = 1, \dots, k$ ), which are samples of

$$\dot{T} = \begin{bmatrix} \dot{s}\dot{\mathbf{Q}}\dot{\mathbf{N}} & \mathbf{0} \\ \dot{\mathbf{t}} & 1 \end{bmatrix}, \quad (6.35)$$

where  $\dot{s} \sim \mathcal{U}(0.5, 1.5)$  is a scaling factor and  $\dot{\mathbf{t}} \sim \mathcal{U}(-2.5, 2.5)^d$  is a translation, with  $\mathcal{U}(a, b)^d$  denoting the  $d$ -dimensional uniform distribution having the open interval  $(a, b)^d$  as support. Samples of the  $d \times d$  random rotation matrix  $\dot{\mathbf{Q}}$  are drawn by extracting the rotational component of a non-singular random matrix as described in (6.31). The  $d \times d$  random noise matrix  $\dot{\mathbf{N}}$  is given by  $\dot{\mathbf{N}} = \mathbf{I} + \dot{\epsilon}$ , where  $\dot{\epsilon} \sim \mathcal{N}(0, 0.1^2)^{d \times d}$  is a  $d \times d$  random matrix with each element having univariate normal distribution  $\mathcal{N}(0, 0.1^2)$ . The purpose of creating the noise in the way using the random matrix  $\dot{\mathbf{N}}$  is to restrict the linear component in the transformation and thus to avoid ill-conditionedness with very high probability.

Depending on the type of transformation that is evaluated, the parameters of  $\mathcal{T}_\star$  have different properties, which are summarised in Table 6.1.

	$\dot{\mathbf{Q}}$	$\dot{\mathbf{t}}$	$\dot{s}$	$\dot{\mathbf{N}}$
linear	$ \det  = 1$	$= \mathbf{0}$	$\sim \mathcal{U}(0.5, 1.5)$	$\sim \mathbf{I} + \epsilon$
affine	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$\sim \mathcal{U}(0.5, 1.5)$	$\sim \mathbf{I} + \epsilon$
similarity	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$\sim \mathcal{U}(0.5, 1.5)$	$= \mathbf{I}$
euclidean	$ \det  = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$= 1$	$= \mathbf{I}$
rigid	$\det = 1$	$\sim \mathcal{U}(0.5, 1.5)^d$	$= 1$	$= \mathbf{I}$

Table 6.1: Properties of components of random transformations for different types of transformations generated according to (6.35).

Once the ground truth set  $\mathcal{T}$  of transitively consistent transformations has been established, a noisy version thereof is synthetically created, as described in the next section.

The error  $e(\mathcal{T}^1, \mathcal{T}^2)$  between two sets of pairwise transformations  $\mathcal{T}^1 = \{T_{ij}^1\}_{i,j=1}^k$  and  $\mathcal{T}^2 = \{T_{ij}^2\}_{i,j=1}^k$  is defined as

$$e(\mathcal{T}^1, \mathcal{T}^2) = \frac{1}{k^2} \sum_{i,j=1}^k \|T_{ij}^1 - T_{ij}^2\|_F. \quad (6.36)$$

### Additive Gaussian Noise

The set of noisy pairwise transformations  $\tilde{\mathcal{T}}^{\mathcal{N}}$  is created by adding to each element of the matrix  $T_{ij}$  a sample from  $\mathcal{N}(0, \sigma^2)$ , which is conducted for all matrices  $T_{ij} \in \mathcal{T}$  with  $i \neq j$ . In the case of homogeneous transformation matrices  $T_{ij}$ , no noise is added to the

last column, which shall always be  $(\mathbf{0} \ 1)^T$ .

Results of the simulations are shown in Fig. 6.1. The first row of graphs show that for all types of transformations the error of synchronised transformations is smaller than the error of the unsynchronised transformations and that the slope of the error in the synchronised case is smaller than in the unsynchronised case. In the second row it can be seen that, even with a high amount of noise ( $\sigma = 0.5$ ), the error of the synchronised transformations decreases with an increasing number of objects  $k$ . As anticipated, with increasing  $k$  there is more information available, directly resulting in a lower error. The last row of graphs shows that increasing the dimensionality results in an increasing error; however, the error of the synchronised transformations increases slower than for the unsynchronised ones.

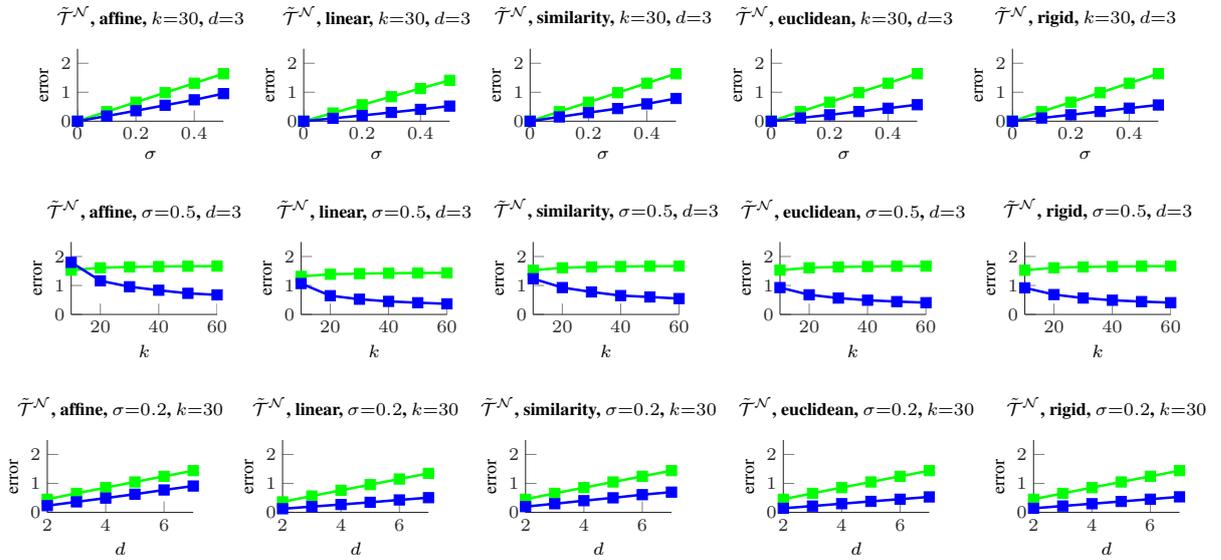


Figure 6.1: Error for additive normal noise  $\tilde{\mathcal{T}}^{\mathcal{N}}$  for different configurations as specified in the graph title with one varying parameter (horizontal axis). Each row of graphs shows a particular varying parameter ( $\sigma$ ,  $k$  and  $d$  from top to bottom) and each column of graphs shows a particular transformation type (affine, linear, similarity, euclidean, rigid, from left to right). The error as defined in (6.36) of the unsynchronised noisy transformations is shown in green and of the synchronised transformations in blue. Shown is the average error of 100 randomly generated sets of ground truth transformations, where for each ground truth transformation 20 runs of adding noise have been performed, resulting in a total of 2000 simulations per graph.

### 6.3.2 Generalised Procrustes Analysis

In addition to evaluating the synchronisation of noisy pairwise transformations we have applied our method for solving the Generalised Procrustes Problem (GPP), which is done on the one hand with missing data and on the other hand with wrong correspondence assignments. For both simulations the 2D fish shapes from the Chui-Rangarajan dataset (Chui and Rangarajan, 2003) with different levels of deformation and noise have been used (refer (Chui and Rangarajan, 2003) for more details). For each level of deformation and noise the dataset contains  $K = 100$  shapes, each comprising  $n = 98$  points in  $d = 2$  dimensions.

Finding the similarity transformation that best aligns two shapes, which is a subroutine for the evaluated reference-based, the iterative mean shape-based and the synchronisation-based method, is performed by an AOP implementation with symmetric scaling factors (Horn et al., 1988). In the reference-based solution of GPP one shape is randomly selected as reference and all other shapes are aligned with the reference. For the iterative mean shape-based method the initial reference shape is selected randomly and then the mean shape is iteratively updated. In the synchronisation-based solution of GPP all  $k^2$  pairwise AOPs are solved first, followed by the synchronisation of the resulting transformations in order to aggregate all information contained in the pairwise transformations. Additionally, the stratified GPA method proposed in (Bartoli et al., 2013) is evaluated for solving the GPP. In our experiments we have observed that by using the stratified GPA method the linear part of the resulting transformations may collapse to the zero matrix; in order to enable a comparison with the other methods in these cases the linear part of the transformation has simply been set to the identity matrix.

In the missing data experiments as well as the wrong correspondence experiments for each single run  $k = 30$  out of  $K = 100$  shapes are randomly selected. For the experiments in the missing points case the missing points are simulated by discarding points according to a given probability. For the experiments with wrong correspondences the correct correspondences are randomly disturbed in order to simulate wrong correspondences.

In contrast to *solving* the AOPs, in both experiments the computation of the error is performed using the original shape (i.e. with all points and with perfect correspondences). With that we investigate up to which amount recovering the original shapes from corrupt shape data is possible. The average shape error of a set of shapes  $\mathcal{X} = \{\mathbf{X}_i\}_{i=1}^k$  is defined as  $e(\mathcal{X}) = \frac{1}{k^2} \sum_{i,j=1}^k \|\mathbf{X}_i - \mathbf{X}_j\|_F$ .

### Missing Points

In every run, additionally to randomly selecting 30 out of 100 shapes, each data point of a shape is considered to be missing with probability  $\eta$ . As the implemented methods solve the AOP only for common points in each pair of shapes, values for  $\eta$  larger than 0.7 have not been investigated because with  $\eta > 0.7$  the cases that the number of common points in a pair of shapes is less than  $d = 2$  occur too frequently (for  $d$ -dimensional data, there must be at least  $d$  points in each shape in order to result in a system that is not under-determined). Also, for  $\eta \leq 0.7$  it is possible that the number of common points in a pair of shapes is less than  $d = 2$ ; in these cases the draw of missing data is simply repeated.

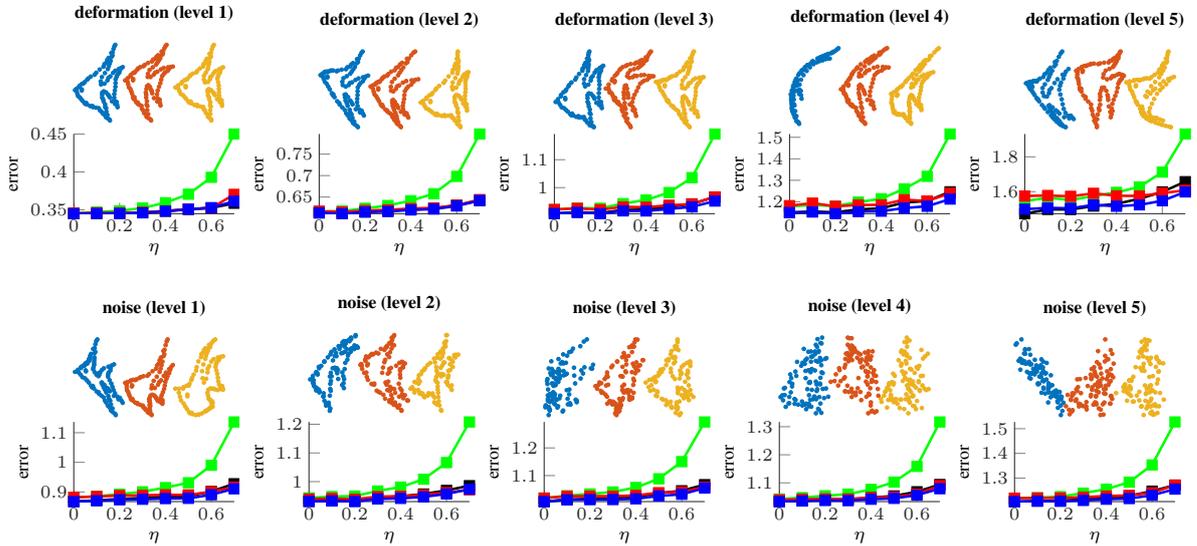


Figure 6.2: Average shape error for the reference-based (green), iterative mean shape-based (black), synchronisation-based (blue) and stratified (red) method for solving the GPP with missing data. The horizontal axis shows the probability  $\eta$  that a point is considered missing. At the top of each graph three shapes according to the particular level of deformation or noise are depicted. Shown is the average shape error for 500 draws of missing data in each graph. In every run  $k = 30$  out of  $K = 100$  shapes are randomly selected, where each shape comprises  $n = 98$  points in  $d = 2$  dimensions.

In Fig. 6.2 the resulting error of the reference-based, the iterative mean shape-based, the synchronisation-based and the stratified solution of the GPP with missing data are shown for different levels of deformation and noise. It can be seen that even with an increasing amount of missing data, when using the synchronisation-based method the error increases only slightly, whilst the error of the reference-based method increases significantly with a larger amount of missing points. With respect to the error, the transformation synchronisation method performs only marginally better than the iterative mean shape-based method and the stratified method. However, the average runtimes

for solving a single GPP instance was 0.007 s for the reference-based method, 0.162 s for the synchronisation-based method, 1.932 s for the iterative mean shape-based method and 2.265 s for the stratified method, illustrating that our method performs significantly better than all other methods when taking runtime and error into account at the same time.

### Wrong Correspondence Assignments

Additionally to the case of missing points, we have applied our method to solve the GPP with wrong correspondence assignments between shapes. In order to mimic practical applications, where it is frequently the case that the true correspondences are unknown and thus it must be assumed that wrong correspondences are present, we do not make any efforts to correct these wrong correspondences (such as using RANSAC (Fischler and Bolles, 1981) or permutation synchronisation (Pachauri et al., 2013)). Instead, for each pair of shapes the AOP is solved whilst being aware that some of the points in the one shape have wrong counterparts in the other shape. Of course this will have influence on the resulting transformations. Thus, the objective of the simulations described in this section is to assess to what extent the transformations from shapes with wrong correspondences can be reconstructed using transformation synchronisation.

In every run, additionally to randomly selecting 30 out of 100 shapes, the correspondences between the  $n$  points in each shape are disturbed. For disturbing the correspondence assignments each pair of shapes that is to be aligned is considered independently. For that, a proportion of  $\nu \in [0, 1]$  points from the total number of  $n$  points is selected. Then, as correspondences between the pair of point-clouds  $\mathbf{X}_i, \mathbf{X}_j \in \mathbb{R}^{n \times 2}$  are implicitly given by the ordering of the rows, the rows corresponding to the previously selected points are reordered randomly in one of the point-clouds, directly resulting in disturbed correspondence assignments between the pair of point-clouds  $\mathbf{X}_i, \mathbf{X}_j$ .

In Fig. 6.3 the reference-based, the iterative mean shape-based, the synchronisation-based and the stratified solution of the GPP with wrong correspondences are shown for different levels of deformation and noise. On the right of Fig. 6.3 examples of the correspondences between pairs of shapes are depicted for different values of  $\nu$ .

It can be seen that for different levels of deformation and different levels of noise with 70% – 80% of wrong correspondences the outcome is only marginally affected when using our proposed method. In contrast, all other evaluated methods result in significantly larger errors, which can be explained by the fact that our method is the only one that is able to make use of the information that is contained in all pairwise transformations.

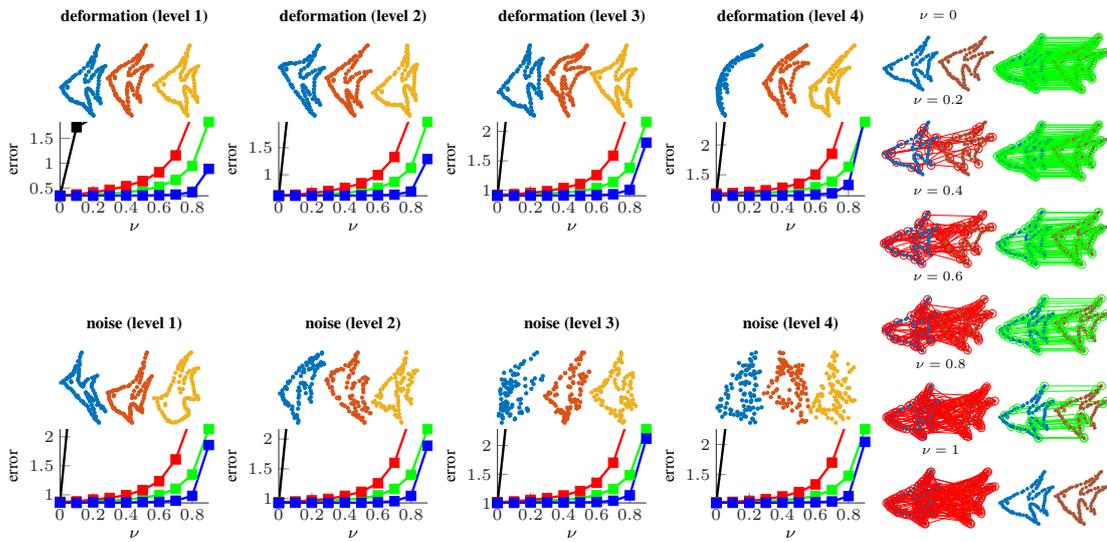


Figure 6.3: Average shape error for the reference-based (green), iterative mean shape-based (black), synchronisation-based (blue) and stratified (red) method for solving the GPP with wrong correspondences. The horizontal axis shows the proportion  $\nu$  of wrong correspondences. At the top of each graph three shapes according to the particular level of deformation or noise are depicted. Shown is the average shape error for 500 runs of disturbing correspondence assignments in each graph. In every run  $k = 30$  out of  $K = 100$  shapes are randomly selected, where each shape comprises  $n = 98$  points in  $d = 2$  dimensions. In the right-most column examples of the correspondence assignments between a pair of shapes are depicted for different values of  $\nu$  in each row. In order to keep the visualisation as coherent as possible, the wrong correspondences (red lines) and the correct correspondences (green lines) are shown separately.

## **6.4 Conclusion**

The alignment of multiple (corresponding) point-clouds simultaneously is generally tackled by iteratively aligning all point-clouds to some reference. Whereas this approach is biased (selecting a fixed reference) or initialisation-dependent (using the adaptive mean as reference) we have presented a method that is completely unbiased and does not depend on initialisation.

Our key observation is that the underlying noise-free transformations can be retrieved from the null space of a matrix that can directly be obtained from pairwise alignments. Whilst related approaches for rotation matrices (Hadani and Singer, 2011a,b; Singer and Shkolnisky, 2011) or permutation matrices (Pachauri et al., 2013) have been proposed, we have generalised the synchronisation method to handle general linear and affine transformations as well as similarity, euclidean and rigid transformations. Experimentally we were able to demonstrate that the proposed method is able to effectively reduce noise from the set of pairwise transformations and to solve the Generalised Procrustes Problem at least as good as existing approaches for the missing data case whilst significantly outperforming other methods for the presented wrong correspondence case.

## **Acknowledgements**

Supported by the Fonds National de la Recherche, Luxembourg (5748689, 6538106, 8864515).

# 7 Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation

## Abstract

Transitive consistency of pairwise transformations is a desirable property of groupwise image registration procedures. The transformation synchronisation method (Bernard et al., 2015b) is able to retrieve transitively consistent pairwise transformations from pairwise transformations that are initially not transitively consistent. In the present paper, we present a numerically stable implementation of the transformation synchronisation method for affine transformations, which can deal with very large translations, such as those occurring in medical images where the coordinate origins may be far away from each other. By using this method in conjunction with any pairwise (affine) image registration algorithm, a transitively consistent and unbiased groupwise image registration can be achieved. Experiments involving the average template generation from 3D brain images demonstrate that the method is more robust with respect to outliers and achieves higher registration accuracy compared to reference-based registration.

## 7.1 Introduction

Image registration has attracted a lot of attention in the medical imaging community mainly due to the vast amount of applications both in research and in the clinic. Applications include surgery planning, multi-modal diagnosis, statistical analyses, normalisation, computational anatomy, longitudinal studies, image segmentation, and many more. The registration (or alignment) of a *moving image* with a *fixed image* can be defined as finding a spatial mapping that transforms the moving image such that it fits the fixed image

## Chapter 7. Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation

---

“best”. To measure the agreement of two images, mostly intensity-based metrics are used as surrogates for the unknown true correspondence. The mapping is in general obtained by optimisation methods, where it is necessary to select an application-dependent transformation model. Commonly, the transformation models are categorised by affine/linear and deformable transformations. Affine transformation models have a low number of degrees-of-freedom that allow only for a coarse alignment and thus are insensitive to overfitting. On the contrary, deformable transformation models are capable of aligning images on a very fine scale; however, controlling the trade-off between overfitting and realistic deformation is difficult. Frequently, affine transformations serve as initialisation for deformable transformations to avoid overfitting in the early stage of the optimisation.

The simultaneous registration of multiple images is more difficult. Different approaches to tackle this problem are: aligning each image individually to a *fixed reference* (e.g. chosen as one of the images); aligning each image with an iteratively *evolving reference image* (Joshi et al., 2004; Reuter et al., 2012); finding a *path of pairwise transformations* containing all images (Škrinjar et al., 2008); *image congealing*, where the variability along the known axes of variation is removed iteratively (Learned-Miller, 2006; Zöllei et al., 2005), and the related *accumulated pair-wise estimates (APE)* approach (Wachinger and Navab, 2013); considering a *minimum description length (MDL)* approach of a Statistical Shape Model built from the correspondences given due to the groupwise image registration (Cootes et al., 2004); or, using a Bayesian formulation for dense template estimation based on *Expectation Maximisation (EM)* (Allasonnière et al., 2007). In the work by Vercauteren et al. (2006), for video mosaicing with motion distortion correction, global alignment is seen as estimation problem on a Lie group.

However, choosing a fixed reference image or a path of sequential transformations induces a bias, and the iterative methods are generally local methods that are initialisation-dependent. In any case this may result in suboptimal alignments. One way of measuring the degree of suboptimality is to use a transitive consistency criterion, which is based on the fact that for a perfect alignment the composite transformation from A to B to C must be identical to the direct transformation from A to C. Registration methods improving this transitive consistency have been proposed for deformable transformations (Gass et al., 2014; Geng, 2007).

Based on recent works on transformation synchronisation (Bernard et al., 2015b), in this paper an unbiased, truly reference-free and transitively consistent approach for aligning multiple images under the affine transformation model is presented. Our approach is useful in applications where the affine alignment of multiple images is required. This includes for example the creation of an affine average template, the creation of an (affine) probabilistic atlas, or the initialisation for deformable multi-image alignment. Furthermore, since our approach is unbiased, the resulting average template is attractive for statistical

analyses. By exploiting redundancies between pairwise transformations, a more global approach of multi-image registration is achieved with the proposed method. Compared to reference-based groupwise registration, this leads to higher registration accuracy.

## 7.2 Methods

In this section, the transformation synchronisation method is briefly recapitulated and a numerically stable implementation thereof is presented that can handle large translations. Subsequently it is described how the method can be applied to multi-image registration.

### 7.2.1 Overview of Transformation Synchronisation

Transformation synchronisation is a method for reconstructing transitively consistent transformations from the set of (disturbed) pairwise transformations (Bernard et al., 2015b), which is briefly summarised in the following.

At first, the case of perfect (undisturbed) data is described. Given is the set  $\mathcal{T} = \{T_{ij}\}_{i,j=1}^k$  containing  $k^2$  pairwise transformations, where  $T_{ij} \in \mathbb{R}^{4 \times 4}$  denotes an invertible affine 3D transformation matrix from image  $i$  to image  $j$  represented in homogeneous coordinates. The set  $\mathcal{T}$  is said to be transitively consistent if  $T_{ij}T_{jl} = T_{il}$  holds for all  $i, j, l = 1, \dots, k$ .

Assuming transitive consistency, every transformation matrix  $T_{ij}$  can be represented as  $T_{i\star}T_{\star j}$ , where  $\star$  denotes *some* reference coordinate system that is fixed for all  $i, j = 1, \dots, k$  (Bernard et al., 2015b). For the case of undisturbed data, for any  $l$  it also holds that  $T_{\star l} = T_{l\star}^{-1}$ . Arranging all  $k^2$  transformations into a  $4k \times 4k$  matrix  $W$  gives

$$W = \begin{bmatrix} T_{11} & \cdots & T_{1k} \\ \vdots & \ddots & \vdots \\ T_{k1} & \cdots & T_{kk} \end{bmatrix} = [T_{ij}] = [T_{i\star}T_{\star j}] = [T_{i\star}T_{j\star}^{-1}] = U_1U_2, \quad (7.1)$$

with  $U_1 = [T_{1\star}^T \ T_{2\star}^T \ \dots \ T_{k\star}^T]^T$  and  $U_2 = [T_{1\star}^{-1}, T_{2\star}^{-1}, \dots, T_{k\star}^{-1}]$ . This shows that for transitively consistent transformations the matrix  $W$  can be factorised into  $U_1U_2$ .

Multiplying (7.1) with  $U_1$  from the right leads to  $WU_1 = U_1U_2U_1$ . Considering now that  $U_2U_1 = k\mathbf{I}_4$ , where  $\mathbf{I}_n$  denotes the  $n \times n$  identity matrix, this results in

$$WU_1 = kU_1 \quad \Leftrightarrow \quad ZU_1 = \mathbf{0}_{4k \times 4} \quad \text{with } Z = W - k\mathbf{I}_{4k}. \quad (7.2)$$

The latter shows that  $U_1$  can, up to an invertible linear transformation multiplied on the right, be obtained by finding the 4-dimensional nullspace of  $Z$ .

Now, if the pairwise transformations are obtained by independent measurements (e.g. pairwise image registrations), in general it does not hold that the transformations are

transitively consistent or inverse consistent. Therefore, in general  $Z$  does not have a 4-dimensional nullspace. Instead, we consider the least-squares approximation of the 4-dimensional nullspace of  $Z$ , which can be obtained by Singular Value Decomposition (SVD) of  $Z$ . Let  $\overline{W} = \begin{bmatrix} \overline{T_{ij}} \end{bmatrix} = U_1 U_2$  be the so-obtained synchronised version from a noisy  $W = \begin{bmatrix} T_{ij} \end{bmatrix}$ .

### 7.2.2 Numerical Stability for Large Translations

Large translations in the pairwise transformations may lead to an ill-posed matrix  $W$ , constituting a problem with respect to the numerical stability of the synchronisation method.

Each affine transformation block  $T_{ij}$  in  $W$  has the form

$$T_{ij} = \begin{bmatrix} A_{ij} & \mathbf{0}_{3 \times 1} \\ t_{ij} & 1 \end{bmatrix} \text{ with } A_{ij} \in \mathbb{R}^{3 \times 3} \text{ and } t_{ij} \in \mathbb{R}^{1 \times 3}. \quad (7.3)$$

Generally, in medical image registration it can be assumed that the anatomical entities depicted in the images are similar to a certain extent as they represent objects of the same (fixed) class (e.g. brains). This imposes certain properties onto the linear part  $A_{ij}$  of the transformation: the rotational part of  $A_{ij}$  can be arbitrary, because in principal the orientation of the patient can be arbitrary. In contrast, the scaling or shearing are not arbitrary (for example it is unlikely that the scale between two individual adult brains differs by a factor of 10 or even more). Both properties imply that the values of the elements in  $A_{ij}$  are bounded, i.e. they are in (or close to) the interval  $[-1, 1]$ . In contrast, the origin of the coordinate frame of each image can, in principal, be arbitrary. With that, the values of the elements in the translation component  $t_{ij}$  do not have such a bound, i.e. they can be orders of magnitude larger than the elements of the linear part, which impairs numerical stability.

Instead of directly finding the 4-dimensional least-squares approximation of the nullspace of  $Z = W - k\mathbf{I}_{4k}$ , a numerically more stable approach is now described. For affine transformations  $T_{ij}$  the matrix  $Z$  is reducible, i.e. for some permutation matrix  $P$ , the matrix  $Z' = P^T Z P$  is a block upper triangular matrix of the form

$$Z' = P^T Z P = \begin{bmatrix} Z_{11} & Z_{12} \\ \mathbf{0}_{3k \times k} & Z_{22} \end{bmatrix}, \quad (7.4)$$

where the lower left block  $\mathbf{0}_{3k \times k}$  contains the zeros from the homogeneous transformation part,  $Z_{11} \in \{1, 1-k\}^{k \times k}$  contains the constant elements from the homogeneous transformation part,  $Z_{22} \in \mathbb{R}^{3k \times 3k}$  contains the linear transformation part, and  $Z_{12} \in \mathbb{R}^{k \times 3k}$  contains the (possibly large) translation components. Using the Dulmage-Mendelsohn decomposition (Dulmage and Mendelsohn, 1958), the permutation  $P$  transforming  $Z$  to  $Z'$

in (7.4) is determined. In the following it is described how  $U'_1$ , spanning the 4-dimensional nullspace of  $Z'$ , can be obtained. From  $U'_1$ , the matrix  $U_1$  can directly be obtained by  $U_1 = PU'_1$ .

Writing  $U'_1 \in \mathbb{R}^{4k \times 4}$  as  $U'_1 = \begin{bmatrix} V_1^T & V_2^T \end{bmatrix}^T$ , where  $V_1 \in \mathbb{R}^{k \times 4}$  and  $V_2 \in \mathbb{R}^{3k \times 4}$ , allows to rewrite eq. (7.2) for  $Z'$  as

$$Z'U'_1 = \mathbf{0}_{4k \times 4} \quad \Leftrightarrow \quad \begin{bmatrix} Z_{11} & Z_{12} \\ \mathbf{0}_{3k \times k} & Z_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \mathbf{0}_{4k \times 4}. \quad (7.5)$$

Eq. (7.5) is solved first for the matrix  $V_2$  by finding the 4-dimensional nullspace of  $Z_{22}$  using Singular Value Decomposition, which is stable since  $Z_{22}$  contains the (well-posed) linear transformation parts only. Once  $V_2$  is known, the remaining part  $V_1$  of  $U'_1$  is determined by solving the linear system of equations  $Z_{11}V_1 = -Z_{12}V_2$  for  $V_1$ . Obtaining  $U_1$  from  $U'_1$ ,  $U_2$  from  $U_1$ , and reconstructing  $\overline{W} = U_1U_2$  eventually gives the synchronised version  $\overline{W}$  of  $W$ .

### 7.2.3 Multi-image Alignment

The processing pipeline of aligning multiple images is illustrated in Fig. 7.1.

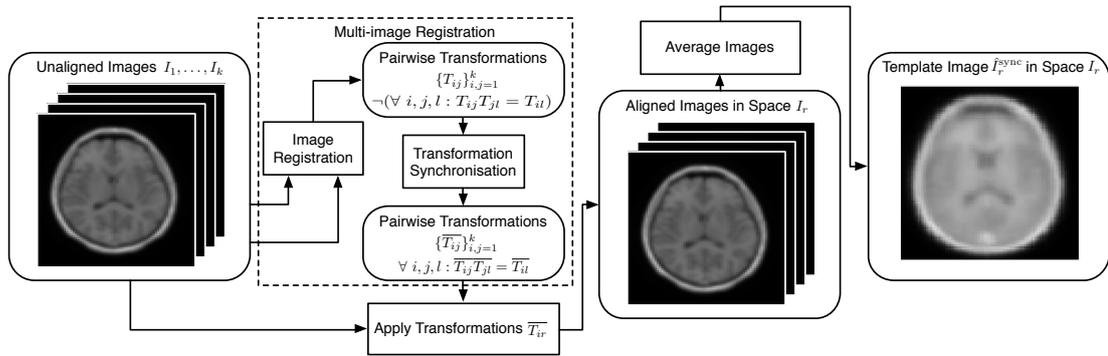


Figure 7.1: Processing pipeline for aligning multiple images using transformation synchronisation. Rectangles represent *methods* and rounded rectangles represent *data*.

First the set of transformations  $\{T_{ij}\}_{i,j=1}^k$  between all pairs of unaligned images  $I_1, \dots, I_k$  is determined using any affine image registration algorithm. Then, all transformations are synchronised using the numerically stable synchronisation method, resulting in the set  $\{\overline{T}_{ij}\}_{i,j=1}^k$  of transitively consistent pairwise transformations.

In order to represent all images in the *same* coordinate system it is necessary to select such a common coordinate system. Due to the transitive consistency of the set of all pairwise transformations, in theory it is irrelevant what is chosen as common coordinate

## Chapter 7. Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation

---

system. To be more specific, up to discretisation and interpolation, the resulting average image is the same for *any* choice of coordinate system, thus, we consider our approach to be *unbiased*. Interpolation bias can for example be handled by a mid-space based normalisation (Reuter et al., 2012).

In order to be able to directly compare our method to the reference-based approach, we create average images with the common coordinate system, image dimension and voxel size of image  $I_r$  for all  $r = 1, \dots, k$ . So, for a given image  $I_r$ , each image  $I_i$  (for  $i = 1, \dots, k$ ) is transformed to the space of the image  $I_r$  by applying the transformation  $T_{ir}$ . We emphasize that  $I_r$  should not be confused with the reference image in reference-based multi-image registration (cf. preceding paragraph). Eventually, the average image  $\hat{I}_r^{\text{sync}}$  of all  $k$  images (represented in the space of  $I_r$ , thus having the same image dimensions) is computed.

The use of transformation synchronisation constitutes the major difference to reference-based template creation approaches, since this method is able to aggregate *all* information contained in the set of pairwise transformations, in contrast to reference-based methods, which only incorporate the information contained in  $k$  pairwise transformations.

### 7.3 Experiments

In this section the results of applying the proposed framework for the unbiased template construction from 17 T1-weighted MR images is described. The dimensions of the images are between  $256 \times 256 \times 122$  and  $512 \times 512 \times 168$  voxels, where the voxel sizes vary from  $0.5 \times 0.5 \times 1$  to approximately  $0.9 \times 0.9 \times 1.4$  (in  $\text{mm}^3$ ).

Two different affine registration methods have been used to find the set of pairwise transformations (FLIRT (Jenkinson and Smith, 2001) and ANTS (Avants et al., 2011)). To run the FLIRT algorithm, we set parameters as follows: normalised mutual information, 12 degrees-of-freedom and search angle  $[-20, 20]$  for the rotation in all directions. Cross-correlation and the affine transformation model were used in ANTS. Other parameters remained unchanged.

The reference-based alignment of images is used as baseline: (i) selection of reference image  $I_r$ ; (ii) registration of all other images with  $I_r$ ; (iii) transformation of all images to space of  $I_r$ ; and (iv) computation of average template  $\hat{I}_r^{\text{ref}}$ . In order to perform a fair analysis the evaluation has been carried out for all  $r = 1, \dots, k$  images  $I_r$  as reference.

Furthermore, to enable a direct comparison between the reference-based and synchronisation-based methods, for the latter a total of  $k$  average template images  $\hat{I}_1^{\text{sync}}, \dots, \hat{I}_k^{\text{sync}}$  are created. The results of the synchronisation-based method are represented in the image

spaces  $I_r$  for  $r = 1, \dots, k$  only for the sake of comparability with the reference-based method. By representing image  $I_i$  in the space of image  $I_r$  we mean applying the respective transformations, i.e.  $T_{ir}$  for the reference-based method and  $\overline{T_{ir}}$  for the synchronisation-based method.

The average transitivity error  $e_{\text{trans}}(\mathcal{T})$  of a set of transformations  $\mathcal{T} = \{T_{ij}\}_{i,j=1}^k$ , which measures the degree of transitive consistency of the set of pairwise transformations  $\mathcal{T}$ , is defined as

$$e_{\text{trans}}(\mathcal{T}) = \frac{1}{k^3} \sum_{i,j,l=1}^k \|T_{ij}T_{jl} - T_{il}\|_F, \quad (7.6)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm.

Let  $ncc(I_i, I_j)$  be the normalised cross-correlation (NCC) between the images  $I_i$  and  $I_j$ . In each experiment the NCC is computed  $k^2$  times, i.e. for all  $r = 1, \dots, k$  and  $i = 1, \dots, k$  the NCC is evaluated. For the reference-based method the NCC is computed as  $c_{ir}^{\text{ref}} = ncc(I_i, \hat{I}_r^{\text{ref}})$ , and for the synchronisation-based method as  $c_{ir}^{\text{sync}} = ncc(I_i, \hat{I}_r^{\text{sync}})$ , where image  $I_i$  is in both cases represented in the space of  $I_r$ .

Additionally to the average transitivity error and NCC, a landmark-based evaluation criterion has been used. Based on a total of 8 bilateral brain structure segmentations of *Substantia Nigra & Subthalamic Nucleus* (as compound object), *Nucleus Ruber, Putamen & Globus pallidus* (as compound object), and *Thalamus* (Bernard et al., 2014), the centre of gravity (COG) of each segmented object is determined. Let  $e_{o,i,j,r}$  be the *cog-error* that is defined as the magnitude of the error vector between the COG of object  $o$  in image  $I_i$  and the COG of object  $o$  in image  $I_j$ , where  $I_i$  and  $I_j$  are both represented in the space of image  $I_r$ . In each experiment the cog-error is computed  $8k^3$  times for all  $r = 1, \dots, k$ ;  $i = 1, \dots, k$ ;  $j = 1, \dots, k$  and for all eight objects  $o = 1, \dots, 8$ .

Results for the measures described above are summarised in Table 7.1.

	$e_{\text{trans}}^{\text{ref}}$	$e_{\text{trans}}^{\text{sync}}$	$c^{\text{ref}}$	$c^{\text{sync}}$	$e^{\text{ref}}$	$e^{\text{sync}}$
F	9.142	<b>0</b>	$0.859 \pm 0.025$	<b><math>0.863 \pm 0.017</math></b>	$5.042 \pm 3.672$	<b><math>4.362 \pm 2.439</math></b>
A	12.004	<b>0</b>	$0.861 \pm 0.022$	<b><math>0.867 \pm 0.017</math></b>	$5.157 \pm 4.062$	<b><math>4.394 \pm 2.483</math></b>

Table 7.1: Comparison of reference-based method and synchronisation-based method for two experiments (F=FLIRT, A=ANTS). The scores shown are the average transitivity errors  $e_{\text{trans}}^{\{\text{ref},\text{sync}\}}$  (high values indicate problems), the mean and the standard deviation of the NCC  $c^{\{\text{ref},\text{sync}\}}$  (higher mean is better), and the mean and the standard deviation of the cog-error  $e^{\{\text{ref},\text{sync}\}}$  in mm (lower is better).

For both experiments the distributions of  $c_{ir}^{\{\text{ref},\text{osync},\text{sync}\}}$  and  $e_{o,i,j,r}^{\{\text{ref},\text{osync},\text{sync}\}}$  are presented as boxplots in Figs. 7.2 and 7.4, where the superscripts *osync* and *sync* are used to

## Chapter 7. Transitively Consistent and Unbiased Multi-Image Registration Using Numerically Stable Transformation Synchronisation

distinguish the original transformation synchronisation method and our proposed transformation synchronisation method with the numerical stability improvement.

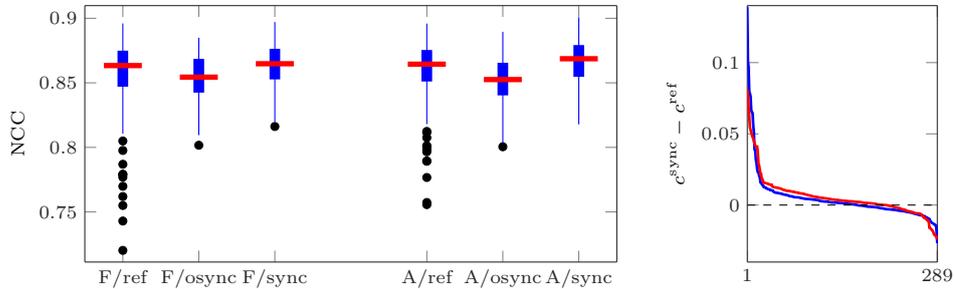


Figure 7.2: Left: distribution of NCC  $c_{ir}$  for  $i = 1, \dots, k$  and  $r = 1, \dots, k$  for the reference-based method and the synchronisation-based methods (with and without the numerical stability improvement) visualised as boxplots (median as red horizontal line, 25th and 75th percentile as blue box, extent of extreme points that are not considered outliers as blue vertical line and outliers as black dots). Right: sorted NCC differences  $c_{ir}^{sync} - c_{ir}^{ref}$  for  $i = 1, \dots, k$  and  $r = 1, \dots, k$  (FLIRT in blue, ANTS in red).

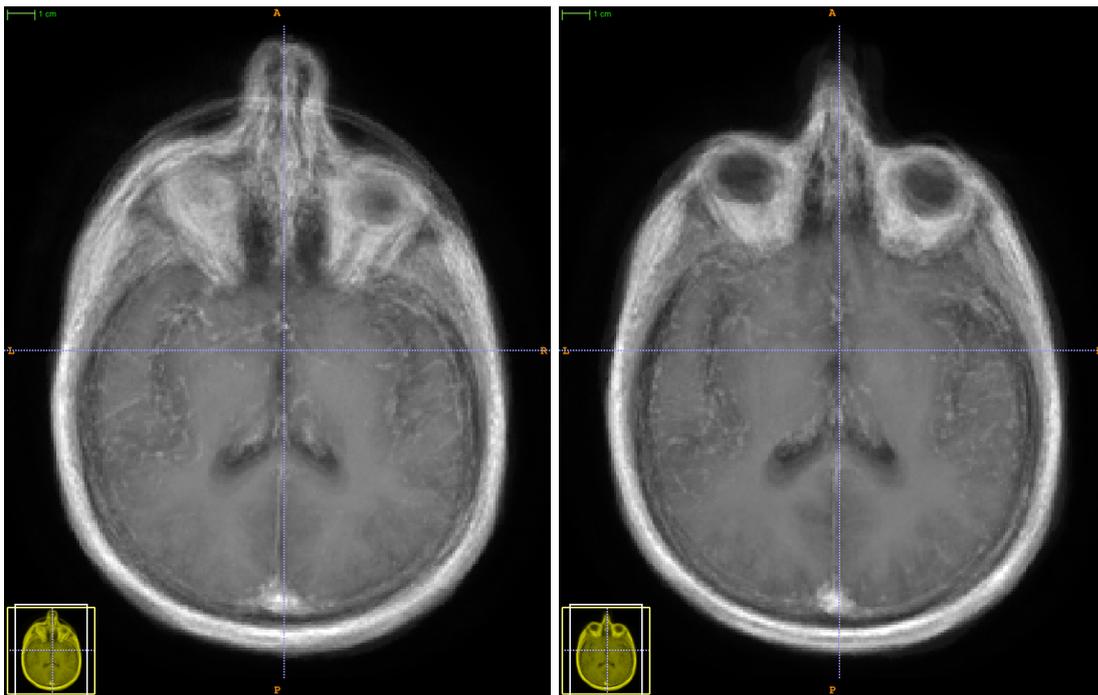


Figure 7.3: Axial slices of brain average templates created using the reference-based method (left) and the proposed synchronisation-based approach (right). Note the “ghost effect” in the area of the nose for the reference-based method stemming from a bad registration.

Fig. 7.2 reveals that with respect to NCC, on average the proposed approach outperforms the reference-based method, whereas the original transformation synchronisation method

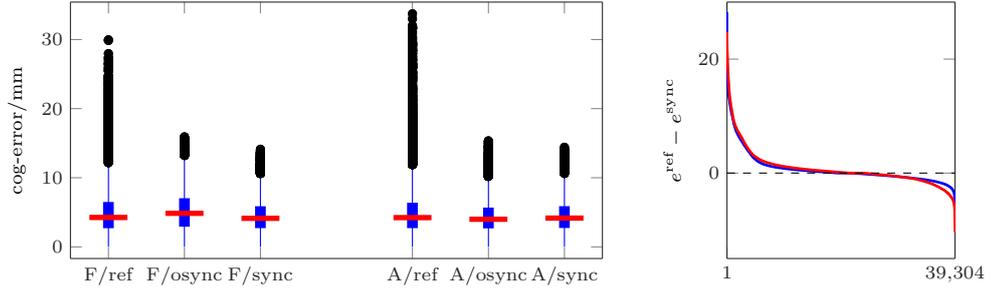


Figure 7.4: Left: distribution of the cog-error  $e_{o,i,j,r}$  for  $i = 1, \dots, k$ ;  $j = 1, \dots, k$ ;  $r = 1, \dots, k$  and  $o = 1, \dots, 8$  for the reference-based method and the synchronisation-based methods (with and without the numerical stability improvement) visualised as boxplots (median as red horizontal line, 25th and 75th percentile as blue box, extent of extreme points that are not considered outliers as blue vertical line and outliers as black dots). Right: sorted cog-error differences  $e_{o,i,j,r}^{ref} - e_{o,i,j,r}^{sync}$  for  $i = 1, \dots, k$ ;  $j = 1, \dots, k$ ;  $r = 1, \dots, k$  and  $o = 1, \dots, 8$  (FLIRT in blue, ANTS in red).

without the numerical stability improvement delivers on average lower scores than both the reference-based method and our proposed method. Furthermore, the boxplots show that using the reference-based method may result in outliers that have considerably lower NCC scores. Fig. 7.3 (left) shows such a case when using the reference-based method for brain average template creation, where there are undesired artefacts in the area of the nose that are not present using the proposed synchronisation method (right).

Considering transitive consistency, which is frequently employed for evaluating the quality of registration methods (Datteri and Dawant, 2012; Gass et al., 2014; Holden et al., 2000), the proposed approach significantly outperforms the reference-based method since our method always results in perfect transitive consistency. Whilst a low average transitivity error does not necessarily imply a good registration (e.g. consider the trivial case of not performing a registration at all, i.e. setting all transformations to identity), a large average transitivity error indicates that there is a problem with the registration.

The cog-error, shown as boxplots in Fig. 7.4, reveal some extreme outliers of up to around 3.5 cm when using the reference-based method. For the synchronisation-based methods the cog-error is always below 2 cm, where the proposed synchronisation method with the numerical stability improvement has a slightly lower median when using FLIRT for finding the pairwise transformations than the original synchronisation method. Note that in general the cog-errors have comparably large values since due to the regularity of the affine transformation model only a coarse alignment of structures is possible.

### 7.4 Conclusion

A solution for the groupwise affine registration of images based on transformation synchronisation has been presented. It has been shown that the method is more robust with respect to outliers than the common reference-based method and that it leads to improved registration accuracy. Our proposed approach can be seen as an averaging procedure that removes noise (accounting for transitive inconsistency) from pairwise transformations.

A shortcoming is that a quadratic number of pairwise transformations needs to be available, which may, depending on the application, be unaffordable. However, because an average template is usually created only once, in many cases the computational overhead may be acceptable, since the extra-effort is compensated by a more reliable result. An alternative approach is to find a trade-off between complete unbiasedness and computational cost by synchronising only a subset of all pairwise transformations using transformation synchronisation for partial data (Thunberg et al., 2015).

On its own, the proposed affine groupwise image registration procedure is valuable for the construction of affine average templates. Moreover, it can be used as initialisation for groupwise deformable image registration, where a good initial solution is essential due to the highly non-convex nature of deformable registration methods. In the work by Jia et al. (2010) a hierarchical groupwise registration scheme is presented that already assumes a groupwise affine registration between all images. Also, in the work by Seghers et al. (2004), for brain template generation the affine registration of a set of images is performed by registering all images to an existing average template. In both approaches our method can be used for the groupwise affine registration.

Due to the unbiasedness of the method it is ideal for statistical analyses. Our experiments suggest that the reference-based method performs significantly worse for certain choices of reference images, rendering the choice of reference crucial. Since a priori it is unknown which of the images is a good reference and which is not, a reference-free method is favourable in many scenarios.

### Acknowledgements

This work is supported by the Fonds National de la Recherche, Luxembourg (5748689, 6538106, 8864515, 9169303).

# Deep Brain Stimulation **Part III**



# 8 Potentials in Deep Brain Stimulation

In this chapter the methodologies introduced in Parts I and II are discussed in the context of deep brain stimulation (DBS). For that, a sketch is given that describes how the works described in Chapters 3–7 could potentially be applied for DBS. The purpose of this is on the one hand to further motivate the methodologies introduced in this thesis and to highlight some ideas that have shaped this thesis. On the other hand, this part shall also be considered as collection of ideas that may serve as inspirations for future work.

## 8.1 DBS Overview

Deep brain stimulation is a neurosurgical procedure where electrodes are permanently implanted into deep regions of the brain. The implanted electrodes have the purpose to modulate pathological brain activity in order to alleviate symptoms of certain neurological diseases, such as Parkinson’s disease (PD) (Deuschl et al., 2006), or essential tremor (Perlmutter and Mink, 2006). The entire DBS procedure can be divided into three phases,

1. the *preoperative* phase, where the surgery is prepared and planned;
2. the *intraoperative* phase, where the electrodes are implanted; and
3. the *postoperative* phase, where the stimulation is initiated and maintained.

Since the first two phases are most important in the context of this thesis, they will be discussed in the following.

### 8.2 Planning

The purpose of the planning phase is to identify the desired target position and the trajectories of the electrodes through the brain. The particular structures that are to be stimulated by the electrodes depend on the disease that is being treated. For example, the *Subthalamic Nucleus* or the *Globus Pallidus interna* (Krack et al., 1998) are common targets for PD. The main challenges during the planning phase is the identification of the target position, and finding a (linear) path for the electrodes from the skull surface to the target position without harming any critical areas.

A common approach is that the neurosurgeon plans the surgery based on computed tomography (CT) and magnetic resonance imaging (MRI), where potentially multiple MRI sequences are used. Usually the target and entry points are manually chosen by the surgeon based on imaging. However, the accurate identification of the targets is very challenging. Reasons for this difficulty include low image contrasts, noise, artefacts and a general lack of the definitive knowledge which exact (regions of) brain structures need to be stimulated in order to achieve the best outcome for a particular patient.

By acquiring the CT image with some sort of fiducial markers, e.g. with the stereotactic frame mounted to the patient's head, and by performing a co-registration between the CT and MR images, transformations between the stereotactic coordinate system and the individual image coordinate systems are available. This allows the transfer of the surgery plan (intrinsically defined in image coordinates) to the stereotactic coordinate system that is eventually used to carry out the surgery.

**Idea 1.** (*Preoperative multi-image registration*)

*When more than two images are to be co-registered (e.g. more than a single MR image sequence is used, or the co-registration with an atlas is desired), the preoperative registration problem is a multi-image registration problem. In a similar fashion as described in Chapter 7, the multi-image registration can be obtained by first obtaining all pairwise image registrations, and then applying the transformation synchronisation method described in Chapters 6 and 7.*

#### 8.2.1 Navigation

The objective of the navigation phase is to place the electrodes at a position such that the outcome for the patient is “optimal”. In addition to the uncertainties occurring during the planning phase, another uncertainty that must be accounted for is that the brain moves after trepanation of the skull due to cerebrospinal fluid (CSF) coming out of the burr hole, which is known as *brain shift* (Miyagi et al., 2007; Khan et al., 2007). Moreover, (slight) uncertainties are also introduced due to the physically unavoidable mechanical

errors of stereotactic devices.

Depending on the considered target structures, two approaches to deal with these uncertainties, which may also be used in combination, are the acquisition of intraoperative microelectrode recordings (MERs) (Israel and Burchiel, 2004), and intraoperative test stimulations in order to assess how well the (awake) patient reacts to stimulations at particular sites (Deuschl et al., 2006). The motivation of using MERs is that certain brain structures have unique functional signatures that help to identify their location. For that, up to five microelectrodes are inserted in parallel, and the tissue activity at the electrodes' tips is recorded at regular intervals along the linear electrode trajectory. However, in order to make use of the full potential of the MERs, an excellent spatial visualisation ability is demanded from the neurosurgeon. In particular, the neurosurgeon *mentally* matches the MERs to the individual patient's anatomy as seen in the preoperative images. Given a patient-specific 3D model of the brain structures, this process can be simplified by algorithmically matching the MERs to the 3D model (Gemmar et al., 2008; Chaovalitwongse et al., 2011).

### 8.2.2 Patient-specific 3D Models

The previous elaborations emphasize one of the benefits of the availability of patient-specific 3D models for DBS. A further use-case of such a 3D model would be the post-operative analysis by visualising the 3D brain structures in combination with the implanted electrodes, the volume of tissue activated (Butson et al., 2007), and fibre tractography derived from diffusion imaging (Calabrese, 2016).

Whilst a patient-specific 3D model could in principal be created manually by the slice-wise annotation of the 3D image, it is too costly to be used as part of the daily clinical routine. Moreover, due to different anatomical structures having similar intensity patterns in MR images (Tu et al., 2008), the manual segmentation is error-prone. Further disadvantages of the slice-wise manual annotation include the consistency in only a particular image slice (Fischl et al., 2002), and reproducibility issues (Castro et al., 2006; Haegelen et al., 2012; Igual et al., 2013). An alternative to the manual annotation would be a fully automated approach. However, a problem of a fully automated procedure for the creation of patient-specific 3D models from images is the clinical acceptance for such highly individualised surgeries. Interactive segmentation approaches, where the user interacts with the segmentation algorithm and gradually produces the final segmentation, try to combine the advantages of both worlds.

One approach of implementing interactive deep brain structure segmentations is to use variational interpolation (Turk and O'Brien, 1999). In combination with the simultaneous multi-modality visualisation and histogram equalisation, this leads to an improved

consistency of the results compared to the slice-wise annotation (Bernard et al., 2014). The idea is that the user draws few contours on orthogonal slices of the objects to be segmented, and then the variational interpolation method creates a smooth 3D object from these contours. However, a downside of using variational interpolation is that it uses smoothness as non-specific general prior. Moreover, another difficulty is that the resulting interpolation due to the variational interpolation method may be non-intuitive and difficult to control in certain scenarios, which is especially a problem in a clinical setting.

**Idea 2.** (*Interactive brain structure segmentation based on SSMs*)

*The idea is to replace the smoothness assumption in variational interpolation by a more specific prior, namely SSMs. With that, a shape-aware surface reconstruction from sparse user-input can be achieved. In the following, a sketch is given how the methodologies introduced in this thesis can be used to implement such a method. For that, it is assumed that a collection of training images is available in order to learn the SSM.*

1. *Image Alignment: Using the approaches in Chapters 6 and 7, all training images are (coarsely) aligned by rigid transformations such that they are represented in the same coordinate system.*
2. *Image Annotation and Correspondence Establishment: After the brain structures of interest have been annotated in the training images, correspondences are established using the method introduced in Chapter 3.*
3. *Dimensionality Reduction: A multi-object shape deformation model with local support deformation factors is created (Chapter 4).*
4. *Interactive Segmentation: By incorporating the shape-aware surface reconstruction method (Chapter 5) into an interactive environment, the shape deformation model then serves as prior within an interactive setting.*

Points 1–3 can be considered the training phase, which has to be conducted only once. The outcome of the first three steps is a *deformable atlas* of the brain structures of interest. Once this atlas has been established, it can be used for the segmentation of unseen images. As such, for using the above procedure in a clinical setting, given the deformable atlas, only the fourth step needs to be conducted in order to obtain a patient-specific 3D model from a new patient’s images. In the following, some application-specific aspects of the four points are discussed.

1. *Image Alignment:* It is proposed to use the *rigid* transformation model in order to preserve angles and distances such that the original patient anatomy is retained.

This alignment is essential for the subsequent correspondence establishment and the dimensionality reduction.

2. *Image Annotation and Correspondence Establishment*: In order to create the SSM it is necessary that annotations of the images are available. However, this can be performed in an interactive manner by the approach based on variational interpolation, simultaneous multi-modality visualisation and histogram equalisation discussed above (Bernard et al., 2014). Since this procedure is conducted only during training phase, the disadvantage of the difficult intuition of variational interpolation in certain scenarios is less prominent compared to a clinical setting.
3. *Dimensionality Reduction*: The purpose of using a *multi-object* shape model is to capture the inter-relation of the deformations between the individual brain structures. By capturing this inter-relation, structures that are visually easy to identify can provide additional geometric information to guide the identification of structures that are more difficult to identify visually.

Due to the alignment of all images it can be expected that there are only minor pose variations between the individual patient’s images (and their segmentations). Thus, the alignment step commonly performed during PDM creation, cf. Section 1.1.2, is not conducted. Minor pose variations, as well as the relative pose between individual brain structures in the multi-object model, are implicitly captured by the PDM. Whilst translations can be perfectly captured using a model that is linear in shape space, for the rotational parts only a first-order approximation is captured by the linear model.

A model with *local support* factors, instead of global support factors, has two main advantages in this context. On the one hand, due to the locality of the deformations, the model is more flexible and thus better capable to adapt to the individual patient anatomy. On the other hand, due to the spatially localised deformations, structures that are far away from each other do not directly affect the deformations of each other, thus allowing a finer control of the interactive procedure. One potential disadvantage is that, due to the increased flexibility, it may be more difficult to fit a model with local support factors coarsely to the patient anatomy during the initialisation phase. Thus, another option that may be worth investigating is to consider a hybrid model that has both global and local support deformation factors.

4. *Interactive Segmentation*: The idea of the interactive environment is that the user gradually draws (partial) contours of those (parts of) objects that exhibit a distinctive boundary in the images. Immediately after the user draws a contour, points on the contour are extracted and used as input to the surface reconstruction method. The surface reconstruction method then fits the SSM to these points. This procedure is repeated until the user is satisfied with the result.

One particular problem is that the quality of SSMs depends on the amount of available training data (Heimann and Meinzer, 2009). However, the annotation of image data is costly and thus the quantity of available data is limited. This has the effect that the resulting SSM may not be able to generalise well to unseen data. One way of tackling this problem is to synthetically enlarge the training data, as for example done by Cootes and Taylor (1995). Another approach to achieve additional elasticity in the SSM, also closely related to the synthetic data enlargement, is the direct modification of the covariance matrix, as done by Wang and Staib (1998, 2000), which has also been adopted in the approach described in Chapter 4. For the particular application of shape-aware surface reconstruction, an alternative to modifying the SSM to obtain an increased flexibility is to not only allow for the linear SSM deformation, but also for additional free-form deformations. These free-form deformations can directly be reflected in the surface reconstruction procedure in a similar manner as the Coherent Point Drift algorithm does for nonlinear point-set registrations (Myronenko et al., 2007; Myronenko and Song, 2010). Another approach, which would also be able to deal with extreme outlier cases, is to provide the option of correcting inaccurate segmentations in a post-processing manner. The latter approach of a posterior correction is also interesting for learning the shape model in an iterative manner. For learning the shape model in an iterative manner, initially a (weak) SSM is trained from only few training instances. However, as soon as a new image has been satisfactorily segmented (by first fitting the weak SSM and then manually correcting the result), these images can be used in order to train a (slightly stronger) SSM.

# Bibliography

- Albrecht, T., Lüthi, M., Gerig, T., and Vetter, T. (2013). Posterior shape models. *Medical Image Analysis*, 17(8):959–973.
- Allasonnière, S., Amit, Y., and Trouvé, A. (2007). Towards a coherent statistical framework for dense deformable template estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1):3–29.
- Amenta, N., Bern, M., and Kamvysselis, M. (1998). A new Voronoi-based surface reconstruction algorithm. In *SIGGRAPH*.
- Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). SCAPE: shape completion and animation of people. In *SIGGRAPH*.
- Aprovitola, A. and Gallo, L. (2016). Knee bone segmentation from MRI: A classification and literature review. *Biocybernetics and Biomedical Engineering*, 36(2):437–449.
- Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(9):698–700.
- Avants, B. B., Tustison, N. J., Song, G., Cook, P. A., Klein, A., and Gee, J. C. (2011). A reproducible evaluation of ANTs similarity metric performance in brain image registration. *Neuroimage*, 54(3):2033–2044.
- Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2011). Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53.
- Bach, F., Mairal, J., and Ponce, J. (2008). Convex sparse matrix factorizations. *arXiv.org*.
- Bajaj, C. L., Bernardini, F., and Xu, G. (1995). Automatic reconstruction of surfaces and scalar fields from 3D scans. *SIGGRAPH*.
- Baka, N., de Bruijne, M., and Reiber, J. (2010). Confidence of model based shape reconstruction from sparse data. In *Biomedical Imaging: From Nano to Macro*.

## Bibliography

---

- Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.
- Bandeira, A. S., Singer, A., and Spielman, D. A. (2013). A Cheeger inequality for the graph connection Laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630.
- Barratt, D. C., Chan, C. S., Edwards, P. J., Penney, G. P., Slomczykowski, M., Carter, T. J., and Hawkes, D. J. (2008). Instantiation and registration of statistical shape models of the femur and pelvis using 3D ultrasound imaging. *Medical Image Analysis*, 12(3):358–374.
- Bartoli, A., Pizarro, D., and Loog, M. (2013). Stratified generalized procrustes analysis. *International Journal of Computer Vision*, 101(2):227–253.
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359.
- Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.
- Berger, M., Tagliasacchi, A., Seversky, L., and Alliez, P. (2014). State of the art in surface reconstruction from point clouds. *EUROGRAPHICS STAR*.
- Bernard, F., Gemmar, P., Hertel, F., Goncalves, J., and Thunberg, J. (2016a). Linear Shape Deformation Models with Local Support Using Graph-based Structured Matrix Factorisation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Bernard, F., Gemmar, P., Husch, A., and Hertel, F. (2012). Improvements on the feasibility of active shape model-based subthalamic nucleus segmentation. *Biomedical Engineering / Biomedizinische Technik*, 57(S1):42–45.
- Bernard, F., Gemmar, P., Husch, A., Saleh, C., Neb, H., Doods, G., and Hertel, F. (2014). Improving the Consistency of Manual Deep Brain Structure Segmentations by Combining Variational Interpolation, Simultaneous Multi-Modality Visualisation and Histogram Equalisation. *Biomedical Engineering / Biomedizinische Technik*, 59(S1):131–134.
- Bernard, F., Salamanca, L., Thunberg, J., Hertel, F., Goncalves, J., and Gemmar, P. (2015a). Shape-aware 3D Interpolation using Statistical Shape Models. In *Shape Symposium*.
- Bernard, F., Salamanca, L., Thunberg, J., Tack, A., Jentsch, D., Lamecker, H., Zachow, S., Hertel, F., Goncalves, J., and Gemmar, P. (2016b). Shape-aware Surface Reconstruction from Sparse Data. *arXiv.org*.

- Bernard, F., Thunberg, J., Gemmar, P., Hertel, F., Husch, A., and Goncalves, J. (2015b). A Solution for Multi-Alignment by Transformation Synchronisation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Bernard, F., Vlassis, N., Gemmar, P., Husch, A., Thunberg, J., Goncalves, J., and Hertel, F. (2016c). Fast correspondences for statistical shape models of brain structures. In *SPIE Medical Imaging*.
- Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359.
- Bertsekas, D. P. (1998). *Network Optimization: Continuous and Discrete Models*. Athena Scientific.
- Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.
- Billings, S. D., Boctor, E. M., and Taylor, R. H. (2015). Iterative most-likely point registration (IMLP): a robust algorithm for computing optimal shape alignment. *PLoS one*, 10(3).
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc.
- Blanc, R. and Székely, G. (2012). Confidence regions for statistical model based shape prediction from sparse observations. *IEEE Transactions on Medical Imaging*, 31(6):1300–1310.
- Blanz, V., Mehl, A., Vetter, T., and Seidel, H. P. (2004). A statistical method for robust 3D surface reconstruction from sparse data. In *3D Data Processing, Visualization and Transmission*, pages 293–300.
- Blanz, V. and Vetter, T. (1999). A morphable model for the synthesis of 3D faces. In *SIGGRAPH*.
- Bolle, R. M. and Vemuri, B. C. (1991). On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13.
- Boscaini, D., Eynard, D., Kourounis, D., and Bronstein, M. M. (2015). Shape-from-Operator: Recovering Shapes from Intrinsic Operators. In *Computer Graphics Forum*.
- Bouaziz, S., Tagliasacchi, A., and Pauly, M. (2013). Sparse iterative closest point. *Symposium on Geometry Processing*.
- Boumal, N. (2015). A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *arXiv.org*.

## Bibliography

---

- Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization*. Cambridge University Press.
- Burer, S. and Monteiro, R. D. (2005). Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444.
- Butson, C. R., Cooper, S. E., Henderson, J. M., and McIntyre, C. C. (2007). Patient-specific analysis of the volume of tissue activated during deep brain stimulation. *Neuroimage*, 34(2):661–670.
- Calabrese, E. (2016). Diffusion Tractography in Deep Brain Stimulation Surgery: A Review. *Frontiers in Neuroanatomy*, 10(45).
- Campadelli, P., Casiraghi, E., and Esposito, A. (2009). Liver segmentation from computed tomography scans: A survey and a new algorithm. *Artificial Intelligence in Medicine*, 45(2):185–196.
- Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11:1–11:37.
- Cashman, T. J. and Fitzgibbon, A. W. (2013). What shape are dolphins? Building 3D morphable models from 2D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):232–244.
- Castro, F. J. S., Pollo, C., Meuli, R., Maeder, P., Cuisenaire, O., Cuadra, M. B., Villemure, J.-G., and Thiran, J.-P. (2006). A cross validation study of deep brain stimulation targeting: from experts to atlas-based, segmentation-based and automatic registration algorithms. *IEEE Transactions on Medical Imaging*, 25(11):1440–1450.
- Cates, J., Meyer, M., Fletcher, T., and Whitaker, R. (2006). Entropy-based particle systems for shape correspondence. In *MICCAI Workshop on Mathematical Foundations of Computational Anatomy*.
- Chan, C. S., Edwards, P. J., and Hawkes, D. J. (2003). Integration of ultrasound-based registration with statistical shape models for computer-assisted orthopaedic surgery. In *SPIE Medical Imaging*.
- Chang, W. and Zwicker, M. (2009). Range Scan Registration Using Reduced Deformable Models. *Computer Graphics Forum*.
- Chaovalitwongse, W., Jeong, Y., Jeong, M. K., Danish, S., and Wong, S. (2011). Pattern Recognition Approaches for Identifying Subcortical Targets during Deep Brain Stimulation Surgery. *Intelligent Systems, IEEE*, 26(5):54–63.
- Chatterjee, A. and Govindu, V. M. (2013). Efficient and robust large-scale rotation averaging. In *International Conference on Computer Vision (ICCV)*.

- Chaudhury, K. N., Khoo, Y., and Singer, A. (2013). Global registration of multiple point clouds using semidefinite programming. *arXiv.org*.
- Chui, H. and Rangarajan, A. (2000). A feature registration framework using mixture models. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*.
- Chui, H. and Rangarajan, A. (2003). A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141.
- Combettes, P. L., Dũng, D., and Vũ, B. C. (2010). Dualization of signal recovery problems. *Set-Valued and Variational Analysis*, 18(3-4):373–404.
- Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer.
- Cootes, T. F., Marsland, S., Twining, C. J., Smith, K., and Taylor, C. J. (2004). Groupwise diffeomorphic non-rigid registration for automatic model building. In *European Conference on Computer Vision (ECCV)*.
- Cootes, T. F. and Taylor, C. J. (1992). Active Shape Models - Smart Snakes. In *British Machine Vision Conference*.
- Cootes, T. F. and Taylor, C. J. (1995). Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing*, 13(5):403–409.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1992). Training Models of Shape from Sets of Examples. In *British Machine Vision Conference*.
- Cremers, D., Rousson, M., and Deriche, R. (2006). A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape. *International Journal of Computer Vision*, 72(2):195–215.
- Dalal, P., Munsell, B. C., Wang, S., Tang, J., Oliver, K., Ninomiya, H., Zhou, X., and Fujita, H. (2007). A Fast 3D Correspondence Method for Statistical Shape Modeling. In *Computer Vision and Pattern Recognition (CVPR)*.
- Datteri, R. D. and Dawant, B. M. (2012). Automatic detection of the magnitude and spatial location of error in non-rigid registration. In *Biomedical Image Registration*, pages 21–30. Springer.
- Davies, R. H., Twining, C. J., Cootes, T. F., and Taylor, C. J. (2010). Building 3-d statistical shape models by direct optimization. *IEEE Transactions on Medical Imaging*, 29(4):961–981.

## Bibliography

---

- Davies, R. H., Twining, C. J., Cootes, T. F., Waterton, J. C., and Taylor, C. J. (2002). A minimum description length approach to statistical shape modeling. *IEEE Transactions on Medical Imaging*, 21(5):525–537.
- de Bruijne, M., van Ginneken, B., Viergever, M. A., and Niessen, W. J. (2004). Interactive segmentation of abdominal aortic aneurysms in CTA images. *Medical Image Analysis*, 8(2):127–138.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *SIGGRAPH*.
- Deuschl, G., Schade-Brittinger, C., Krack, P., Volkmann, J., Schäfer, H., Bötzel, K., Daniels, C., Deuschländer, A., Dillmann, U., and Eisner, W. (2006). A randomized trial of deep-brain stimulation for Parkinson’s disease. *N Engl J Med*, 355(9):896–908.
- Dryden, I. L. and Mardia, K. V. (1998). *Statistical Shape Analysis*. Wiley.
- Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the l1-ball for learning in high dimensions. In *International Conference on Machine Learning (ICML)*.
- Dulmage, A. L. and Mendelsohn, N. S. (1958). Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10(4):516–534.
- Durrleman, S., Pennec, X., Trounev, A., and Ayache, N. (2009). Statistical models of sets of curves and surfaces based on currents. *Medical Image Analysis*, 13(5):793–808.
- Durrleman, S., Prastawa, M., Charon, N., Korenberg, J. R., Joshi, S., Gerig, G., and Trounev, A. (2014). Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *Neuroimage*, 101:35–49.
- Edelsbrunner, H. and Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72.
- Eggert, D. W., Lorusso, A., and Fisher, R. B. (1997). Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290.
- Fang, Q. and Boas, D. (2009). Tetrahedral mesh generation from volumetric binary and grayscale images. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*.

- Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., and others (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–355.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Fletcher, P. T., Lu, C., and Joshi, S. (2003). Statistics of shape via principal geodesic analysis on lie groups. In *Computer Vision and Pattern Recognition (CVPR)*.
- Fleute, M. and Lavallée, S. (1998). Building a complete surface model from sparse data using statistical shape models: Application to computer assisted knee surgery. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.
- Fleute, M., Lavallée, S., and Julliard, R. (1999). Incorporating a statistically based shape model into a system for computer-assisted anterior cruciate ligament surgery. *Medical Image Analysis*, 3(3):209–222.
- Fonov, V. S., Evans, A. C., McKinstry, R. C., Almlí, C. R., and Collins, D. L. (2009). Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *Neuroimage*, 47:S102.
- Gal, R., Shamir, A., Hassner, T., Pauly, M., and Cohen Or, D. (2007). Surface reconstruction using local shape priors. *Symposium on Geometry Processing*.
- Galilei, G. (1638). Discorsi e dimostrazioni matematiche, intorno à due nuoue scienze attenenti alla mecanica & i movimenti locali.
- Gao, L., Heath, D. G., Kuszyk, B. S., and Fishman, E. K. (1996). Automatic liver segmentation technique for three-dimensional visualization of CT data. *Radiology*, 201(2):359–364.
- Gass, T., Székely, G., and Goksel, O. (2014). Detection and Correction of Inconsistency-based Errors in Non-Rigid Registration. In *SPIE Medical Imaging*.
- Gemmar, P., Tercero, M. E., Henrichs, T., Thesen, Y., Gronz, O., and Hertel, F. (2008). Fuzzy classifier for microelectrode recording-based target navigation in deep brain stimulation. In *Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*.
- Geng, X. (2007). Transitive Inverse-consistent Image Registration and Evaluation. *PhD Thesis*.

## Bibliography

---

- Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Gower, J. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.
- Gower, J. C. and Dijksterhuis, G. B. (2004). *Procrustes problems*, volume 3. Oxford University Press Oxford.
- Guo, G., Jiang, T., Wang, Y., and Gao, W. (2012). Recovering Missing Contours for Occluded Object Detection. *IEEE Signal Processing Letters*, 19(8):463–466.
- Guo, G., Jiang, T., Wang, Y., and Gao, W. (2013). 2-D shape completion with shape priors. *Chinese Science Bulletin*, 58(27):3430–3436.
- Hadani, R. and Singer, A. (2011a). Representation theoretic patterns in three dimensional Cryo-Electron Microscopy I: The intrinsic reconstitution algorithm. *Annals of mathematics*, 174(2):1219–1241.
- Hadani, R. and Singer, A. (2011b). Representation Theoretic Patterns in Three-Dimensional Cryo-Electron Microscopy II—The Class Averaging Problem. *Foundations of Computational Mathematics*, 11(5):589–616.
- Haeffele, B., Young, E., and Vidal, R. (2014). Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *International Conference on Machine Learning (ICML)*.
- Haegelen, C., Coupé, P., Fonov, V., Guizard, N., Jannin, P., Morandi, X., and Collins, D. L. (2012). Automated segmentation of basal ganglia and deep brain structures in MRI of Parkinson’s disease. *International Journal of Computer Assisted Radiology and Surgery*, 8(1):99–110.
- Harman, H. H. (1976). *Modern factor analysis*. University of Chicago Press.
- Heckel, F., Konrad, O., Karl Hahn, H., and Peitgen, H.-O. (2011). Interactive 3D medical image segmentation with energy-minimizing implicit functions. *Computers & Graphics*, 35(2):275–287.
- Heimann, T. and Meinzer, H.-P. (2009). Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563.
- Heimann, T., Wolf, I., and Meinzer, H.-P. (2006). Optimal landmark distributions for statistical shape model construction. In *SPIE Medical Imaging*.
- Heimann, T., Wolf, I., Williams, T., and Meinzer, H.-P. (2005). 3D active shape models using gradient descent optimization of description length. In *Information Processing in Medical Imaging*.

- Hein, M. and Bühler, T. (2010). An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In *Neural Information Processing Systems (NIPS)*.
- Herman, G. T., Zheng, J., and Bucholtz, C. A. (1992). Shape-based interpolation. *IEEE Computer Graphics and Applications*, 12(3):69–79.
- Hill, A., Cootes, T. F., and Taylor, C. J. (1995). Active Shape Models and the Shape Approximation Problem. *Image and Vision Computing*, 14(8):601–607.
- Holden, M., Hill, D. L., Denton, E. R., Jarosz, J. M., Cox, T. C., Rohlfing, T., Goodey, J., and Hawkes, D. J. (2000). Voxel similarity measures for 3-D serial MR brain image registration. *IEEE Transactions on Medical Imaging*, 19(2):94–102.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. A., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. *SIGGRAPH*.
- Horaud, R., Forbes, F., Yguel, M., Dewaele, G., and Zhang, J. (2011). Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):587–602.
- Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.
- Horn, B. K. P., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135.
- Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.
- Igual, L., Soliva, J. C., Hernández-Vela, A., Escalera, S., Vilarroya, O., and Radeva, P. (2013). A Supervised Graph-Cut Deformable Model for Brain MRI Segmentation. *Deformation Models*, pages 237–259.
- Israel, Z. and Burchiel, K. J. (2004). *Microelectrode recording in movement disorder surgery*. Thieme.
- Jenatton, R., Obozinski, G., and Bach, F. (2010). Structured sparse principal component analysis. *International Conference on Artificial Intelligence and Statistics*.
- Jenkinson, M. and Smith, S. (2001). A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2):143–156.
- Jia, H., Wu, G., Wang, Q., and Shen, D. (2010). ABSORB: Atlas building by self-organized registration and bundling. *Neuroimage*, 51(3):1057–1070.

## Bibliography

---

- Jian, B. and Vemuri, B. C. (2011). Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645.
- Jiang, B., Ding, C., Luo, B., and Tang, J. (2013). Graph-Laplacian PCA: Closed-form solution and robustness. In *Computer Vision and Pattern Recognition (CVPR)*.
- Joshi, S., Davis, B., Jomier, M., and Gerig, G. (2004). Unbiased diffeomorphic atlas construction for computational anatomy. *Neuroimage*, 23:S151–S160.
- Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Symposium on Geometry Processing*.
- Kendall, D. G. (1984). Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121.
- Khan, M. F., Mewes, K., Gross, R. E., and Škrinjar, O. (2007). Assessment of brain shift related to deep brain stimulation surgery. *Stereotactic and Functional Neurosurgery*, 86(1):44–53.
- Kotcheff, A. C. and Taylor, C. J. (1998). Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2(4):303–314.
- Kovnatsky, A., Bronstein, M. M., Bresson, X., and Vandergheynst, P. (2015a). Functional correspondence by matrix completion. In *Computer Vision and Pattern Recognition (CVPR)*.
- Kovnatsky, A., Glashoff, K., and Bronstein, M. M. (2015b). MADMM: a generic algorithm for non-smooth optimization on manifolds. *arXiv.org*.
- Krack, P., Pollak, P., Limousin, P., Hoffmann, D., Xie, J., Benazzouz, A., and Benabid, A. L. (1998). Subthalamic nucleus or internal pallidal stimulation in young onset Parkinson’s disease. *Brain*, 121(3):451–457.
- Kurtek, S., Klassen, E., Ding, Z., and Srivastava, A. (2010). A novel Riemannian framework for shape analysis of 3D objects. In *Computer Vision and Pattern Recognition (CVPR)*.
- Last, C., Winkelbach, S., Wahl, F. M., Eichhorn, K. W., and Bootz, F. (2011). A locally deformable statistical shape model. In *Machine Learning in Medical Imaging*.
- Learned-Miller, E. G. (2006). Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250.
- Leordeanu, M. and Hebert, M. (2005). A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In *International Conference on Computer Vision (ICCV)*.

- Leordeanu, M. D. (2010). Spectral graph matching, learning, and inference for computer vision. *PhD Thesis*.
- Liu, J. and Udupa, J. K. (2009). Oriented active shape models. *IEEE Transactions on Medical Imaging*, 28(4):571–584.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision (ICCV)*.
- Lu, M., Zheng, B., Takamatsu, J., and Nishino, K. (2011). 3D shape restoration via matrix recovery. In *Asian Conference on Computer Vision Workshops (ACCV Workshops)*.
- Lüthi, M., Jud, C., and Vetter, T. (2013). A unified approach to shape model fitting and non-rigid registration. In *Machine Learning in Medical Imaging*.
- Maciel, J. (2002). Global Matching: optimal solution to correspondence problems. *PhD Thesis*.
- Maciel, J. and Costeira, J. P. (2003). A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199.
- Maier-Hein, L., Franz, A. M., dos Santos, T. R., Schmidt, M., Fangerau, M., Meinzer, H.-P., and Fitzpatrick, J. M. (2012). Convergent iterative closest-point algorithm to accommodate anisotropic and inhomogeneous localization error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1520–1532.
- Maron, H., Dym, N., Kezurer, I., Kovalsky, S., and Lipman, Y. (2016). Point registration via efficient convex relaxation. *SIGGRAPH*.
- McGuinness, K. and O’Connor, N. E. (2010). A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444.
- Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278.
- Meyer, M. and Anderson, J. (2007). Key point subspace acceleration and soft caching. *SIGGRAPH*.
- Miyagi, Y., Shima, F., and Sasaki, T. (2007). Brain shift: an error factor during implantation of deep brain stimulation electrodes. *Journal of Neurosurgery*, 107(5).
- Munsell, B. C., Temlyakov, A., Styner, M., and Wang, S. (2012). Pre-organizing shape instances for landmark-based shape correspondence. *International Journal of Computer Vision*, 97(2):210–228.
- Myronenko, A. and Song, X. (2010). Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275.

## Bibliography

---

- Myronenko, A., Song, X., and Carreira-Perpinán, M. A. (2007). Non-rigid point set registration: Coherent Point Drift. In *Neural Information Processing Systems (NIPS)*.
- Neumann, T., Varanasi, K., Theobalt, C., Magnor, M., and Wacker, M. (2014). Compressed manifold modes for mesh processing. In *Computer Graphics Forum*.
- Neumann, T., Varanasi, K., Wenger, S., Wacker, M., Magnor, M., and Theobalt, C. (2013). Sparse localized deformation components. *SIGGRAPH Asia*.
- Nguyen, Q., Gautier, A., and Hein, M. (2015). A Flexible Tensor Block Coordinate Ascent Scheme for Hypergraph Matching. *arXiv.org*.
- Nocedal, J. and Wright, S. J. (2006). Numerical Optimization. Springer.
- Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832.
- Pachauri, D., Kondor, R., and Singh, V. (2013). Solving the multi-way matching problem by permutation synchronization. In *Neural Information Processing Systems (NIPS)*.
- Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*.
- Park, S., Guo, X., Shin, H., and Qin, H. (2006). Surface completion for shape and appearance. *The Visual Computer*, 22(3):168–180.
- Pauly, M., Mitra, N. J., Giesen, J., Gross, M. H., and Guibas, L. J. (2005). Example-Based 3D Scan Completion. *Symposium on Geometry Processing*.
- Perlmutter, J. S. and Mink, J. W. (2006). Deep brain stimulation. *Annu. Rev. Neurosci.*, 29:229–257.
- Pizarro, D. and Bartoli, A. (2011). Global optimization for optimal generalized procrustes analysis. In *Computer Vision and Pattern Recognition (CVPR)*.
- Pizer, S. M., Fletcher, P. T., Joshi, S., Thall, A., Chen, J. Z., Fridman, Y., Fritsch, D. S., Gash, A. G., Glotzer, J. M., and Jiroutek, M. R. (2003). Deformable m-reps for 3d medical image segmentation. *International Journal of Computer Vision*, 55(2-3):85–106.
- Purves, D., Augustine, G. J., Fitzpatrick, D., Katz, L. C., LaMantia, A.-S., and McNamara, J. O. (1997). *Neuroscience*. Sinauer Associates.
- Rajamani, K. T., Styner, M. A., Talib, H., Zheng, G., Nolte, L.-P., and Ballester, M. A. G. (2007). Statistical deformable bone models for robust 3D surface extrapolation from sparse data. *Medical Image Analysis*, 11(2):99–109.

- Rangarajan, A., Chui, H., and Bookstein, F. L. (1997). The softassign procrustes matching algorithm. In *Information Processing in Medical Imaging*.
- Rasouljan, A., Rohling, R., and Abolmaesumi, P. (2012). Group-wise registration of point sets for statistical shape models. *IEEE Transactions on Medical Imaging*, 31(11):2025–2034.
- Raya, S. P. and Udupa, J. K. (1990). Shape-based interpolation of multidimensional objects. *IEEE Transactions on Medical Imaging*, 9(1):32–42.
- Recht, B., Fazel, M., and Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501.
- Reuter, M., Schmansky, N. J., Rosas, H. D., and Fischl, B. (2012). Within-subject template estimation for unbiased longitudinal image analysis. *Neuroimage*, 61(4):1402–1418.
- Rodolà, E., Rota Bulò, S., Windheuser, T., Vestner, M., and Cremers, D. (2014). Dense non-rigid shape correspondence using random forests . In *Computer Vision and Pattern Recognition (CVPR)*.
- Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*.
- Rustamov, R. M., Ovsjanikov, M., Azencot, O., Ben-Chen, M., Chazal, F., and Guibas, L. (2013). Map-based exploration of intrinsic shape differences and variability. *SIGGRAPH*.
- Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.
- Schroers, C., Setzer, S., and Weickert, J. (2014). A Variational Taxonomy for Surface Reconstruction from Oriented Points. In *Computer Graphics Forum*.
- Schwarz, T., Heimann, T., Tetzlaff, R., Rau, A.-M., Wolf, I., and Meinzer, H.-P. (2008). Interactive surface correction for 3D shape based segmentation. In *SPIE Medical Imaging*.
- Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Robotics: Science and Systems*.
- Seghers, D., D’Agostino, E., Maes, F., Vandermeulen, D., and Suetens, P. (2004). Construction of a brain template from MR images using state-of-the-art registration and segmentation techniques. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

## Bibliography

---

- Seim, H., Kainmüller, D., Lamecker, H., Bindernagel, M., Malinowski, J., and Zachow, S. (2010). Model-based Auto-Segmentation of Knee Bones and Cartilage in MRI Data. *Proc. of Medical Image Analysis for the Clinic: A Grand Challenge*.
- Shahid, N., Kalofolias, V., Bresson, X., Bronstein, M., and Vandergheynst, P. (2015). Robust Principal Component Analysis on Graphs. In *International Conference on Computer Vision (ICCV)*.
- Shen, C. H., Fu, H., Chen, K., and Hu, S. M. (2012). Structure recovery by part assembly. *SIGGRAPH Asia*.
- Singer, A. (2011). Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20–36.
- Singer, A. and Shkolnisky, Y. (2011). Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming(). *SIAM Journal on Imaging Sciences*, 4(2):543–572.
- Sjostrand, K., Rostrup, E., Ryberg, C., Larsen, R., Studholme, C., Baezner, H., Ferro, J., Fazekas, F., Pantoni, L., Inzitari, D., and Waldemar, G. (2007). Sparse Decomposition and Modeling of Anatomical Shape Variation. *IEEE Transactions on Medical Imaging*, 26(12):1625–1635.
- Škrinjar, O., Bistoquet, A., and Tagare, H. (2008). Symmetric and Transitive Registration of Image Sequences. *International Journal of Biomedical Imaging*, 2008:1–9.
- Stegmann, M. B., Sjöstrand, K., and Larsen, R. (2006). Sparse modeling of landmark and texture variability using the orthomax criterion. In *SPIE Medical Imaging*.
- Stoll, C., Karni, Z., Rössl, C., Yamauchi, H., and Seidel, H.-P. (2006). Template Deformation for Point Cloud Fitting. *Symposium on Point-Based Graphics*.
- Styner, M. A., Rajamani, K. T., Nolte, L.-P., Zsemlye, G., Székely, G., Taylor, C. J., and Davies, R. H. (2003). Evaluation of 3D correspondence methods for model building. In *Information Processing in Medical Imaging*.
- Suinesiaputra, A., Frangi, A. F., Üzümcü, M., Reiber, J. H., and Lelieveldt, B. P. (2004). Extraction of myocardial contractility patterns from short-axes MR images using independent component analysis. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pages 75–86. Springer.
- Tan, J. H. and Acharya, U. R. (2014). Active spline model: A shape based model—interactive segmentation. *Digital Signal Processing*, 35:64–74.
- Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160.

- Taylor, J., Bordeaux, L., Cashman, T., Corish, B., Keskin, C., Sharp, T., Soto, E., Sweeney, D., Valentin, J., and Luff, B. (2016). Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In *SIGGRAPH*.
- Taylor, J., Stebbing, R., Ramakrishna, V., Keskin, C., Shotton, J., Izadi, S., Hertzmann, A., and Fitzgibbon, A. (2014). User-Specific Hand Modeling from Monocular Depth Sequences. In *Computer Vision and Pattern Recognition (CVPR)*.
- Tena, J. R., De la Torre, F., and Matthews, I. (2011). Interactive region-based linear 3d face models. *SIGGRAPH*.
- Thodberg, H. H. (2003). Minimum description length shape and appearance models. In *Information Processing in Medical Imaging*.
- Thompson, D. W. (1917). *On Growth and Form*. Cambridge University Press.
- Thunberg, J., Bernard, F., and Goncalves, J. (2015). On Transitive Consistency for Linear Invertible Transformations between Euclidean Coordinate Systems. *arXiv preprint arXiv:1509.00728*.
- Timinger, H., Pekar, V., von Berg, J., Dietmayer, K., and Kaus, M. (2003). Integration of interactive corrections to model-based segmentation algorithms. In *Bildverarbeitung für die Medizin*.
- Treece, G. M., Prager, R. W., and Gee, A. H. (2000). Surface interpolation from sparse cross sections using region correspondence. *IEEE Transactions on Medical Imaging*, 11(19):1106–1114.
- Tu, Z., Narr, K. L., Dollár, P., Dinov, I., Thompson, P. M., and Toga, A. W. (2008). Brain anatomical structure segmentation by hybrid discriminative/generative models. *IEEE Transactions on Medical Imaging*, 27(4):495–508.
- Turk, G. and O’Brien, J. F. (1999). Shape transformation using variational implicit functions. In *SIGGRAPH*.
- Üzümcü, M., Frangi, A. F., Sonka, M., Reiber, J. H. C., and Lelieveldt, B. P. F. (2003). ICA vs. PCA Active Appearance Models: Application to Cardiac MR Segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.
- van den Hengel, A., Dick, A. R., Thormählen, T., Ward, B., and Torr, P. H. S. (2007). Interactive 3D Model Completion. In *Digital Image Computing Techniques and Applications*.
- van Ginneken, B., de Bruijne, M., Loog, M., and Viergever, M. A. (2003). Interactive shape models. In *SPIE Medical Imaging*.

## Bibliography

---

- van Ginneken, B., Frangi, A. F., Frangi, R. F., Staal, J. J., Ter Haar Romeny, B. M., and Viergever, M. A. (2002). Active Shape Model Segmentation with Optimal Features. *IEEE Transactions on Medical Imaging*, 21:924–933.
- van Kaick, O., Zhang, H., Hamarneh, G., and Cohen Or, D. (2011). A survey on shape correspondence. In *Computer Graphics Forum*.
- Vercauteren, T., Perchant, A., Malandain, G., Pennec, X., and Ayache, N. (2006). Robust mosaicing with correction of motion distortions and tissue deformations for in vivo fibered microscopy. *Medical Image Analysis*, 10(5):673–692.
- Wachinger, C. and Navab, N. (2013). Simultaneous registration of multiple images: similarity metrics and efficient optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1221–1233.
- Walker, M. W., Shao, L., and Volz, R. A. (1991). Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367.
- Wang, L. and Singer, A. (2013). Exact and stable recovery of rotations for robust synchronization. *Information and Inference*.
- Wang, Y. and Staib, L. H. (1998). Boundary finding with correspondence using statistical shape models. In *Computer Vision and Pattern Recognition (CVPR)*.
- Wang, Y. and Staib, L. H. (2000). Boundary finding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):738–743.
- Windheuser, T., Schlickewei, U., Schmidt, F. R., and Cremers, D. (2011). Geometrically consistent elastic matching of 3D shapes: A linear programming solution. *International Conference on Computer Vision (ICCV)*.
- Xu, Y. and Yin, W. (2013). A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789.
- Xue, N., Doellinger, M., Ho, C. P., Surowiec, R. K., and Schwarz, R. (2015). Automatic detection of anatomical landmarks on the knee joint using MRI data. *Journal of Magnetic Resonance Imaging*, 41(1):183–192.
- Yang, D., Zhang, S., Yan, Z., Tan, C., Li, K., and Metaxas, D. (2015). Automated anatomical landmark detection on distal femur surface using convolutional neural network. In *International Symposium on Biomedical Imaging (ISBI)*.
- Yang, Y., Yu, Y., Zhou, Y., Du, S., Davis, J., and Yang, R. (2014). Semantic Parametric Reshaping of Human Body Models. In *3DV Workshop on Dynamic Shape Measurement and Analysis*.

- Zewail, R., Elsafi, A., and Durdle, N. (2007). Wavelet-Based Independent Component Analysis For Statistical Shape Modeling. In *Canadian Conference on Electrical and Computer Engineering*.
- Zhang, S., Zhan, Y., Dewan, M., Huang, J., Metaxas, D. N., and Zhou, X. S. (2011). Sparse shape composition: A new framework for shape prior modeling. In *Computer Vision and Pattern Recognition (CVPR)*.
- Zhang, S., Zhan, Y., Dewan, M., Huang, J., Metaxas, D. N., and Zhou, X. S. (2012). Towards robust and effective shape modeling: sparse shape composition. *Medical Image Analysis*, 16(1):265–277.
- Zheng, G., Rajamani, K. T., and Nolte, L.-P. (2006). Use of a dense surface point distribution model in a three-stage anatomical shape reconstruction from sparse information for computer assisted orthopaedic surgery: a preliminary study. In *Asian Conference on Computer Vision (ACCV)*.
- Zhou, F. and De la Torre, F. (2013a). Deformable Graph Matching. In *Computer Vision and Pattern Recognition (CVPR)*.
- Zhou, F. and De la Torre, F. (2013b). Factorized graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1774–1789.
- Zhou, X., Kitagawa, T., Hara, T., Fujita, H., Zhang, X., Yokoyama, R., Kondo, H., Kanematsu, M., and Hoshi, H. (2006). Constructing a probabilistic model for automated liver region segmentation using non-contrast X-ray torso CT images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.
- Zöllei, L., Learned-Miller, E., Grimson, E., and Wells, W. (2005). Efficient population registration of 3D data. In *Computer Vision for Biomedical Image Applications*.
- Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286.