Learning Non-rigid, 3D Shape Variations using Statistical, Physical and Geometric Models

Chao Zhang
PhD

University of York
Computer Science

June 2018

ABSTRACT

3D shape modelling is a fundamental component in computer vision and computer graphics. Applications include shape interpolation and extrapolation, shape reconstruction, motion capture and mesh editing, etc. By "modelling" we mean the process of learning a parameter-driven model.

This thesis focused on the scope of statistical modelling for 3D non-rigid shapes, such as human faces and bodies. The problem is challenging due to highly non-linear deformations, high dimensionality, and data sparsity. Several new algorithms are proposed for 3D shape modelling, 3D shape matching (computing dense correspondence) and applications.

First, we propose a variant of Principal Component Analysis called "Shell PCA" which provides a physically-inspired statistical shape model. This is our first attempt to use a physically plausible metric (specifically, the discrete shell model) for statistical shape modelling.

Second, we further develop this line of work into a fully Riemannian approach called "Shell PGA". We demonstrate how to perform Principal Geodesic Analysis in the space of discrete shells. To achieve this, we present an alternate formulation of PGA which avoids working in the tangent space and deals with shapes lying on the manifold directly. Unlike displacement-based methods, Shell PGA is invariant to rigid body motion, and therefore alignment preprocessing such as Procrustes analysis is not needed.

Third, we propose a groupwise shape matching method using functional map representation. Targeting at near-isometric deformations, we consider groupwise optimisation of consistent functional maps over a product of Stiefel manifolds, and optimise over a minimal subset of the transformations for efficiency.

Last, we show that our proposed shape model achieves state-of-the-art performance in two very challenging applications: handle-based mesh editing, and model fitting using motion capture data. We also contribute a new algorithm for human body shape estimation using clothed scan sequence, along with a new dataset "BUFF" for evaluation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Chapter 3:

$\mathcal{M}$	generic Riemannian manifold or shape space
dist, g	generic Riemannian distance and metric
δ	physical thickness of a thin shell
s^δ	thin shell with thickness $\delta > 0$
s	mid-layer of a thin shell
$\bar{s}$	reference shell in a stress-free state
ϕ	elastic deformation of a smooth shell
$\mathcal{W}$	deformation energy of smooth shell
$\mathbf{W}$	deformation energy of discrete shell
γ	bending energy weight
W_{mem}	membrane energy density
S	discrete shell as triangular mesh
$\mathbf{u}$	vertex displacement of discrete shell
G	first fundamental form of discrete shell in $\mathbb{R}^{2,2}$
$\mathcal{G}[\Phi]$	tangential distortion tensor (pointwise)
X, T, e	node, face and edge, respectively, of a triangle mesh
$\lvert T\rvert, \lvert e\rvert$	area of triangle T, edge length of e
θ_e	dihedral angle associated with an edge e
D_e	area associated with an edge e; $D_e = \frac{1}{3}(\lvert T\rvert + \lvert T'\rvert)$
$\mathbf{C}, \mathbf{G}$	covariance matrix, Gram's matrix
m	number of a given set of shells (triangular meshes)

Chapter 4:

exp	(time-discrete) exponential map
log	(time-discrete) logarithm
v	tangent vector
$\mathfrak{s}$	geodesic path
$\mathcal{T}, \mathcal{E}$	set of faces and edges
Q, b	3D rotation and translation, i.e. $Q \in SO(3)$, $b \in \mathbb{R}^3$

p	principal variations
α	vector of weights
K	time-discrete sampling length
J	number of principal variations used for defining local manifold
$\mathcal{I}, \mathbf{I}$	smooth geodesic interpolation, and discrete geodesic interpolation, respectively

Chapter 5:

T, T_F	mapping and its functional representation between two shapes
Φ	set of Laplacian Beltrami basis
$\mathbf{C_{ji}}$	functional map (matrix) from shape i to j
$\mathcal{C}$	set of functional map matrices
$V_k(\mathbb{R}^n)$	Stiefel manifold, the set of orthonormal k-frames in $\mathbb{R}^n$
$\mathbf{P}$	columns of functions represented as LB basis coefficients
$\mathbf{Q}$	functional representation of Laplacian operator (diagonal matrix of eigenvalues)

ACKNOWLEDGEMENTS

I would like to extend my thanks to the many people in my life, who generously contributed to this thesis in time, efforts or thoughts.

Firstly I would like to give a massive amount of thanks to my supervisor, William Smith. Through my PhD study, Will is extremely generous about sharing his limited time and precious experiences. His insightful thoughts inspire me in determining research directions and devoting research efforts. His suggestions on collaborative research are extraordinary valuable and indispensable for my PhD study and my future career as well.

I was grateful to have the opportunities to collaborate with many brilliant researchers across the world. Martin Rumpf (University of Bonn) and Behrend Heeren (University of Bonn) kindly invited me for a couple of wonderful visits to Bonn and showed me the intriguing mathematical bits of our work. I also felt very lucky to be invited for an internship at Max Planck Institute (Tübingen). The internship was a highlight in my PhD journey. In particular, it was a pleasure to work with Gerard Pons-Moll and Serge Pujades, and to receive the guidance from Michael Black.

I thank British Machine Vision Association (BMVA) for generously offering me the travel bursaries to present my work at CVPR conference in Las Vegas, USA.

I would not be able to make it without the happy hours over the many afternoon teas with my fellow students, Zhe (Hugo), Jianjia, and Shuai. I would like to thank my office mates Zhenyu (James), Fred, Anil, Dizhong and Sarah for their quiet accompany and funny conversations which keep me cheerful.

To my parents, I would never take their unconditional love and consistent support for granted. It is their financial and emotional support which hold my back over the unpaid days. They may not be able to read the above words, but it is undoubted that I love them more than any other sons or daughters to their parents.

Last but not least, I would like to thank my best friend, my partner, my soulmate, Wang Zhaoming. It is her who turns each ordinary day, with few or many bugs, to every extraordinary day. I simply could not imagine how boring and lifeless my life could be without her tolerances and scoldings. Her restless passion in pursuing unconventional knowledge impacts me to be positive and wide-minded. Thanks so much for taking good care of us.

DECLARATION

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. The contributions contained in this thesis are my own work. Main contents have been published in the following papers. For all these works I made the major contribution in design, implementation and experiments. All sources are acknowledged as References.

- **Chao Zhang, Behrend Heeren, Martin Rumpf, William Smith**, "Shell PCA: statistical shape modelling in shell space", in Proc. of International Conference on Computer Vision (ICCV), Santiago, Chile, 2015
- **Chao Zhang, William Smith, Arnaud Dessein, Nick Pears, Hang Dai**, "Functional Faces: Groupwise Dense Correspondence using Functional Maps", in Proc. of Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, 2016
- **Chao Zhang, Sergi Pujades, Michael Black, Gerard Pons-Moll**, "Detailed, accurate, human shape estimation from clothed 3D scan sequences", in Proc. of Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 2017
- **Behrend Heeren**[*]**, Chao Zhang**[*]**, Martin Rumpf, William Smith**, "Principal Geodesic Analysis in the Space of Discrete Shells", Computer Graphics Forum (Proceedings of SGP), Paris, France, 2018
  [*] indicates equal contributions

Chao Zhang
June, 2018

CHAPTER 1

INTRODUCTION

1.1 Context and Setup

Computer representations of geometry are at the heart of many problems in digital design and multimedia production. They are widely used in today's computer games, film creation, 3D printing, computer-aided design, augmented reality, simulation, ergonomics and many other areas. Modelling 3D shapes (shape collections) is an important tool for analysing and understanding natural objects and their variations. Real world objects are naturally perceived as 3D shapes. Only recently however have these large 3D datasets [2, 20] become available and easily accessed with the help of rapidly advancing techniques to capture objects such as Microsoft Kinect in the consumer market and custom-built commercial solutions in the high end.

As the objects types increase and the size of datasets explodes, it becomes a necessity to study the similarity and dissimilarity manifested in the data. For example, given tens of thousands of human body scans, people may wonder what does the average male or female look like? Also, it is of great interest to visualise how bodies vary from person to person. Thus, statistical studies of 3D shapes could offer us deep insights on people and many other creatures. Normally, a typical 3D scanned mesh has tens or hundreds of thousands of vertices and triangles, corrupted with noise. At first glance, dealing with triangular meshes seems quite cumbersome due to its high dimensionality and unordered nature. However, mesh simplification and registration approaches are there to process acquired data and output clean, modest size 3D meshes with consistent vertex semantics (i.e. a particular vertex has the same semantic meaning across different shapes). It is now when the 3D modelling techniques come into play. Given a set 3D meshes of a particular type, say, human faces, the essential goal of modelling is to learn the characteristic shape variations of 3D faces. In parallel, research on 2D shape modelling also shed light on this direction for understanding 3D shapes. Take 2D face recognition for instance, Active Shape Models (ASM) show that the facial variations caused by identity changes could be captured by many fewer modes compared to the number of pixels [21]. Along this line, an important

school of methods called 3D parametric models emerged and become useful tools in dealing with 3D data. Typical parametric models include Blend shapes [22], skeleton models [23] and statistical models [24]. Hence, both the variability of shapes and nonlinear joints rotations could be encoded as a low dimensional vector of coefficients. To this end, the problem becomes how to learn such a suitable parametric model and use it for various applications instead of dealing directly with high dimensional original shapes.

The most commonly used parametric models are statistical models. Statistical models of the shape or appearance of shapes are widely used in computer vision and graphics. These methods aim to extract an average shape and a series of orthogonal principal modes that are capable of approximating original shapes. It has been successfully applied in various tasks of 3D objects, such as human faces and bodies. Depending on the nature of objects and their variations, a skeleton-based template has to be adopted to deal with articulations, e.g human poses.

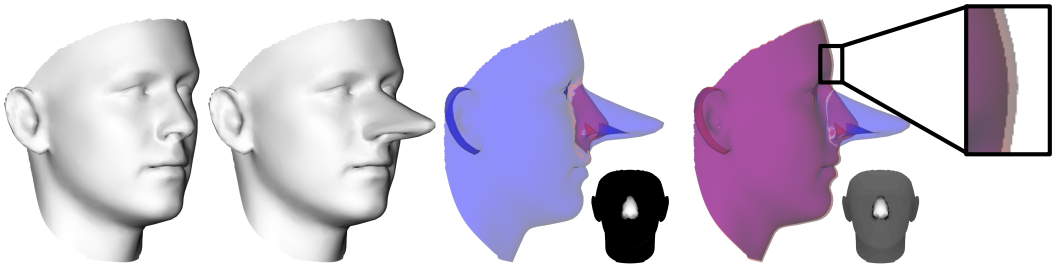
Statistical models also bring some attractive properties for practical use. First of all, it is straightforward to learn such a model on a given dataset (shapes in correspondence), especially when a skeleton is not needed. However, bringing scanned shapes into correspondence is itself a challenging problem already. Second, to approximate seen or unseen samples of the class, linear combinations of the chosen modes are used. Therefore, most optimisation problems boil down to the choice of a vector of coefficients. Third, due to the low dimensional prior nature, parametric models can be used to constrain synthesis or analysis problems. Their parameter space provides a compact representation which has been proved to be very useful in applications such as motion tracking, shape reconstruction, image synthesis, mesh editing, landmarks detection and so on.

Several measures have been proposed to numerically evaluate statistical models. The first is to capture the variability in the training data as efficiently as possible, measured by the *compactness*. Hence, one purpose of statistical modelling is dimensionality reduction. The second is to approximate unseen data as accurately as possible, measured by the *generalisation* ability of the model. Third, it is required that the model exhibits high *specificity*, i.e. the model only generate instances that are plausibly members of the object class being modelled.

1.2 Problems and Challenges

There are still many problems impeding the further applications of parametric 3D models. To emphasise the goals of this thesis, several key challenges are listed below:

1. 3D scans have different number of nodes and topologies after capture and need to be processed before building a parametric model. The common tool in use is "template-fitting" in which a hand-crafted template is fitted to each individual scan. However, the choice of template will influence the final shape representation. Also, individually aligning shapes one by one will ignore the knowledge of the shape collection as a whole. A shape that seems to be difficult to fit on its own, however may be easier once the context of other



(a) Original shape (b) Deformed shape (c) Trimmed alignment (d) Procrustes alignment

Figure 1.1: Different alignments result in different vertex representations (hence different vertex-based models)

shapes is used. In other words, computing correspondence for multiple shapes of the same object should benefit from each other. We would seek to match shapes in a groupwise manner, and meanwhile avoid the computational expense.

2. Important sources of shape variability are often highly nonlinear. For example, it is common to deal with nonlinear motion(such as articulation, bending, flapping and soft tissue dynamics) and nonlinear shape changes(such as aging, gaining or losing weight or shape differences between different individuals.) Vertex-based models are simple and efficient to use, however, it is difficult to see how vertex displacements can be used to describe highly nonlinear deformations. In practice, building such a model often requires human-expert knowledge in order to create a hand-crafted skeleton model. We would like to be able to capture these variations without using a data-specific skeleton.

3. Traditional vertex-based models often require a wealth of training data to be able to extract meaningful variations. In practice, the dimensionality of the raw shape data is usually orders of magnitude larger than the number of training samples. As such, we would like parametric models that are physically valid and could interpolate between sparse training shapes. Moreover, the model is expected to learn meaningful variations even training data is sparse.

4. A widespread problem in shape modelling is that shapes have to be carefully aligned to remove the effect of rigid body motion. Usually, generalised Procrustes analysis [25] is used and the metric to be minimised is the sum of squared Euclidean distances between landmarks (cf. Fig. 1.1). However, different metrics yield different alignments. And, since variations are usually described in terms landmark or vertex displacements, different alignments would yield different statistical shape models. Moreover, for nonrigid deformations a meaningful rigid alignment may not exist. Instead of choosing a metric to be minimised during alignment, we would like the model to be invariant to rigid body motion completely.

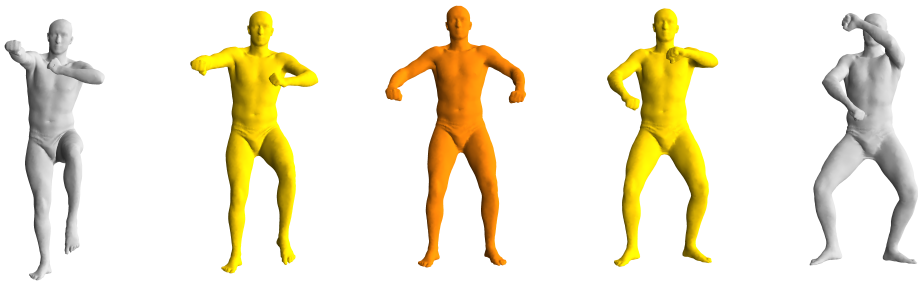
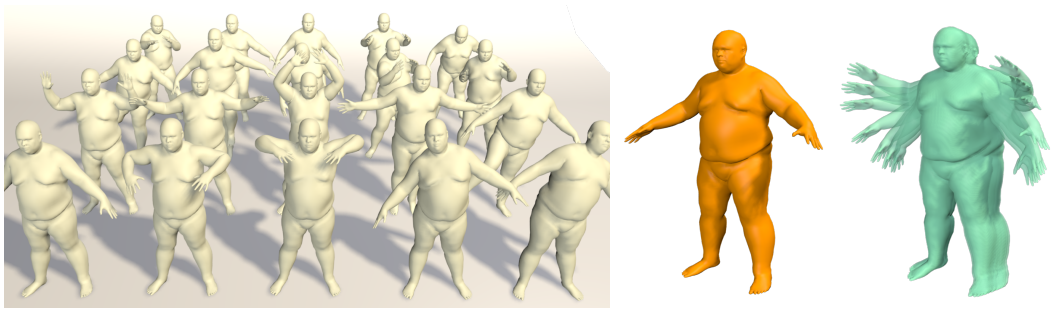


Figure 1.2: Interpolations between two human body poses.



(a) input data (b) average (c) principal mode

Figure 1.3: An example statistical model of human body shapes.

1.3 Contributions

The contributions of this work concern physically-inspired statistical shape models, multiple shape matching (by which we mean computing dense correspondence), and their applications. While the scope of the chapters differ by ingredient (correspondence, modelling, or applications), they all illustrate techniques for better analysis of 3D shapes and the use of manifold representations and optimisation on manifolds is a recurring theme. These contributions will be discussed in detail now.

1.3.1 Shape Modelling in the Space of Shells

First, we show how to perform statistical analysis in "shell space". Thin shells, or their discrete counterparts discrete shells, can be considered to reside in a shell space in which the notion of distance is given by the elastic energy required to deform one shape into another. An example of geodesic path in such a space is shown in Fig. 1.2. Concretely, the discrete shell model equips a triangle mesh with a physical model in which deformation of the mesh dissipates energy. Treating dissipated energy as a distance measure leads to a nonlinear manifold representation. It is in this setting that we show how to perform statistical analysis of a set of shapes, providing a hybrid between physical and statistical shape modelling.

In Chapter 3, we have presented an extension of Principal Component Analysis to operate

Figure 1.4: Human body shapes in dense correspondence. (Same colour indicates corresponding points)

in shell space, based on vertex-displacement. We introduce for a given set of input shapes (i) a general notion of an average and (ii) a covariance operator as a generalisation of a covariance matrix. Principal components are obtained via an eigen-decomposition of the Gram matrix arising from an inner product based on the Hessian of an elastic energy.

In Chapter 4, we show how to learn nonlinear, physically plausible modes of shape variation from a set of highly varying training shapes, which can be used for projection onto a low dimensional submanifold and thus sparse representation by a small set of weights (cf. Fig. 1.3). The model is fully Riemannian, works directly with meshes and does not require problem-specific articulated skeletons yet it is able to handle different kinds of nonlinear deformation. Most importantly, the whole framework is rigid body motion invariant and does not need alignment step for preprocessing at all.

1.3.2 Groupwise Shape Correspondence via Functional Maps

Next, we contribute a method for computing dense correspondence for multiple non-rigid 3D shapes (cf. Fig. 1.4). In Chapter 5, we present a method for dense 3D correspondence that addresses the scalability of groupwise methods and elegantly handles multiple notions of equivalence between shapes. We heuristically design a set of real-valued functions that are appropriate specifically to the problem of face correspondence, although the proposed method could be applied to general non-rigid shapes of any class. Specifically, we adopt the functional maps representation and show how groupwise optimisation of functional correspondence can be

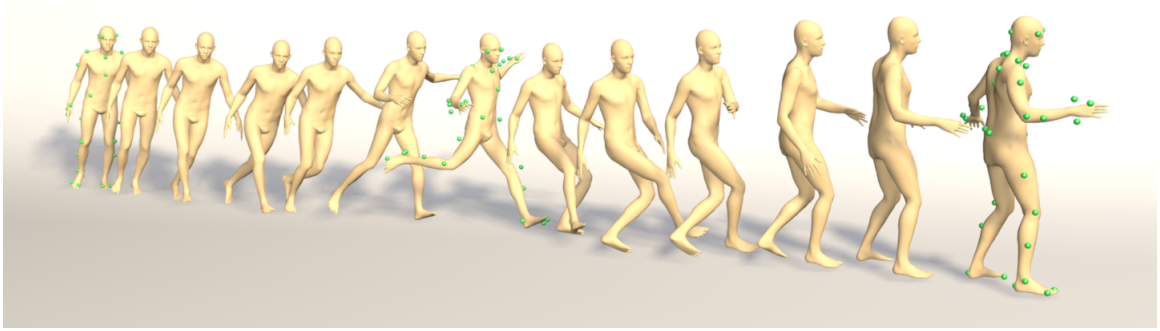


Figure 1.5: Estimated human body shapes and poses from motion capture markers data (in Green) without using Skeleton model.

performed via cycle-consistent optimisation on the product of Stiefel manifolds.

1.3.3 Applications of Statistical Human Body Models

Finally, we demonstrate some interesting applications using either the state-of-the-art body model [26] or the general shape model learned on human body data in Chapter 4.

One primary application using statistical body models is the estimation of human pose and intrinsic shape from observations. In particular, we address the problem of estimating the body shape and poses of a person wearing clothing from 3D scan sequences or visual hulls. We contribute a new approach to recover a personalised shape of the person. The estimated shape deviates from a parametric model to fit the 3D scans. We demonstrate the method using high quality 4D data as well as sequences of visual hulls extracted from multi-view images. We also make available "BUFF", a new 4D dataset that enables quantitative evaluation.

Furthermore, we present two specific applications: model-constrained mesh editing and reconstruction of a dense animated mesh from sparse motion capture markers using the statistical knowledge as a prior. Our models are learned from much less training data and are entirely mesh-based without using any articulation model. Nevertheless, our results are qualitatively very similar to the state-of-the-art results.

1.4 Thesis Outline

The remainder of this thesis is organised as follows:

Chapter 2: Related Work: We describe related work both in 3D shape correspondence and in 3D shape modelling.

Chapter 3: PCA in Shell Space: An extension of Principal Component Analysis in shell space is described and applied in extracting principal modes from sparse human shapes and reconstructing 3D human faces.

Chapter 4: PGA in Shell Space: A rigid body motion invariant Principal Geodesic Analysis is

developed to analyse shape variations without the need of alignment and to reconstruct articulated deformations without the use of skeleton models.

Chapter 5: Groupwise Shape Matching via Functional Maps: A groupwise shape matching framework is introduced to solve multiple non-rigid shape matching problem, with a special treatment of 3D faces.

Chapter 6: Applications: A method to estimate body shape and poses from 3D scan sequences is described using the state-of-the-art statistical body model. Two specific applications: data-driven mesh editing and model constrained mesh reconstruction from motion capture markers are demonstrated using our Riemannian statistical model.

CHAPTER 2

RELATED WORK

2.1 Dense 3D Shape Correspondence

Given two 3D shapes represented as 2-manifold triangle meshes, by shape correspondence we seek to establish a meaningful mapping between them. The mapping can be between the two sets of mesh vertices, between two coarse sets of feature points selected on the meshes, or a continuous one between all points on the two manifolds. This is a fundamental problem in computer graphics and shape modelling, with such applications as texture mapping [27], mesh morphing [28], and shape registration [29]. Both rigid shape matching and non-rigid shape matching are relatively well-studied problems with several recent books and surveys such as [4, 30]. It is out of scope for the thesis to review all existing shape matching methods. Instead, this section concentrates on two categories of existing methods: 1) methods that use functional maps as the correspondence representation; 2) methods that aim at multiple shape matching (i.e. treating all shapes in a collection simultaneously). To begin with, a brief introduction to existing methods is given to lay the foundation for the elaboration of the mentioned two aspects.

Point-based correspondence. A majority of shape matching methods represent the mapping between shapes as a point-to-point correspondence. This representation renders map estimation intractable due to the exponential nature of the space of all possible correspondences. Most methods choose to obtain a sparse landmarks point-to-point correspondence first, and then extend it to dense correspondence [31–38].

Part or segment based correspondence. Another set of methods attempt to put shape parts or segments into correspondence [39–43]. These techniques either pre-segment the shape and try to compute correspondences, or seek to jointly solve the segmentation and correspondence problems with a optimisation framework.

Alignment-based method. There are a few works which propose to optimise the deformation of one shape by aligning it with another [44, 45]. Nevertheless, in most cases, reliable point correspondences are still required either in pre-processing stage as feature matches or during the

alignment step.

2.1.1 Functional-maps based shape correspondence

A novel map representation for shape matching called "functional maps" was introduced in [46]. Instead of building spatial point-to-point correspondence, this method turns to matching real-valued functions defined over the surfaces of the shapes. Due to the linear nature of the maps in functional space, multiple constraints can be incorporated into a linear system and the solution could then be obtained by solving it in the least squares sense. Moreover, the functional map representation is highly compact - the correspondence between high resolution meshes can be captured in a matrix of typical size 30×30.

In the functional map framework [46], the Laplacian-Beltrami eigenfunctions of the shapes play a key role. It is used to represent the linear map as a matrix mapping the Fourier coefficients from one shape to another. The assumption that the Laplacian bases computed from shapes independently are compatible is often unrealistic. This is due to that eigenfunctions are only defined up to sign flips, and the ordering of the eigenfunctions is not preserved across shapes, especially in the higher frequencies. Moreover, harmonic bases are often incompatible for shapes which are not isometric with each other. To fix this, Kovnatsky et al. [47] propose the construction of common approximate eigenbases for multiple shapes using approximate joint diagonalisation algorithms. The coupling between the joint approximate eigenbases depends on a given set of corresponding functions on two shapes, e.g. sparse point correspondence.

The choice of functions also plays a crucial role in applying functional maps framework. Rather than identifying and using the most stable functions (which might be not informative at all), given a set of shapes with known correspondences (therefore ground truth functional maps available), Corman et al. [48] apply a supervised learning approach to find the optimal weights for those functions and to improve the quality of the obtained functional maps. Furthermore, they demonstrate how to extract the most reliable functional subspaces across shapes by computing a set of orthonormal basis with decreasing confidence values.

Assuming that shapes are nearly isometric, Pokass et al. [49] observe that the functional maps are just sparse matrix exhibiting nearly diagonal structure. The method only uses some repeatable regions and propose to optimise a permutation matrix and functional correspondence in an alternative way. The problem is formulated as a permuted sparse coding one. To handle partial matching, the method is extended to a robust permuted sparse coding by adding an additional sparse coding term addressing outliers. This method successfully recovers dense correspondence while using very limited information from two shapes, i.e. only a region detector without a feature descriptor.

Jointly analysing shapes collection enables a variety of applications in data-driven shape analysis such as shape co-segmentation. Unlike point-based maps, functional maps provide a concise representation encoding similarities at multiple levels. Huang et al. [50] introduce

a framework for computing consistent functional maps within man-made shape collections. To allow for partially similar shapes, cycle-consistency via constrained matrix factorisation is applied to allow the extraction of a small number of latent basis functions across shapes, and therefore to enable the usage of low-rank matrix recovery method.

A particular challenging setting for shape matching is partial correspondence, where one is shown only a subset of the shape and has to match it to a full version. Such applications are common for instance in robotics where one has to match an object scanned by means of a 3D scanner with a reference shape known in advance. Rodola et al. [51]propose to exploit the changes of the Laplacian-Beltrami basis as a result of part removal as a prior on the spectral representation of the functional maps. In particular, the approach looks for the largest and most regular parts that minimise correspondence distortion. Moreover, a new benchmark to evaluate partial correspondence for deformable shapes is introduced.

While functional maps become a very successful representation in shape matching, it is also noted that it is limited in recovering the point-to-point correspondence from such as map, by mapping peaks on one shape to another. Rodolà et al. [52] consider the point-wise map recovery problem as a point cloud alignment problem in the embedded space and use a regularised probabilistic model to impose smoothness constraint. A simple symmetrisation of the standard nearest neighbour approach is shown to improve the results consistently. Furthermore, in [53] they extend the technique to solve the partial-to-full shape matching problem, and provide an efficient GPU implementation to account for large shape matching problems.

Similar to partial functional correspondence, Litany et al. [54] consider a more general setup where the shapes are allowed to undergo non-rigid deformations and only partial views are available. The method simultaneously solves the segmentation of the reference shape, and for a dense correspondence to (subsets of) the parts.

Although classical descriptor preservation constraints are used widely in [46] and its follow-up works, this formulation requires many descriptor functions to obtain good results. Nogneng and Ovsjanikov [55] pointed out this problem and showed that much more information can be encoded into function preservation constraints via commutativity with an underlying map, which is already seen in the original work but with Laplace-Bletrami operator. The new commutativity constraints also help the functional map estimation process closer to point-to-point maps, while maintaining the linear complexity of the optimisation.

Functional maps is not only a very useful tool for shape matching, but also an effective way to many other map-based applications. Boscaini et al. [56] present a framework for reconstructing shapes from intrinsic operators such as shape difference operator and Laplacian operator, using functional maps to representing the correspondence.

Shape comparison is a fundamental operation in shape analysis, and most approaches use a single scalar number to denote the similarity. A detailed understanding of how two shapes differ goes beyond a single score. Rustamov et al. [57] develop a novel difference operator, derived

from a shape map, aimed at providing detailed information about the location and nature of the differences or distortions between the two shapes being compared.

2.1.2 Multiple shape correspondence

The problem of multiple shape correspondence is fundamental in many applications including partially overlapping range scans [58], image keypoint matching [59], assembling fractured surfaces [60] and structure from motion [61]. We only discuss approaches that are applied to **deformable 3D shapes**.

The problem can be generally stated as: given n input shapes $S_1, S_2, \cdots, S_n$, find a meaningful relation (or mapping) between their elements. Traditional correspondence matching is often done between a pair of shapes (one is called source shape, the other is target shape), and is thus called pairwise shape matching. Recently, a specialised form of shape matching called multiple shape matching is to work on a set of shapes simultaneously (i.e. $N > 2$). As 3D data increase in size, multiple shape matching is preferred over traditional pairwise shape matching for several reasons. First, the group of shapes can provide information to distinguish what structure or parts are common to all the shapes and should be considered in the correspondence. Second, the output of shape matching is often fed to a statistical shape modelling method and thus consistency is vital. Moreover, pairwise methods using intrinsic nature of shapes could not resolve symmetry ambiguity. For example, a common used invariant property geodesic distance is preserved both in the straight match and its reflective version. Another point is that pairwise methods usually work well if and only if the two shapes are similar and involve nearly isometric deformation.

Although multiple shape correspondence could be solved by simply computing pairwise correspondence between all the pairs of shapes in the group, the brute force solution is not used in practice because it results in a large number of pairwise maps. Compared to pairwise methods, multiple shape matching could benefit from the contextual information of a shape collection. It leads to improvements in terms of correspondence robustness, consistency and efficiency as shown in [62]. Besides that, a full set of pairwise maps are often redundant and thus not necessary. Assuming the deformation is nearly-isometric amongst the shape collection, the mapping from j to i could be approximated as the inverse of i to j. Using mapping composition, the mapping from i to k could be obtained with map i to j and map j to k. Another important point made in [63] is that, unlike pairwise mapping, the mapping between two dissimilar shapes could be obtained via a route passing more similar shapes, which reduces the accumulated error.

A classical method in solving multiple shape correspondence is called “minimum description length” [64]. This method uses an objective function which is defined in an information theoretic framework. They suggest that the optimal correspondence between a shape collection is the one that gives the most “compressible” model - i.e. minimises the description length of the training data. However, this attractive but highly computational-expensive approach is not suitable for high resolution meshes.

Most of the works on multiple shape matching choose to improve a given set of pairwise maps rather than computing them directly. Nguyen et al. [62] propose to identify "bad" maps and replace them by compositions of other maps, provided that most of the initial maps are accurate. The method is based on the estimation of "inconsistency" and its connection with "accuracy". The work use model graph as the data structure, where each node denotes one shape and the edge weight connecting two shapes is the "inconsistency".

Huang et al. [63] employ similar cycle-consistency criteria as [62] to enforce neighbor-preservation and alignment with the initial maps. The method proceeds by first building soft maps from a set of automatically chosen base shapes to all other shapes in the collection, and then computes a point-to-point map from each base to other shapes using a global optimisation. These are used to create a compact graphical data structure from which globally optimal cycle-consistent maps are extracted.

Huang et al. [65] propose to formulate the multiple shape matching problem as a semidefinite program (SDP). Representing correspondence as a point-to-point map in a discrete way, the maps are expected to be sparse and symmetric matrices. After convex relaxation, either full point-to-point correspondence or only selected point-to-point correspondence could be recovered with theoretical guarantees. A framework using functional maps as representation for cycle-consistent correspondence is also presented in [50].

Aiming to explicitly minimise the distortion of the maps over all shape pairs, Sahillioğlu and Yemez [66] present a multiple shape matching method based on dynamic programming. The approach only works well when matching shape extremities, and it is sensible to outlier shapes. Especially, its accuracy is order-dependent just like applying pairwise method sequentially.

More recently, Cosmo et al. [67] introduce an approach to obtain cycle-consisitent matches without requiring initial pairwise solutions by optimising a joint measure of metric distortion directly. The formulation of the problem as a series of quadratic programs with sparsity-inducing constraints also allows for partiality and outlier shapes.

2.2 Data-driven 3D Shape Modelling

2.2.1 Elastic shape modelling

Physically-based elastic energy models have been widely used for simulation, interpolation, mesh editing and, more recently, statistical modelling. The classical model for elastically deformable surfaces is the shell model, originally introduced in a graphics context by Terzopoulos et al. [68], for thin, flexible materials. Grinspun et al. [69] introduced the discrete shell model in which a triangle mesh is a spatially-discrete representation of the mid-surface of a shell. The model was used for simulation of deformable materials under physical forces. In the direction of improving efficiency, the as-rigid-as-possible (ARAP) framework [12] is based on alternating minimisation

over vertex positions and local rotations of an energy that measures deviation from rigidity. Von Radziewsky et al. [70] recently showed how model reduction can be used to efficiently evaluate elastic deformation models, including the discrete shell energy. This enables elastic models to be used in real-time applications (see also [71]).

2.2.2 Articulated models

The natural representation for deformations due to articulation is a skeleton model comprised of joint locations and relative orientations. Heap and Hogg [72] extended classical 2D landmark-based statistical modelling into the articulated domain by building linear models over joint angles rather than vertex positions. For 3D shapes, skeletons are used to deform dense surface models (usually meshes) via a process known as skinning [73]. In their Shape Completion and Animation of People (SCAPE) framework, Anguelov et al. [3] learn a combined pose and deformation model and a model of the variability in body shape. A skeleton is used to drive mesh deformation using a method based on deformation transfer [74] and variations in body shape are learnt using a linear model of bodies in a standard pose. The Dyna model [11] is built on top of SCAPE and adds a linear dynamics model whose coefficients depend upon the skeleton pose. The SMPL model [**?**] shows that pose-dependent blend shapes can depend linearly on the rotation matrices of the skeleton joints yet still achieve high realism of pose dependent shape and dynamics. A drawback of all of these approaches is that articulated models must be handcrafted for a specific object class and cannot capture general deformations.

2.2.3 Triangle deformation models

A popular approach is to build models based on the statistics of triangle deformations [75–77]. Instead of being trained to reproduce the input meshes directly, they are trained to reproduce the local deformations that produced those meshes. Unlike elastic models, these are not physically-motivated. Sumner and Popović [74] express deformation in terms of affine transformation and a displacement - the same as the deformation model used in SCAPE. Sumner et al. [14] used deformation gradients for mesh-based inverse kinematics. Hasler et al. [78] use a nonlinear representation of triangle deformations with 15 DoF which captures the relationship between pose and shape. Freifeld and Black [6] derive a 6D Lie group representation of triangle deformations with no redundant degrees of freedom. None of these approaches are rigid body motion invariant. Fröhlich and Botsch [15] additionally introduce a bending term, expressing deformations in terms of changes to geometric quantities (triangle edge lengths and the dihedral angle between adjacent triangles). Gao et al. [5] introduce a rotation-invariant mesh difference representation in which plausible deformations often form a near linear subspace. The deformations produced by all of these approaches will not in general be realisable by a connected triangle mesh. Hence, these models require a further step to solve for the mesh that best fits the desired deformations,

which might be unsatisfactory from a theoretical standpoint.

2.2.4 Riemannian shape modelling

There have been numerous attempts to cast shape modelling or statistical shape analysis in a Riemannian setting, e.g. [79–81]. To this end, one considers the space of shapes, e.g. triangular meshes, as a Riemannian manifold $\mathcal{M}$ with a metric g. Kilian et al. [80] showed how to compute geodesic paths between triangle meshes using a metric that measures changes in triangle edge lengths. Frequently, the underlying metric is based on measuring the lack of isometry, [3, 6, 11, 74–78]. To avoid irregular, isometric shape deformation an additional regularisation is required. Heeren et al. [1] take a similar approach but use the discrete shell model which includes a bending term and leads to time-discrete geodesic paths with physical meaning (they minimise the dissipation of thin shell elastic energy). The resulting shell space was subsequently further explored [82] by introducing time-discrete versions of Riemannian concepts such as the exponential and logarithmic maps and parallel transport.

2.2.5 Principal Geodesic Analysis

Let us briefly recall classical Principal Component Analysis (PCA) on $\mathbb{R}^{\mathbb{N}}$ before we consider Riemannian manifolds. For data points $s^1, \dots, s_n$, the arithmetic average is given by

$$\bar{s} = \operatorname*{arg\,min}_{s \in \mathbb{R}^N} \sum_{i=1}^{n} \|s - s_i\|^2 = \frac{1}{n} \sum_{i=1,\dots,n} s_i \tag{2.1}$$

Then, Gram's matrix is defined by $G = \frac{1}{n} D D^T \in \mathbb{R}^{n,n}$, where $D \in \mathbb{R}^{n,N}$ represents the data matrix whose ith row is given by $(s_i - \bar{s})^T \in \mathbb{R}^{1,N}$. In particular, the entries of G depend on the underlying (Euclidean) scalar product as $G_{i,j} = \frac{1}{n}\langle s_i - \bar{s}, s_j - \bar{s}\rangle$. Since G is a symmetric and positive semi-definite matrix we obtain non-negative eigenvalues $\{\lambda_j\}$ and corresponding orthonormal eigenvectors $\{w_j\}$, i.e. $Gw_j = \lambda_j w_j$ for $j = 1, \dots, n$. Finally, the principal modes of variations of the data $\{s_1 - \bar{s}, \dots, s_n - \bar{s}\}$ are obtained via $v_j = \lambda_j^{-1/2} D^T w_j \in \mathbb{R}^N$.

PCA in Euclidean space easily translates to Riemannian manifolds [79]. To this end, one considers data points $s_1, \dots, s_n$ on the manifold M and performs a classical PCA for the logarithms of the input shapes s_j with respect to their Frechet mean - the Riemannian counterpart of the arithmetic average. Thereby, the tangent vector u^j represents the geometric variations of s^j relative to the average $\bar{s}$ in an infinitesimal sense. Here, the metric $g_{\bar{s}}$ is taken into account as the scalar product on these infinitesimal shape variations. Thus, Gram's matrix is defined by $G_{ij} = \frac{1}{n} g_{\bar{s}}(u^i, u^j)$ and its spectral decomposition (v_j, λ_j) is called Principal Geodesic Analysis (PGA).

PGA models are now widely used. Fletcher et al. [79] originally propose the approach for modelling medially-defined anatomical objects. Freifeld and Black [6] use PGA to build

statistical models on their Lie group representation of triangle deformations. Tournier et al. [83] use PGA to build a statistical skeleton model. Sommer et al. [84] explore the effect of this linearisation and proposed a numerical scheme for computing exact principal geodesics. Similarly, Huckemann et al. [85] propose a scheme for exact geodesic PCA. Instead of formulating a least-squares notion of PGA, Zhang and Fletcher [86] compute principal geodesics with an explicit noise model leading to a probabilistic model. Rather than seeking geodesic curves passing through the mean (principal geodesics), Hauberg [87] proposes to compute more general principal curves in a Riemannian setting.

Besides discretised surfaces, curved shapes can be represented by parameterised functions leading to an infinite dimensional shape space. This allows modelling in a way that is invariant to re-parameterisation of the curves, essentially unifying registration and modelling. Srivastava et al. [88] achieve this using a square-root velocity representation which simplifies evaluation of an elastic metric. Kurtek et al. [89] model parameterised surfaces. Demisse et al. [90] model curved shape variation in terms of deformations, again in a re-parameterisation invariant manner.

2.2.6 3D morphable models

3D Morphable Model (3DMM) is one of the most influential works in 3D statistical shape modelling. Blanz et al. [24] directly use PCA (introduced in 2.2.5) for 3D face analysis and synthesis. Any face sample is represented as the linear combination of all training data after mean subtracted. In order to recover high-quality surfaces, Amberg et al. [91] propose a multiview stereo method based on 3DMM. Rather than using the analysis by synthesis paradigm, they focus on how to measure the distance from the observation to the hypothesis. To cope with expression changes in modelling facial data, Amberg et al. [92] assumed that the parameters governing identities and expressions are independent and therefore extended the 3D morphable model by adding expression coefficients. However, the assumption seems problematic because expression and identity are intercorrelated.

Besides global 3DMM, parts-based 3DMM like [93–95] are proposed to overcome some drawbacks of the global model. These work built multiple 3DMM and therefore improved the representational capabilities. The segmentation is done manually in most work, thus data dependent. The key step is how to handle the segments boundaries. Instead of pre-segmentation, an optimal segment is proposed in [96]. Lastly, the assembled model may not be a good reconstruction as a whole.

2.2.7 Multilinear models

Let us now consider applying multilinear model on faces. Assume we have a database of registered and rigidly aligned 3D faces of d_2 identities performance d_3 expressions, let $S \in \mathbb{R}^{N \times d_2 \times d_3}$ denote a three-dimensional array. We center each s by substracting the mean $\bar{s}$ over all

shapes. Using semantic correspondence, the different identities are associated with the second mode of S, and the different expressions are associated with the third mode. A higher order singular value decomposition (HOSVD) decomposes S into a multilinear model tensor M and orthogonal factor matrices $U_2 \in \mathbb{R}^{d_2 \times m_2}$ and $U_3 \in \mathbb{R}^{d_3 \times m_3}$ so that

$$S = M \times_2 U_2 \times_3 U_3,$$

where $M \times_n U_n$ denotes the n-th mode product of tensor M and a matrix U_n. Truncating columns of U_n could reduce the dimensionality of M, where $m_n \leq d_n$ defines the number of remaining columns of U_n. Reconstructing a 3D face f given coefficients for identity $w_2 \in \mathbb{R}^{m_2}$ and expression $w_3 \in \mathbb{R}^{m_3}$ could simply be done by

$$f = \bar{s} + M \times_2 w_2^T \times_3 w_3^T.$$

Vlasic et al. [97] use multilinear model to separately parametrise the space of geometric variations due to various attributes. Dale et al. [98] use 3D multilinear model to transfer the facial performance between two videos, without manual operation or special acquisition device. Mpiperis et al. [99] use the model for identity and expression recognition for 3D faces. Bolkart and Wuhrer [100] represent each motion sequence compactly by decoupling identity and expression variations. Cao et al. [101] generate user specific blendshapes and track the facial performance in 2D videos. Brunton et al. [102] use multiple localized multilinear models to reconstruct 3D faces from noisy and partially occluded face scans.

2.2.8 Applications

The most successful application of statistical shape model is due to Blanz and Vetter [24], in which a technique for modelling textured 3D faces is introduced. Thereafter, many face-related applications have been developed. In [103], face recognition across variations in poses and illuminations is presented by simulating the process of image formation in using a 3D face model. Statistical models also play a key role for robust human pose and shape estimation. A popular body model is SCAPE [3], which factors triangle deformations into pose and shape variations. Recent work has proposed to make SCAPE more efficient by approximating the pose dependent deformations with Linear Blend Skinning (LBS) [104, 105]. SMPL model [26] simulates variations due to pose and shape using linear functions. Additionally, Dyna model [11, 106] incorporate dynamic soft-tissue deformations.

The body and the hands have often been studied separately for several reasons. SCAPE [3] are learned from subjects making a tight fist, while more recent SMPL [26] assumes an open rigid hand. The resolution of most body scanners makes the hands too small and renders the fingers hard to recognise. Also, occlusion of the hand by the body often leads to missing data.

A triangulated mesh with LBS [73] provides a realistic model and fits image data better [107]. Despite the fixed shape, meshes are useful for computing contact points during interaction [108]. A recent hand model [109] employs smooth loop subdivision surfaces, which facilitate efficient and accurate computation of derivatives.

Apart from many significant work on learning realistic, articulated, 3D models of the human body, there are few such models of animals. Unlike human body model, real scans in a large quantity of a specific pose are not available. Zuffi et al. [110] learn a 3D animal model from a small set of 3D scans of toy animals from quadruped families.

2.2.9 Connection with functional maps

Taking advantage of functional maps [46], a novel shape difference operator is proposed in [57]. Instead of using a single number to represent the dissimilarity/similarity, the difference operator provides much more information on where two shapes are different and how they are different. Multiple applications of such as shape difference operator are shown, Among them, a fundamental one is exploration of intrinsic shape space. Shape difference operator naturally provides the differences between the base shape and all others. Then, standard PCA is applied on the vectorized shape differences. Therefore, the deviation of distortions from the average is depicted by summing up all principal components using a indicator function at a specific point. Boscaini et al. [56] showed how to reconstruct shapes from these shape difference operators, enabling shape analogy synthesis and style transfer. While the original shape differences operator captures only intrinsic distortion, Cormen et al. [111] use offset surfaces to capture extrinsic distortion.

As an alternative representation to 3D mesh, functional maps provide more flexibility and possibility to explore matrix form of correspondence. It also raise the question whether statistical modelling could be done in this space so that point-to-point correspondence is no longer needed.

CHAPTER 3

PRINCIPAL COMPONENT ANALYSIS IN SHELL SPACE

3.1 Introduction

The dimensionality of the 3D shapes is orders of magnitude greater than the number of samples in the training set. Typical meshes may contain tens of thousands of vertices. In contrast, the number of training samples that can feasibly be collected is typically only in the tens or hundreds. Hence, the underlying shape space is sampled very sparsely.

In such a scenario, the quality of the model is dependent on the validity of the assumed or learnt structure of the manifold on which the data is assumed to lie. For example, PCA assumes that the input data lies on (or can be well approximated by) a hyper-planar manifold, the axes of which are those that capture maximal variance. This makes it optimal with respect to compactness in Euclidean space but a poor choice when the data contains highly nonlinear variations.

In this chapter, we use a physically-motivated, nonlinear model of surfaces, thin shells, to represent 3D deformable objects. Geodesics in the space of shells have already shown great promise in realistically interpolating and extrapolating between sparse samples of shapes undergoing complex deformations [15, 69, 82]. In other words, shell space potentially provides a useful constraint in modelling the nonlinear variability in a sparsely sampled set of shapes. This motivates our idea of performing statistical shape analysis in shell space.

We consider a linear space of nodal displacements and not the tangent space in a Riemannian setup of infinitesimal displacements. This is computationally much more efficient, as no higher resolution of the paths from the average to each input shape is required. We define a notion of covariance based on the Hessian of an elastic energy term. In analogy to PCA, we extract principal components based on an eigendecomposition of the resulting Gram matrix. The resulting principal components are able to capture nonlinear articulations and complex deformations.

We provide results on human face and body data and evaluate the resulting models in terms of compactness, generalisation and specificity.

The remainder of this chapter will be organised as follows: a physical material called "thin shell" and its deformations will be introduced in Sec. 3.2. Then, based on shell deformations, *averaging* and *covariance* operators are described in Sec. 3.3 before PCA in shell space is developed. The implementation aspect of this method is summarised in section Sec. 3.4. In the end, we demonstrate the experimental results applying Shell PCA in Sec. 3.5, with comparison to traditional Euclidean PCA.

3.2 Shells and shell deformations

A thin shell s^δ is a physical material with tiny but positive thickness δ. Mathematically, this shell is represented by a smooth surface s embedded in $\mathbb{R}^3$ which is thought of as the middle layer of the physical material, i.e.

$$s^\delta = \{p + z\, n(p) \,|\, p \in s\,, z \in (-\delta/2, \delta/2)\}\,,$$

where $n : s \to S^2$ denotes the unit normal field.

In the following we will consider a reference material $\bar{s}^\delta \subset \mathbb{R}^3$ which is in a stress-free state and an elastic deformation ϕ^δ thereof. All quantities corresponding to $\bar{s}^\delta$ and its middle layer $\bar{s}$, respectively, will be labelled with a bar, e.g. $\bar{g}$ denotes the first fundamental form of $\bar{s}$. For the sake of clarify, we omit the thickness δ in the following without causing confusion.

The corresponding elastic deformation energy is given by

$$\mathcal{W}^\delta[\phi^\delta] = \int_{\bar{s}^\delta} W_{\text{mem}}(D\phi^\delta)\,\mathrm{d}x\,, \tag{3.1}$$

where $D\phi^\delta \in \mathbb{R}^{3,3}$ and W_{mem} denotes some elastic energy density.

LeDret and Raoult [112] have shown in the context of Γ-convergence that to leading order, the energy Eq. (3.1) scales linearly in the thickness parameter δ and after rescaling with $\frac{1}{\delta}$ is given by the *membrane* energy

$$\mathcal{W}_{\text{mem}}[\bar{s}, \phi] = \int_{\bar{s}} W_{\text{mem}}(\mathcal{G}[\phi])\,\mathrm{d}a\,, \tag{3.2}$$

where $\phi : \bar{s} \to \mathbb{R}^3$ is the deformation of the middle layer and $\mathrm{d}a = \sqrt{\det \bar{g}}\,\mathrm{d}t$ denotes the area element. The Cauchy-Green strain tensor $\mathcal{G}[\phi] \in \mathbb{R}^{2,2}$ is given by

$$\mathcal{G}[\phi] = \bar{g}^{-1} g_\phi\,, \tag{3.3}$$

where g_ϕ is the intrinsic first fundamental form on the deformed shell $\phi(s)$. The membrane

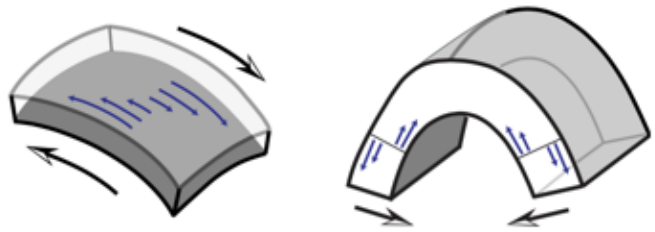


Figure 3.1: Membrane (left) and bending (right) distortion of Shell deformation. (Figure courtesy of Heeren et al. [1])

energy density in Eq. (3.2) can be chosen e.g. as

$$W_{\text{mem}}(A) = \frac{\mu}{2}\operatorname{tr} A + \frac{\lambda}{4}\det A - (\frac{\mu}{2} + \frac{\lambda}{4})\log\det A - \mu - \frac{\lambda}{4}\,.$$

Here the trace accounts for local length changes while the determinant accounts for local change of area.

Friesecke et al. [113] demonstrated that for isometric deformations ϕ (for which $\mathcal{W}_{\text{mem}}[\bar{s}, \phi] = 0$) the leading order term of Eq. (3.1) is *cubic* in the thickness δ and after rescaling with $\frac{1}{\delta^3}$ is given by an energy term that solely depends on the so-called relative shape operator. This *bending* energy is supposed to account for out of plane bending and changes in curvature. One particular choice is the *Willmore* energy

$$\mathcal{W}_{\text{bend}}[\bar{s}, \phi] = \int_{\bar{s}} |\bar{H} - H \circ \phi|^2 \,\mathrm{d}a\,, \tag{3.4}$$

which measures changes in the mean curvature H. Note that H can be regarded as the trace of the matrix-valued relative shape operator.

As shown in Fig. 3.1, the corresponding elastic energy could be divided into two kinds of deformation: membrane deformation and bending energy. Approximating thin shell energy by summing up these two types of deformation is widely used in the computer graphics community [1, 82] and we follow this representation and this could be written as

$$\mathcal{W}_{\bar{s}}[\phi] = \mathcal{W}_{\text{mem}}[\bar{s}, \phi] + \gamma\, \mathcal{W}_{\text{bend}}[\bar{s}, \phi] \tag{3.5}$$

where the physical parameter γ is a bending weight which can be viewed as a parameter that balances between membrane energy and bending energy.

For two shells $\bar{s}, s$ we define

$$\mathcal{W}[\bar{s}, s] = \min_{\phi:\phi(\bar{s})=s} \mathcal{W}_{\bar{s}}[\phi]\,. \tag{3.6}$$

We might consider this as an approximation of the squared Riemannian distance in the shell

space although it is not symmetric. However, for infinitesimal small deformations the Hessian of $\mathcal{W}$ allows us to retrieve a symmetric, positive-definite Riemannian metric on the space of shell [82].

3.2.1 Discrete shell and its deformations

The above description of shell deformation is based on a "smooth shell". In practice, we discretise a smooth shell s by a triangular mesh S. Furthermore, we assume a fixed connectivity constraint that means there is a one-to-one mapping (dense correspondence) between all vertices and all faces of two meshes as e.g. in [80]. Hence we will represent a triangular mesh S by the vector of its vertex positions $\mathbf{x} \in \mathbb{R}^{3n}$, where n is the number of vertices in the mesh, and each vertex has three degrees of freedom in 3D space, i.e. x, y, z.

We can think of each triangle $T \subset \mathbb{R}^3$ being parametrised over the unit triangle $\omega \subset \mathbb{R}^2$ consisting of the nodes $(0,0)$, $(1,0)$ and $(0,1)$. If $q_0, q_1, q_2 \in \mathbb{R}^3$ are the nodes of T we consider the local linear parametrisation $X_T : (t_1, t_2) \mapsto t_1 q_1 + t_2 q_2 + (1 - t_1 - t_2) q_0$ with $0 \leq t_1, t_2 \leq 1$. Hence we have

$$DX_T = [\partial_{t_1} X_T \,|\, \partial_{t_2} X_T] = [q_1 - q_0 \,|\, q_2 - q_0] \in \mathbb{R}^{3,2}$$

and can deduce an element-wise constant discrete first fundamental form (to allow surface properties such as length and area) via $G_T = (DX_T)^T DX_T \in \mathbb{R}^{2,2}$. Due to the dense correspondence we can define the discretisation of the distortion tensor elementwise via

$$\mathcal{G}[\Phi]_{\bar{T}} = \bar{G}_{\bar{T}}^{-1} G_T.$$

Hence the discrete *membrane* energy is

$$\mathbf{W}_{\text{mem}}[\bar{\mathbf{x}}, \mathbf{x}] = \sum_{\bar{T} \in \bar{s}} |\bar{T}| \, W_{\text{mem}}(\mathcal{G}[\Phi]_{\bar{T}}) \,, \tag{3.7}$$

where the membrane energy density W_{mem} characterises the edge length and area changes. $|\bar{T}|$ denotes the area of $\bar{T}$. As a result, discrete membrane energy amounts to the edge length and area changes considering all the triangles of the reference mesh $\bar{\mathrm{S}}$.

Next, we consider discrete *bending* energy. Concretely, we make use of the discrete shells energy proposed in [69]:

$$\mathbf{W}_{\text{bend}}[\bar{\mathbf{x}}, \mathbf{x}] = \sum_{\bar{e} \in \bar{s}} \frac{(\theta_{\bar{e}} - \theta_e)^2 |\bar{e}|^2}{D_{\bar{e}}} \,, \tag{3.8}$$

where the sum is over all undeformed edges $\bar{e} \in \bar{\mathrm{S}}$. If T and T' share one edge e we have $D_e = \frac{1}{3}(|T| + |T'|)$ and θ_e denote the complement of dihedral angle at the undeformed and deformed edge, respectively, illustrated in Figure 3.2. Analogously to Eq. (3.5), the discrete

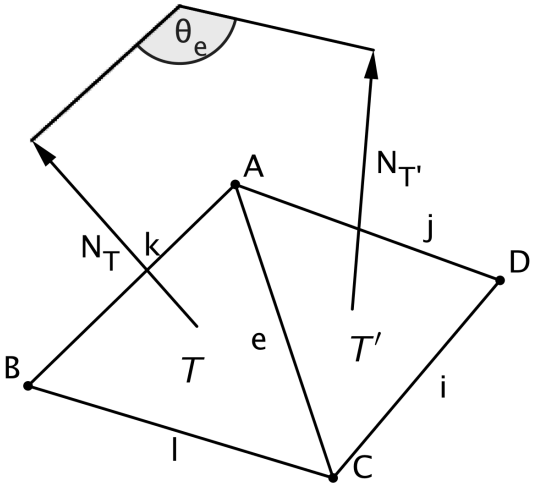


Figure 3.2: Illustration of geometric quantities used in discrete bending energy computation

deformation energy is given as the sum of Eq. (3.7) and Eq. (3.8):

$$\mathbf{W}[\bar{\mathbf{x}}, \mathbf{x}] = \mathbf{W}_{\text{mem}}[\bar{\mathbf{x}}, \mathbf{x}] + \gamma\, \mathbf{W}_{\text{bend}}[\bar{\mathbf{x}}, \mathbf{x}] \tag{3.9}$$

Note that due to the dense correspondence there is a natural element-wise linear deformation between two meshes $\bar{\mathbf{x}}$ and $\mathbf{x}$. Hence, different from Eq. (3.6), we do not need to optimise for deformations in Eq. (3.9).

Briefly, discrete shell energy is effectively measuring changes in triangle **edge length, area and dihedral angle** between undeformed shell and deformed one, and potentially is a well-suited metric candidate for deformation modelling. Hence, this direction will be explored in the following sections.

3.3 Principal component analysis in shell space

PCA is among statistical tools to modelling 3D meshes. A principal component analysis (PCA) relies on the notions of *averaging* and *covariance*. It then uses an eigen-decomposition of the covariance matrix to extract linear principal components. Following [114], we will now introduce for a given set of input meshes (i) a general notion of an average (depending on a distance measure d) and (ii) a covariance operator as a generalization of a covariance matrix (depending on an inner product g). We will introduce two particular choices for d and g, respectively, namely the standard Euclidean distance/metric and a physically-based distance/metric induced by the shell deformation energy. The corresponding PCA will be referred to as Euclidean PCA in the former and as Shell PCA in the latter setup.

Averaging. Consider a given set of input data $\mathrm{S}_1, \ldots, \mathrm{S}_m$ which we now consider as triangular meshes that are in dense correspondence. Hence we can represent each mesh S_i by its vector

of vertex positions $\mathbf{x}_i \in \mathbb{R}^{3n}$. For a given (squared) distance measure $d^2 : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \to \mathbb{R}$, the group average $\hat{\mathbf{x}} \in \mathbb{R}^{3n}$ is given by the Fréchet mean (we admit there exist many options for the mean as shown in [81], Fréchet mean is chosen here for its simplicity).

$$\hat{\mathbf{x}} := \arg\min_{\mathbf{x}} \sum_{i=1}^{m} d^2[\mathbf{x}_i, \mathbf{x}] \,. \tag{3.10}$$

In the Euclidean setup we have $d^2[\mathbf{x}_i, \mathbf{x}] = \|\mathbf{x}_i - \mathbf{x}\|^2$ and hence

$$\hat{\mathbf{x}}_{\text{euc}} = \tfrac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \,.$$

However, in the Shell PCA setup we have $d^2[\mathbf{x}_i, \mathbf{x}] = \mathbf{W}[\mathbf{x}_i, \mathbf{x}]$ and Eq. (3.10) becomes a nonlinear optimization problem and the average $\hat{\mathbf{x}}_{\text{shell}}$ has to fulfil the necessary condition

$$F[\hat{\mathbf{x}}_{\text{shell}}] := \sum_{i=1}^{m} \partial_2 \mathbf{W}[\mathbf{x}_i, \hat{\mathbf{x}}_{\text{shell}}] = 0 \,. \tag{3.11}$$

Here ∂_2 denotes differentiation with respect to the second argument of $\mathbf{W}$.

Covariance operator. Covariance operator is a generalisation of covariance matrix. Inherently a PCA is defined on a linear space. As a result, we will consider the linear space of vertex displacements $\mathbf{u}_i = \mathbf{x}_i - \hat{\mathbf{x}} \in \mathbb{R}^{3n}$, $i = 1, \ldots, m$, from the average $\hat{\mathbf{x}}$. Given an inner product g on $\mathbb{R}^{3n}$ we define the covariance operator

$$\mathbf{Cov}\, \mathbf{v} = \frac{1}{m} \sum_{i=1}^{m} g(\mathbf{v}, \mathbf{u}_i)\mathbf{u}_i \tag{3.12}$$

and an associated matrix $\mathbf{C} = (\mathbf{C}_{ij})_{ij} \in \mathbb{R}^{m,m}$ via

$$\mathbf{C}_{ij} := g(\mathbf{u}_i, \mathbf{u}_j) \,. \tag{3.13}$$

Obviously, C is symmetric and positive semi-definite, that means there exist a spectral decomposition so that

$$\mathbf{C} = O\Lambda O^T \,, \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m) \,,$$

where $\lambda_1 \geq \ldots \geq \lambda_m \geq 0$ are eigenvalues of $\mathbf{C}$ and O is an orthogonal matrix, i.e. $OO^T = O^T O = \operatorname{id}_m$. Then, we define $\mathbf{w}_1, \ldots, \mathbf{w}_m$ via

$$\mathbf{w}_k := \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{m} O_{ik}\mathbf{u}_i \tag{3.14}$$

if $\lambda_k > 0$ and $\mathbf{w}_k = 0$ else. A straight forward calculation reveals that

$$\mathbf{Cov}\,\mathbf{w}_k = \frac{\lambda_k}{m}\mathbf{w}_k$$

and $g(\mathbf{w}_k, \mathbf{w}_l) = \delta_{kl}$, i.e. $\mathbf{w}_1, \ldots, \mathbf{w}_m$ are in fact eigenvectors of **Cov** . Formally, we can extend $\mathbf{w}_1, \ldots, \mathbf{w}_m$ to an orthonormal basis of $\mathbb{R}^{3n}$ with $\mathbf{Cov}\,\mathbf{w}_k = 0$ for $k > m$.

Remark 1: Usually eigenvectors of **Cov** are found by a spectral decomposition of the $(3n)$-by-$(3n)$ covariance matrix. However, as in most applications $m \ll 3n$ it is more efficient to decompose $C \in \mathbb{R}^{m,m}$ as defined in Eq. (3.13) and obtain eigenvectors via Eq. (3.14).

Remark 2: Due to the rigid body motion invariance the representation of a discrete shell S by its vertex positions $\mathbf{x}$ is not unique. In fact, S is represented by an equivalence class of vertex position vectors induced by rigid body motions. This issue becomes crucial when defining vertex displacements $\mathbf{u} = \mathbf{x} - \hat{\mathbf{x}}$, as we can construct an arbitrary large displacement by a simple translation. However, this obstacle is overcome by taking $\mathbf{x}$ such that $\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \leq \|\mathbf{y} - \hat{\mathbf{x}}\|^2$ for all $\mathbf{y}$ in the equivalence class.

Computing principal modes. For a data set $\mathbf{u}_1, \ldots, \mathbf{u}_m$ the first component $\mathbf{v}_{(1)}$ of a principal component analysis (PCA) is defined as

$$\begin{aligned}\mathbf{v}_{(1)} &= \arg\max_{\|\mathbf{v}\|=1} \sum_{i=1}^{m} g(\mathbf{v}, \mathbf{u}_i)^2 \\ &= \arg\max_{\|\mathbf{v}\|=1} g(\mathbf{Cov}\,\mathbf{v}, \mathbf{v}) \end{aligned} \tag{3.15}$$

where we used the definition Eq. (3.12) of **Cov** in the second equality. If we now write $\mathbf{v} = \sum_k \alpha_k \mathbf{w}_k$, i.e. represent $\mathbf{v}$ in the orthonormal basis as defined in Eq. (3.14), we get $g(\mathbf{Cov}\,\mathbf{v}, \mathbf{v}) = \sum_k \alpha_k^2 \lambda_k$. Hence Eq. (3.15) is equivalent to solving

$$\bar{\alpha} = \arg\max_{\|\alpha\|=1} \sum_{k=1}^{m} \alpha_k^2 \lambda_k \,.$$

As λ_1 is the largest eigenvalue we have $\bar{\alpha} = (1, 0, 0, \ldots)$ and hence $\mathbf{v}_{(1)} = \mathbf{w}_1$. Similarly we obtain further components $\mathbf{v}_{(k)}$ as $\mathbf{v}_{(k)} = \mathbf{w}_k$ for $k = 2, \ldots, m$. Hence the principal components are given by the eigendisplacements of **Cov** as defined in Eq. (3.14).

Algorithm 1 Computation of Shell PCA principal modes

Input: $s_1, \ldots, s_m \in \mathcal{M}$
Output: elastic average $\hat{s}$, principal modes $\mathbf{w_1}, \ldots, \mathbf{w_m}$
// initialise first input s_1 as elastic average
$\hat{s} = s_1$
// compute elastic average
solve for elastic average $\hat{s}$ using (Eq. 3.10)
// rigid body alignment
align each shape to average $\hat{s}$
// compute vertex displacement $\mathbf{u}$
for $i := 1$ to m **do**
    $\mathbf{u}_i = s_i - \hat{s}$
end for
// assemble Gram's matrix
compute $\mathbf{G}$ using (Eq. 3.13)
// compute principal modes
calculate principal modes $\mathbf{w}$ using (Eq. 3.14)

Choice of metric. We have not specified an inner product g on the space of discrete shells yet. For the Euclidean setup we can define g as the standard scalar product on $\mathbb{R}^{3n}$, i.e.

$$g_{\text{euc}}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}\,.$$

It has been shown in [82] that for a deformation energy as defined in Eq. (3.9) the bilinear form

$$g_{\text{shell}}(\mathbf{u}, \mathbf{v}) = \frac{1}{2}\,\mathbf{u}^T (\mathbf{Hess}\,\mathbf{W})[\hat{\mathbf{x}}, \hat{\mathbf{x}}]\mathbf{v} \tag{3.16}$$

in fact defines a metric on the space of discrete shells modulo rigid body motions. Here $\mathbf{Hess}\,\mathbf{W} = \partial_2^2 \mathbf{W} \in \mathbb{R}^{3n,3n}$ denotes the Hessian matrix w.r.t. the second argument of $\mathbf{W}$ which is a positive semi-definite symmetric matrix.

Visualising of principal modes. A PCA model amounts to an average shape $\hat{\mathrm{S}}$ represented by its vertex positions $\hat{\mathbf{x}} \in \mathbb{R}^{3n}$ and its eigendisplacements $\mathbf{w}_k$ obtained by the PCA. A simple linear visualization of these dominant modes is to compute meshes via $\hat{\mathbf{x}} + t\,\mathbf{w}_k$, where $t \in [-T, T]$.

However, a more reasonable way to express pure nonlinear variations is to use the nonlinear shooting method via the time-discrete exponential map $\mathrm{EXP}_{\mathbf{x}}$ as proposed in [82]. On a smooth manifold $\mathcal{M}$ the exponential map $\exp_{\mathbf{x}}$ maps a tangent vector $\mathbf{v}$ at some point $\mathbf{x}$ to the endpoint $\mathbf{x}(1)$ of the geodesic $t \mapsto \mathbf{x}(t)$ with $\mathbf{x}(0) = \mathbf{x}$ and $\dot{\mathbf{x}}(0) = \mathbf{v}$. Here, we use the exponential map on shell space as a natural extrapolation of shell variations and pick up the corresponding time-discrete definition from [82]. In fact, for the visualizing of (nonlinear) principal modes of

variation we consider the elastic average $\hat{\mathbf{x}}$ as start point $\mathbf{x}$ and a (possibly scaled) mode $\alpha\mathbf{w}_k$, $\alpha \in \mathbb{R}$, as the initial velocity v.

Projection and reconstruction. In an analogous fashion to Euclidean PCA, Shell PCA can also be used for reconstructing shapes from a set of PCA coefficients. Given the elastic average $\hat{\mathbf{x}}$, the eigenvectors $\mathbf{w}_k$ and some (possibly unseen) shape $\mathbf{x} \in \mathbb{R}^{3n}$, we first compute the vertex displacement $\mathbf{u} = \mathbf{x} - \hat{\mathbf{x}}$. We then project $\mathbf{u}$ onto the Shell PCA space $\mathcal{W}_m$, where $\mathcal{W}_m$ is the linear subspace spanned the principal modes $\mathbf{w}_k, k = 1, \ldots, m$. The projection is given by

$$P_{\mathcal{W}_m} : \mathbf{u} \to P_{\mathcal{W}_m}\mathbf{u} := \sum_{k=1}^{m} g_{\text{shell}}(\mathbf{w}_k, \mathbf{u})\mathbf{w}_k \,,$$

where g_{shell} is the Shell Eq. (3.16). Finally we are able to get the reconstruction $\tilde{\mathbf{x}}$ via nonlinear shooting:

$$\tilde{\mathbf{x}} := \mathrm{EXP}_{\hat{\mathbf{x}}}(P_{\mathcal{W}_m}\mathbf{u}) \,.$$

Elastic vs. Riemannian PCA We consider a linear space of (possibly large) nodal displacements and *not* the tangent space in a Riemannian setup of infinitesimal displacements. This is computationally more efficient (no higher resolution of geodesic paths from the average $\hat{\mathbf{x}}$ to each input shape $\mathbf{x}_i$ is required) and sufficient also for large displacements due to the involved nonlinear elastic deformation energy and its invariance w.r.t. rigid body motions. However, the input displacements $\mathbf{u}_i = \mathbf{x}_i - \hat{\mathbf{x}}$ as well as the resulting principal modes $\mathbf{w}_k$ can be considered as approximate tangent vectors in the Riemannian tangent space at the average shape $\hat{\mathbf{x}}$ whereas this approximation is only valid in a neighbourhood of $\hat{\mathbf{x}}$. Furthermore, this motivates the nonlinear shooting of principal modes using the discrete exponential map.

3.4 Implementation and optimization

The implemention is realized as an extension of the open source C++ library QuocMesh[1]. Computationally, the most demanding part is evaluating the shell mean $\hat{\mathbf{x}}_{\text{shell}} \in \mathbb{R}^{3n}$ in Eq. (3.11). This is done by means of Newton's method, i.e. for $F[.]$ as defined in Eq. (3.11) and an initial guess $\mathbf{x}_0$ we compute iteratively

$$DF[\mathbf{x}_k]\,\mathbf{D}_k = -F[\mathbf{x}_k] \,, \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k \mathbf{D}_k \,,$$

until $\|DF[\mathbf{x}_k]\| < \epsilon$ for some $k < K_{\max}$ and set $\hat{\mathbf{x}}_{\text{shell}} = \mathbf{x}_k$. The stepsize τ_k is determined by Armijo's backtracking line search [115]. Note that each iteration step requires (i) an evaluation of F and DF, i.e. the assembling and addition of m Hessian matrices $\partial_2^2\mathbf{W}[.,.] \in \mathbb{R}^{3n,3n}$, and (ii) solving a linear system in $3n$ dimensions. To improve the robustness and efficiency of the

[1] http://numod.ins.uni-bonn.de/software/quocmesh/index.html

Figure 3.3: Human bodies data with large nonlinear, articulated deformations from FAUST dataset [2]. These 5 samples are used in our experiment for evaluating the principal modes in section Sec. 3.5.1

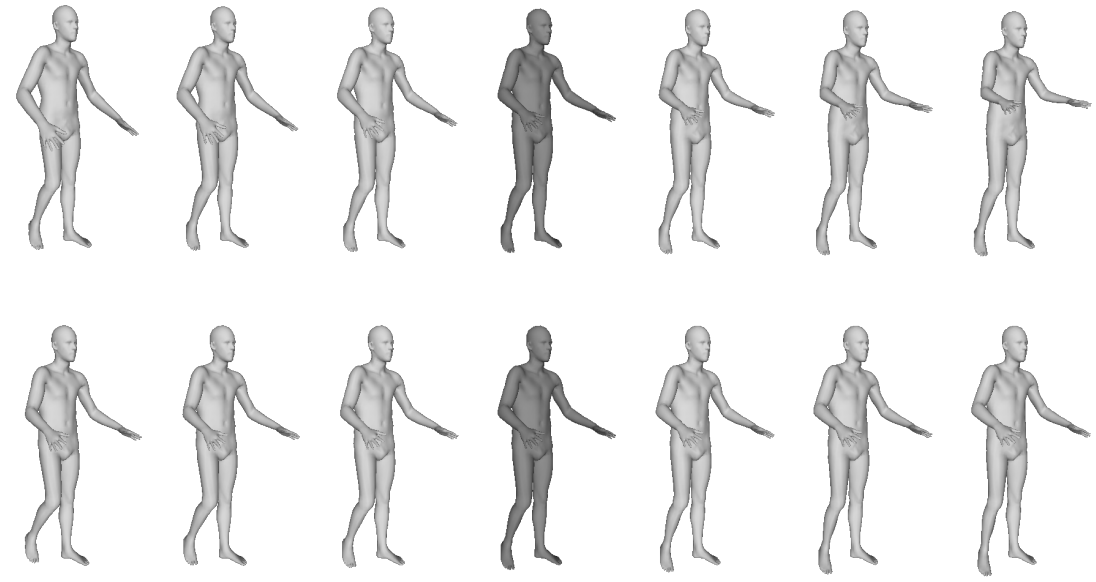
optimization we make us of an hierarchical scheme based on progressive meshes [116, 117] as it was used in [80]. The computation of the nonlinear mean for the input data shown in Fig. 3.3 with $m = 5$ and $n \approx 6000$ takes 5 minutes on an Intel Core 3.40GHz. However, the running time can be improved substantially by picking up the two-level hierarchical method proposed in [15]. Note that the spectral decomposition of Eq. (3.13) by means of a standard QR algorithm is fast as m is usually small.

3.5 Experiments

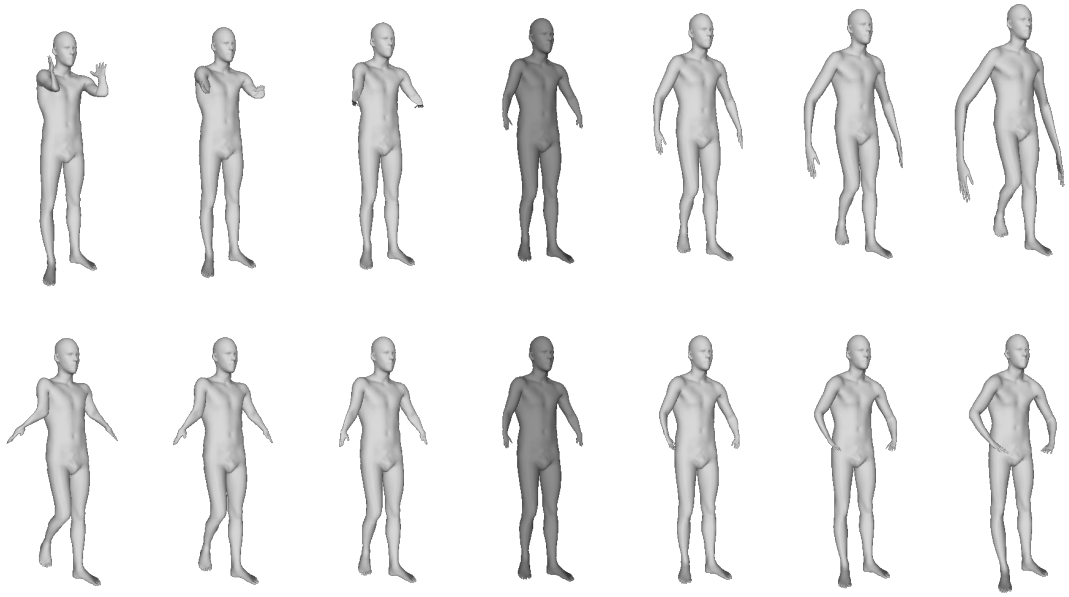
In this section we provide an experimental evaluation of our model. We begin with a qualitative evaluation of the Shell PCA model by visualising the principal components. Next we compare reconstructions using Euclidean and Shell PCA . Finally, we provide a quantitative comparison in terms of compactness, generalisation and specificity. We use two datasets in our experiments. The first contains scans of human bodies drawn from the FAUST dataset [2]. The 5 training shapes are shown in Fig. 3.3. The meshes are watertight, genus zero and we apply groupwise simplification to reduce their resolution to 6,000 vertices. For this data, Euclidean PCA fails to obtain a meaningful average, let alone principal components due to the articulated motion. The second is the B3D(AC)^2 [118] dataset containing facial motion sequences in dense correspondence. From this dataset, we extract a subset consisting of 40 expressions of a single subject. The meshes are genus three (holes for the mouth and eyes) with a boundary. Again, we apply groupwise simplification to reduce the mesh resolution to 3,000 vertices. Note that in both cases, the training data is extremely sparse and that there are large, nonlinear deformations between shapes.

3.5.1 Qualitative evaluation of PCA

In Fig. 3.4, we show the first two principal components for the body data. The elastic average for Shell PCA and the linear average for Euclidean PCA are depicted in the middle column (shapes in dark grey), respectively. In each row, we show a geodesic path traversing from the average



(a) Shell PCA modes (elastic average in dark grey)



(b) Euclidean PCA modes (linear average in dark grey)

Figure 3.4: Top 2 modes of body data with (a) Shell PCA and (b) Euclidean PCA. The mean shape in dark color is placed in the middle, and negative and positive shooting (linear combination for Euclidean PCA) results are shown on the left and the right, respectively.

in each direction along the principal component. Note that the shell PCA modes successfully capture the nonlinear, articulated motion. The first mode appears to capture the raising and lowering of the arms and the second the bending of the leg, while Euclidean PCA fails to capture meaningful deformations and leads to degenerated surfaces.

To emphasise the nonlinear nature of the Shell PCA modes, we show the vertex trajectories for the first principal component in Fig. 3.5. This is done by generating a sequence of shapes by

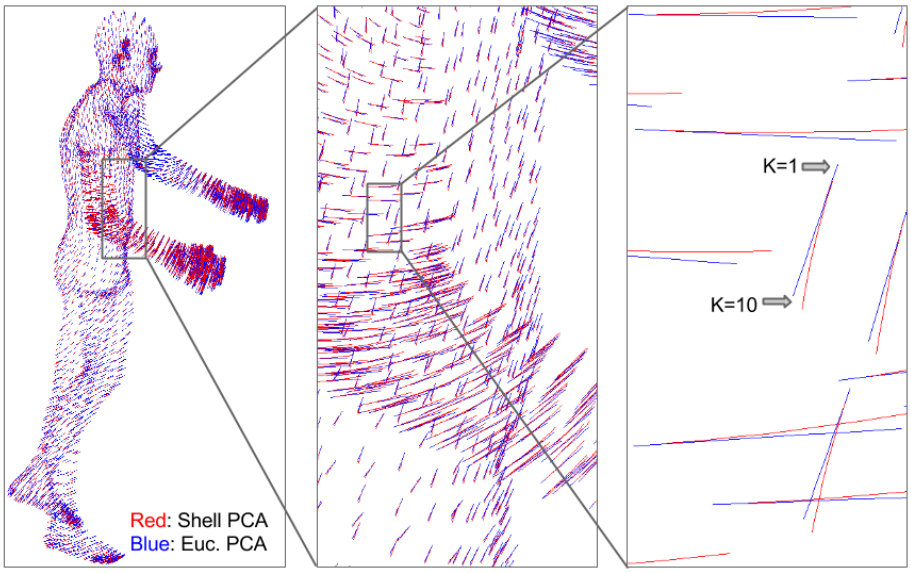


Figure 3.5: Vertex trajectories of Shell PCA and Euclidean PCA. Note that the vertices follow a curve with the Shell PCA (red) and a straight line with the Euclidean PCA (blue).

nonlinear shooting and plotting the resulting trajectories in red. For comparison, the Euclidean PCA trajectories are shown in blue. Shell PCA clearly leads to nonlinear trajectories.

In Fig. 3.6, we show the first three principal components for the facial expression data. We show Euclidean PCA in the first row, Shell PCA with linear combination in the second row, and Shell PCA with nonlinear shooting in the third row. Note that, while both Euclidean and Shell PCA capture similar characteristics in their principal three modes, Shell PCA with nonlinear shooting prevents the surface from folding over itself and retains a more plausible face shape (for example, mode 1 of Euclidean PCA appears to correspond approximately to mode 3 of Shell PCA but Shell PCA preserves a more plausible chin shape in the positive direction).

3.5.2 Shape reconstruction

In Fig. 3.7 we show the result of reconstructing a face using an increasing number of model dimensions. The results in the top row are for Euclidean PCA and in the bottom row for Shell PCA. While PCA is optimal in terms of minimising Euclidean distance, it is clear that there is a perceptual improvement in the reconstruction results using Shell PCA. Using only 5 dimensions (column 2), Shell PCA has successfully reconstructed the strong elastic deformation of the smile while the Euclidean PCA reconstruction does not successfully convey the smiling expression.

We also show some results of human pose reconstruction using SCAPE dataset. The failed case in reconstructing human poses further supports our claim that displacement is not meaningful to be used for large deformations.

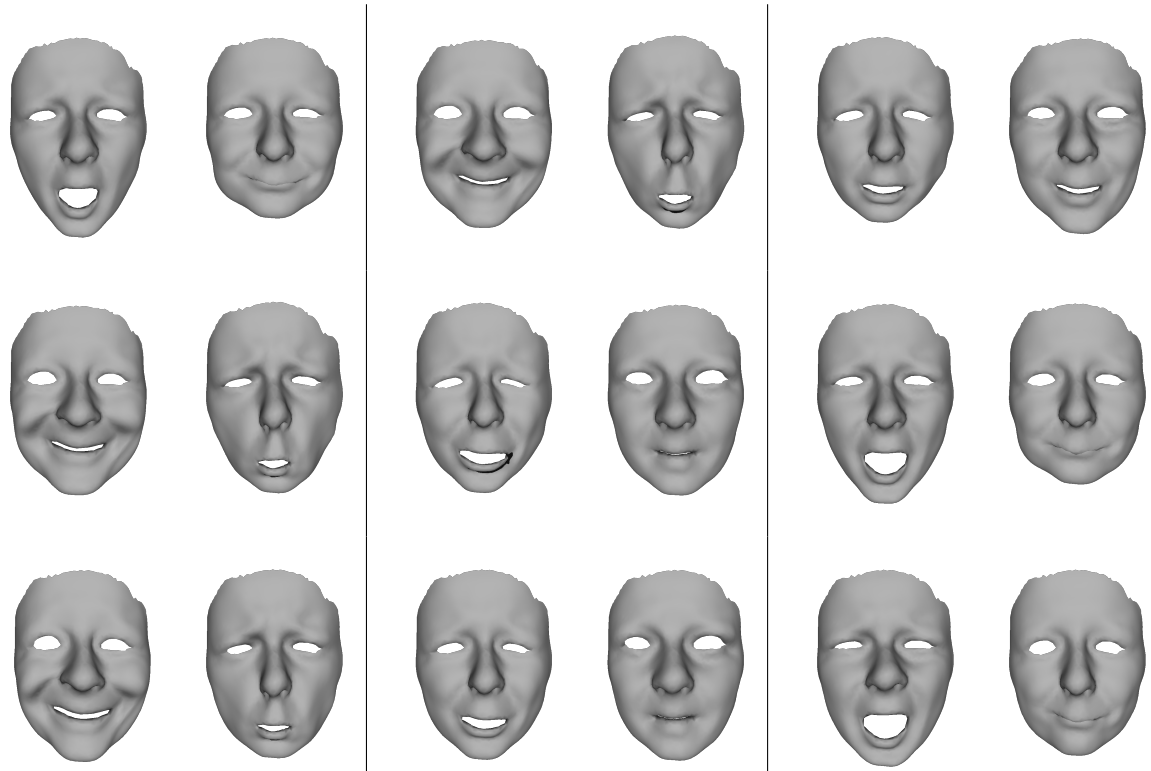
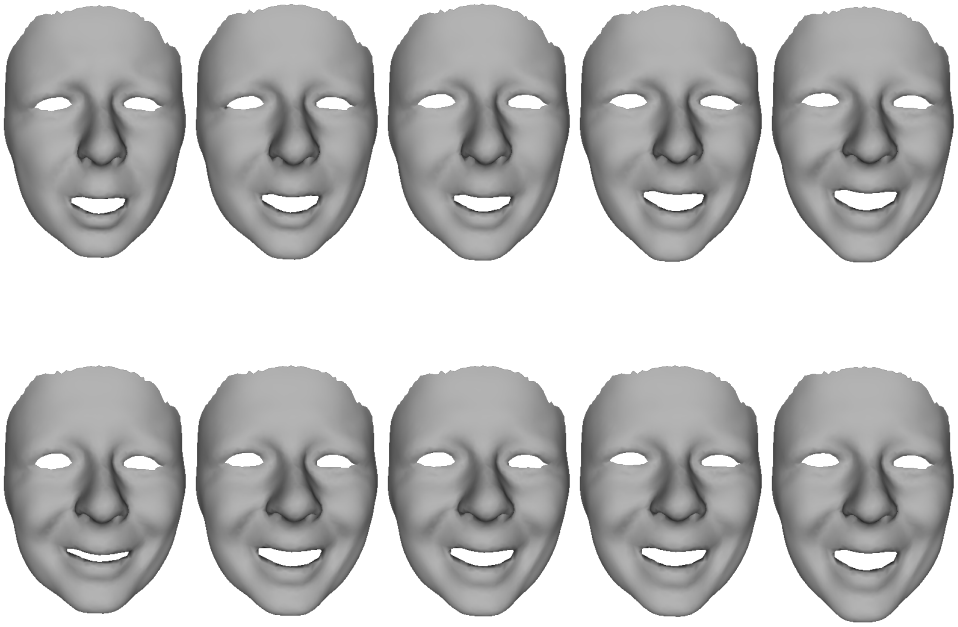


Figure 3.6: Top 3 modes of the face expression data. (Top: Euclidean PCA; middle, Shell PCA with linear combination; bottom: Shell PCA with nonlinear shooting. Col. 1 and 2 show mode 1, col. 3 and 4 show mode 2, col. 5 and 6 show mode 3.)

3.5.3 Quantitative evaluation

We now provide a quantitative comparison between a Euclidean PCA and Shell PCA model using the facial expression data.

Compactness measures how efficiently a model captures the variability in the training data. Specifically, it is the cumulative variance captured by the top K principal components as a proportion of the total variance within the training data. Hence, we define the compactness as $\alpha(K) = \frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$. We show compactness as a function of K for the expression dataset for Shell and Euclidean PCA in Fig. 3.9. Euclidean PCA is optimal in the sense of least squares, i.e. measuring in the (squared) Euclidean norm. Shell PCA is optimal in the sense of (squared) Riemannian distance in shell space. It is clear that Euclidean PCA is superior in terms of compactness, implying that Euclidean variance in the expression data is more easily captured than elastic variance. However, we should be cautious in how we interpret these plots since the variances in the two models are computed under different distance measures. Generalisation evaluates the ability of a shape space to represent unseen examples of the class. Given a set of m training shapes, generalisation is measured using leave-one-out cross reconstruction of training samples, i.e. the model is learned using $m-1$ samples and the model is then fitted to the excluded sample. The fitting error is measured using two distances: the mean vertex-to-vertex



dim=1 dim=5 dim=10 dim=39 GT

Figure 3.7: Reconstruction results of using increasing number of parameters. (Top: Euclidean PCA reconstruction results, bottom: Shell PCA reconstruction results. From left to right: dimension = (1,5,10,39), and the last column for ground-truth.)

Euclidean distance, and the elastic deformation energy. Generalisation is then reported as mean fitting error averaged over all trials and is a function of the number of model parameters. It is expected that the mean error decreases until convergence as the number of shape parameters increases. In the following experiment, the training data consists of $m = 40$ shapes.

The generalisation error results are shown in Fig. 3.10a (measured in terms of Euclidean distance) and Fig. 3.10b (measured in terms of elastic deformation energy). The errors are computed between the reconstructed and ground truth shape.

It is unsurprising that Shell PCA outperforms Euclidean PCA in terms of minimising elastic deformation energy. However, perhaps more surprising it that Shell PCA seems to perform better in terms of Euclidean distance once more than 7 model dimensions are used. This implies that the Shell principal components are more successfully modelling the underlying shape space, enabling better reconstructions of unseen data.

Specificity measures how well a model is able to generate instances that are similar to real data. Firstly, a set of instances are sampled from the learned shape space. Then, for each sample, the error to closest real sample is computed and an average taken over all samples. It is expected that the mean distance increases until convergence with increasing number of parameters. This is because increasing numbers of model dimensions gives the model more flexibility to create more variable shapes, increasing the likelihood that they lie a long way from real samples. In

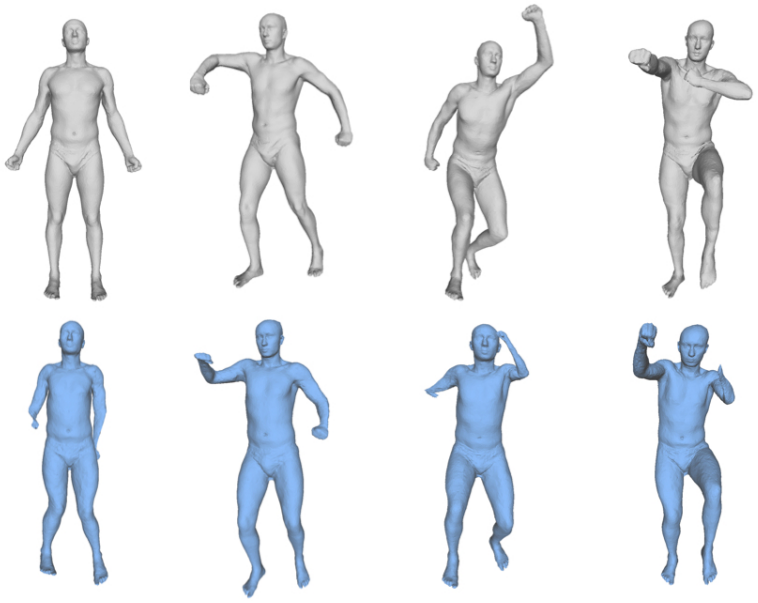


Figure 3.8: Failed cases in reconstructing human poses of SCAPE [3]. Top: ground truth shape; bottom: reconstructions.

practical applications, generalisation ability and specificity are traded off against each other.

To measure specificity, we consider a PCA model as a probabilistic model by assuming that the data forms a Gaussian cloud in either Euclidean or shell space. Hence, we consider the shape parameter vector to be drawn from a multivariate normal distribution zero mean and standard deviations given by the eigenvalues. We generate 10,000 random samples for both the Euclidean and Shell PCA models. We show the specificity error as a function of the number of parameters in terms of Euclidean distance (Fig. 3.11a) and elastic deformation energy (Fig. 3.11b). In both cases, Shell PCA converges more slowly but gives better specificity for > 20 dimensions. Again, this suggests that the shell principal components are a better model of the underlying shape space.

3.6 Summary and limitations

In this chapter, we presented a nonlinear (elastic) deformation based covariance analysis which is a variant of Principal Components Analysis to operate in shell space, providing a hybrid between physical and statistical modelling of shape variation. Principal components are obtained via an eigendecomposition of a covariance matrix arising from an inner product based on the Hessian of an elastic energy. We have shown that such a model is better able to capture the nonlinear variations present in articulated body pose data and facial expression data with complex deformations.

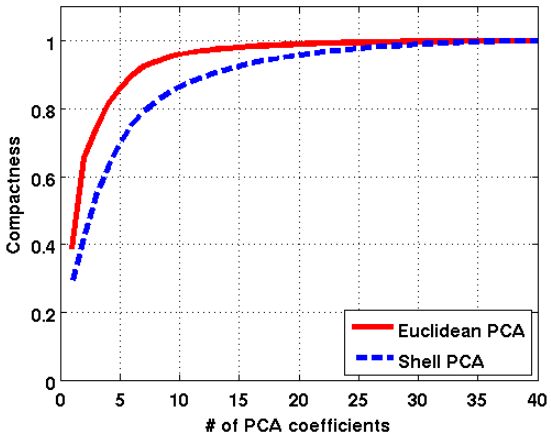
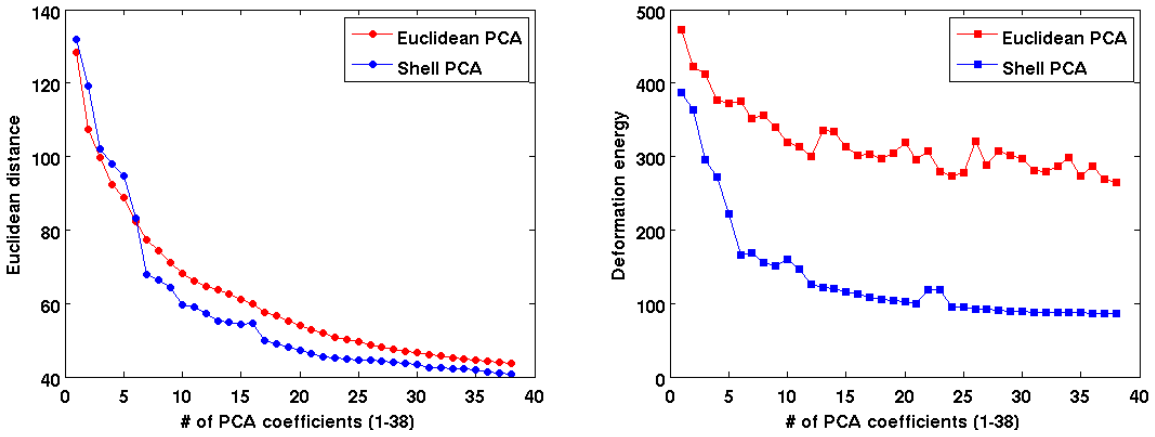


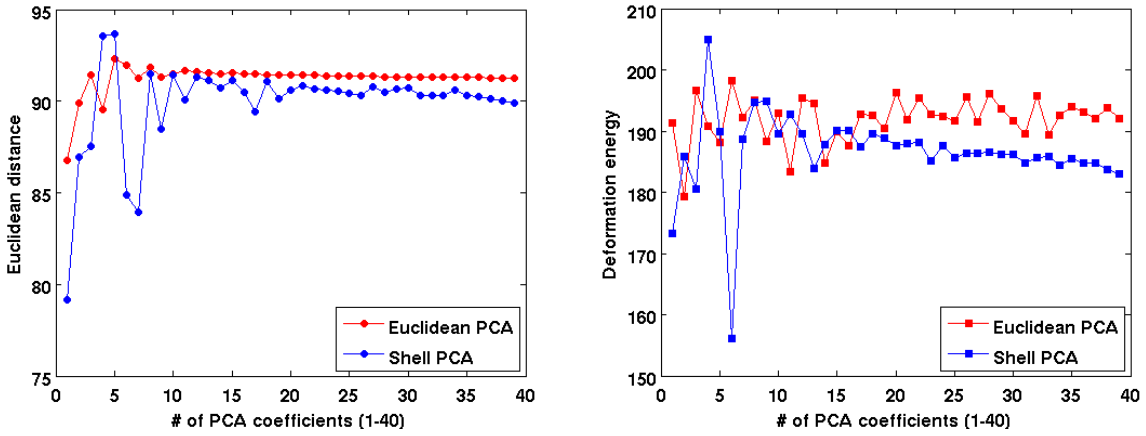
Figure 3.9: Compactness of Shell PCA and Euclidean PCA.



(a) Error in Euclidean distance

(b) Error in deformation energy

Figure 3.10: Generalisation error of Shell PCA and Euclidean PCA.



(a) Error in Euclidean distance

(b) Error in deformation energy

Figure 3.11: Specificity error of Shell PCA and Euclidean PCA.

Our method is capable of capturing non-linear deformations without using hand-crafted skeleton (*Challenge 2*) and is physically-inspired due to the usage of thin shell deformation energy (*Challenge 3*). However, the model is still based on vertex displacement and hence is alignment-dependent. As a result, we chose to minimise rigid alignment error, i.e. Procrustes analysis, before computing the displacement between input shapes and the average. This choice is arbitrary and affects effective shape variations extraction. Furthermore, thin shell deformation energy only behaves like a Riemannian distance within a small deformation. To this end, the shell PCA model allows for not-very-large deformations. In our experiments, facial expression is shown suitable to be modelled with Shell PCA, while human poses not. Taking uplifting arms and laying them down for an example, the displacements are not good representation for the pose changes and so it is not considered in our evaluation. In the next chapter, we would overcome these drawbacks by proposing a PGA variant that avoids performing any operations in the tangent space and works directly with objects lying on Riemannian manifold.

CHAPTER 4

PRINCIPAL GEODESIC ANALYSIS IN SHELL SPACE

4.1 Introduction

The analysis of principal variations on shape spaces has attracted a lot of attention. Classical tools from PCA on linear vector spaces has been transferred to shape spaces considered as Riemannian manifolds and applied, for instance, to tasks such as classification, reconstruction and clustering [119–124].

Such "statistical" shape models seek to learn principal modes that compactly capture shape variability within or between classes. The classical approach to statistical shape modelling is due to Kendall [125] and deals with objects represented by a configuration of landmark points. A point in Kendall's shape space corresponds to a configuration of landmarks in which the effect of rotation, translation and (optionally) scaling have been "factored out". Usually, linear PCA is performed in the tangent space to this Riemannian manifold. Great advances have been made in the decades since, including dense modelling of discretely sampled surfaces [24, 126], modelling of continuous curves on infinite dimensional Riemannian manifolds [88], hybrid articulated and statistical models [3, 26] and state-of-the-art human body models that capture dependencies between body shape, pose and dynamic deformations [11]. However, there remain important challenges in learning statistical models of nonlinear shape variation.

Shape space must be viewed in a Riemannian setting and statistical notions have to be adapted from the Euclidean setup. The method in the previous chapter did not do this. There was no underlying manifold representation and the model was built using vertex displacement that are not meaningful for large, nonlinear deformations. We introduce a nonlinear Riemannian principal component analysis and propose an effective time-discretisation thereof.

We take into account a Riemannian structure on the space of triangular surfaces, which reflects the physics of viscous shells with a metric measuring the energy dissipation caused by

Figure 4.1: Visualisation of physically sound bending (col. 3) and membrane (col. 5) energy dissipation between resting pose (col. 1) to sample poses (col. 2 and col. 4).

membrane distortion and normal bending (see Fig. 4.1 - note the high bending energy in the armpits and membrane energy associated with stretching at the knee).

The shell space in which we work is a space of equivalence classes of shapes that differ only by rigid body motions (see Fig. 4.3) and we take special care to transfer this invariance to our time-discrete statistical analysis. Therefore, our whole framework is invariant to rigid body motion and does not require a preprocessing alignment step.

The use of the words "learning" and "manifold" in this chapter should not cause confusion with *manifold learning*. Unlike manifold learning, we work with manifolds that arise as a natural property of the discrete shell model. We endow our input meshes with a physical model (the discrete shell model) so that we have a notion of the energy required to deform each shape. Then, we treat each input shape (now a shell) as a point on a high dimensional, nonlinear, Riemannian manifold - this is shell space. In this space, we perform statistical learning in a way that respects its known Riemannian geometry. In practice, our treatment of the manifold uses a time-discretisation for reasons of computational tractability. So, another potential source of confusion here is the use of the word "discrete". In the context of the discrete shell model, "discrete" refers to spatial discretisation of a continuous surface using a triangular mesh representation. In the context of our proposed time-discrete statistical model, "discrete" refers to discretisation of geodesic paths.

4.2 Review of principal geodesic analysis

Let us briefly recall classical Principal Components Analysis (PCA) before we translate the concept to Riemannian manifolds. For data points $s^i \in \mathbb{R}^{3n}$ with $i = 1, \ldots, m$, one defines the

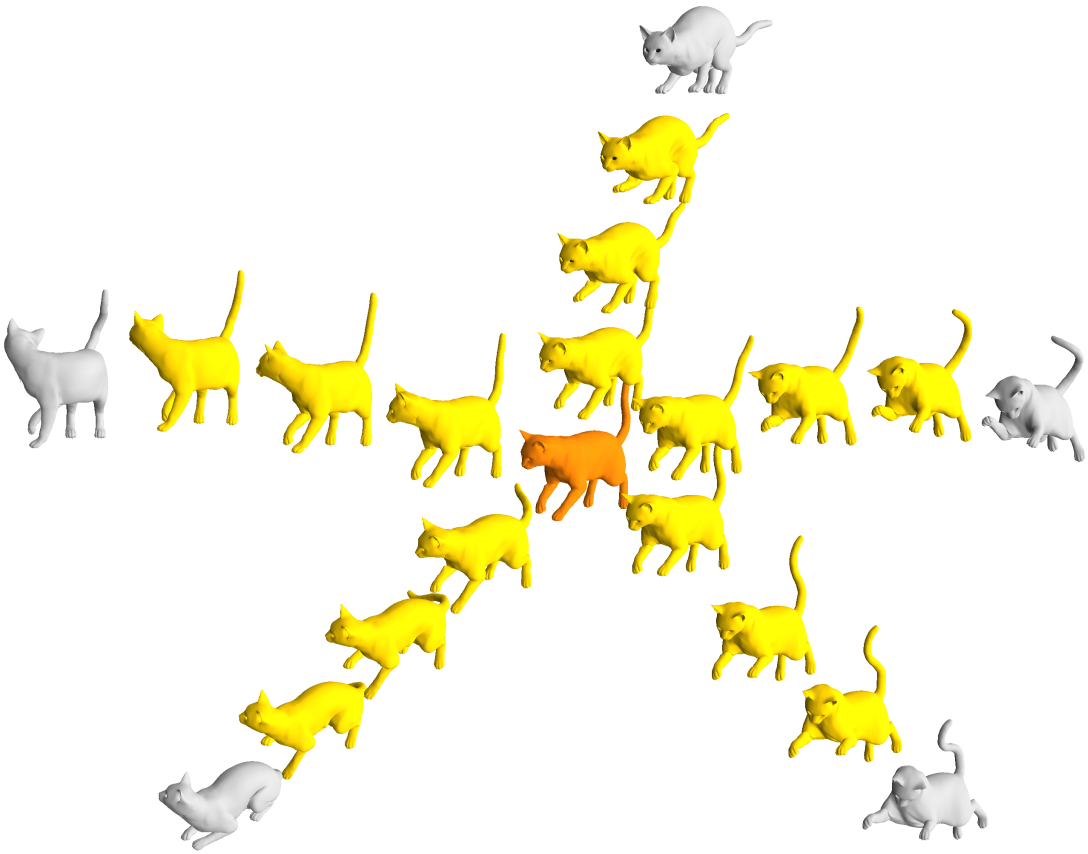


Figure 4.2: Visualisation of discrete geodesic spider connecting mean shape to sparse input data exhibiting highly non-linear soft-tissue deformations.

covariance operator

$$\mathbf{Cov} : \mathbb{R}^{3n} \to \mathbb{R}^{3n};\ y \mapsto \tfrac{1}{m} \sum_{i=1,\dots,m} \langle y, s_i - \bar{s} \rangle (s_i - \bar{s})$$

where $\bar{s} = \frac{1}{m} \sum_{i=1,\dots,m} s_i$ is the arithmetic average of the data and $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product. The covariance operator generalises the concept of covariance from finite to infinite dimensional settings. Note that in the Euclidean case, the covariance operator applied to y is nothing other than the matrix-vector product between the covariance matrix $\mathbf{C}$ the vector y.

On the linear subspace spanned by displacements of the data points from the mean, i.e. $\mathrm{span}(\{s_i - \bar{s} \,|\, i = 1, \dots, m\})$, there exists a sequence of nonnegative, monotonically decreasing eigenvalues λ_j and associated eigenvectors $\mathbf{v}_j$ such that $\mathbf{C}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ for $j = 1, \dots, m$. The eigenvectors are orthonormal, i.e. $\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}$, and are referred to as the principal components of the data set. In order to perform PCA efficiently, one uses the fact that the spectrum of $\mathbf{C} \in \mathbb{R}^{3n \times 3n}$ coincides with the spectrum of the Gram matrix $\mathbf{G} \in \mathbb{R}^{m \times m}$, where $\mathbf{G}_{ij} = \frac{1}{n} \langle s_i - \bar{s}, s_j - \bar{s} \rangle$ for $i, j = 1, \dots, m$. The jth principal component $\mathbf{v}_j$ can then be obtained from the jth eigenvector of the Gram matrix $\mathbf{w}_j = (w_{j,1}, \dots, w_{j,m})^T$ via the relation $\mathbf{v}_j = (\sqrt{\lambda_j})^{-1} \sum_{i=1,\dots m} w_{j,i}(s_i - \bar{s})$.

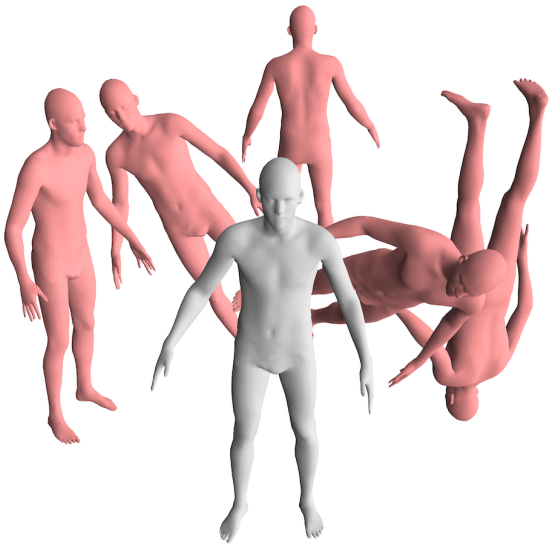


Figure 4.3: Equivalence classes of discrete shells incorporating rigid body motion invariance.

By definition PCA operates on a linear vector space with a scalar product. To transfer it to a manifold $\mathcal{M}$, a suitable vector space and an associated inner product need to be identified. Furthermore, the arithmetic average has to be replaced by a suitable average on the manifold.

4.2.1 Riemannian geometry

Let us denote by $(\mathcal{M},\, g)$ a (smooth) *Riemannian manifold* with metric g, i.e. $g_s(.,.)$ is an inner product on $T_s\mathcal{M}$ for each $s \in \mathcal{M}$. For a differentiable path $\mathfrak{s} : [0,1] \to \mathcal{M}$ the *path energy* is defined by

$$\mathcal{E}[(\mathfrak{s}(t))_{t\in[0,1]}] = \int_0^1 g_{\mathfrak{s}(t)}\left(\dot{\mathfrak{s}}(t), \dot{\mathfrak{s}}(t)\right)\ \mathrm{d}t\,, \tag{4.1}$$

where the velocity $\dot{\mathfrak{s}}(t)$ at time t is an infinitesimal variation of $\mathfrak{s}(t)$. Given two points $s_A, s_B \in \mathcal{M}$ a minimizing path $(\mathfrak{s}(t))_{t\in[0,1]}$ of Eq. (4.1) among all paths with $\mathfrak{s}(0) = s_A$ and $\mathfrak{s}(1) = s_B$ is called *shortest geodesic*. For fixed s_A, existence and uniqueness of a shortest connecting geodesic to s_B is given as long as s_B is in the vicinity of s_A. In particular, a geodesic path $\mathfrak{s} : [0,1] \to \mathcal{M}$ solves the Euler–Lagrange equation associated with the path energy. The latter is given by the boundary value problem $\nabla_{\dot{s}(t)}\dot{s}(t) = 0$ subject to $\mathfrak{s}(0) = s_A$ and $\mathfrak{s}(1) = s_B$. See Fig. 4.4 for an illustration. Here $\nabla_{\dot{s}(t)}$ denotes the covariant derivative along the curve. This basically implies that at each point of the curve, there is no way to further reduce the segment energy. If $\mathfrak{s} : [0,1] \to \mathcal{M}$ represents the solution to this problem, the *exponential map* is given by $\exp_{s_A}(v) = s(1)$ which is well-defined for sufficiently small v. Furthermore, for $t \in \mathbb{R}$ the exponential map fulfills the scaling law

$$\exp_{s_A}(tv) = s(t)\,.$$

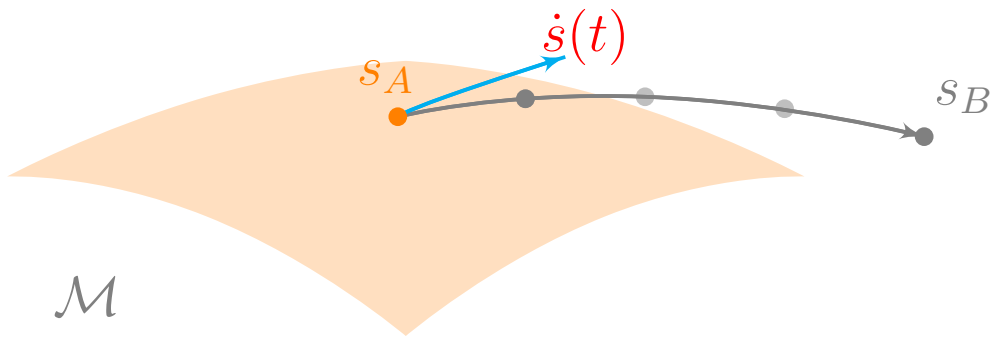


Figure 4.4: Two points sampled from a Riemannian manifold. The shortest geodesic path is shown with the initial velocity at s_A.

In particular, for sufficiently small $r > 0$ the exponential map $\exp_{s_A}$ is a bijection defined in the neighbourhood ball. The inverse operator is given by the *geometric logarithm* $\log_{s_A}$, i.e.

$$\log_{s_A}(s(t)) = tv\,,$$

for $t \in \mathbb{R}$.

For $s_B \in \mathcal{M}$ the logarithm $\log_{s_A}(s_B)$ can be considered as an infinitesimal representation of the geometric variation s_B with respect to s_A. Finally, the (squared) Riemannian distance is defined by

$$\operatorname{dist}^2(s_A, s_B) = \min_{\substack{\mathfrak{s}:[0,1]\to\mathcal{M} \\ \mathfrak{s}(0)=s_A,\,\mathfrak{s}(1)=s_B}} \mathcal{E}[\mathfrak{s}]\,. \tag{4.2}$$

4.2.2 Fréchet mean

In fact, the arithmetic average of input objects $s_1, \ldots, s_m \in \mathbb{R}^{3n}$ can be characterized as the minimizer of the functional $s \mapsto \sum_{i=1,\ldots,m} \operatorname{dist}^2(s, s^i)$ with $\operatorname{dist}(\cdot,\cdot)$ being the Euclidean distance. If we now consider instead a set of input objects $s_1, \ldots, s_m \in \mathcal{M}$ where the Euclidean distance is simply replaced by the distance on the manifold Eq. (4.2). To this end, we obtain a Riemannian mean $\bar{s}$, also called Fréchet mean [127]. In particular, Eq. (4.2) implies that $\bar{s}$ is the minimizer of

$$\bar{s} = \operatorname*{arg\,min}_{s\in\mathcal{M}} \sum_{i=1}^{m} \min_{\substack{\mathfrak{s}_i:[0,1]\to\mathcal{M} \\ \mathfrak{s}_i(0)=s,\mathfrak{s}_i(1)=s_i}} \mathcal{E}[\mathfrak{s}_i] \tag{4.3}$$

which is the sum of all path energies of a spider of paths $(\mathfrak{s}_i(t))_{t\in[0,1]}$ centered at s and ending at the input objects s_i (cf. Fig. 4.2). For small Riemannian distances shortest geodesics are unique

and the Fréchet mean of nearby objects on $\mathcal{M}$ is unique as well [128].

4.2.3 Gram matrix and principal components

Given a sequence of data points $s_1, \ldots, s_m \in \mathcal{M}$ and a corresponding Fréchet mean $\bar{s}$, the tangent vector $u_j = \log_{\bar{s}} s_j$ represents the geometric variation of s_j relative to the average $\bar{s}$ in the linear vector space $T_{\bar{s}}\mathcal{M}$. A Riemannian analogue of PCA (called Principal Geodesic Analysis (PGA) following [79]) can be obtained by taking the Riemannian metric $g_{\bar{s}}$ as the natural scalar product on the space $T_{\bar{s}}\mathcal{M}$.

The corresponding Gram matrix $\mathbf{G} \in \mathbb{R}^{m\times m}$ is then defined with its entries as $\mathbf{G}_{ij} = \frac{1}{m} g_{\bar{s}}(u_i, u_j)$ in analogy to the Euclidean case and the associated covariance operator is given by

$$\mathbf{Cov} : T_{\bar{s}}\mathcal{M} \to T_{\bar{s}}\mathcal{M};\ u \mapsto \tfrac{1}{m} \sum_{j=1,\ldots,m} g_{\bar{s}}(u, u_j) u_j.$$

As for PCA on $\mathbb{R}^{3n}$ we obtain a monotonically decreasing sequence of nonnegative eigenvalues λ_j and associated eigenvectors $\mathbf{v}_j \in T_{\bar{s}}\mathcal{M}$ such that $\mathbf{C}\mathbf{v}_j = \lambda_j \mathbf{v}_j$ for $j = 1, \ldots, J$. In particular, the eigenvectors are orthonormal with respect to $g_{\bar{s}}(.,.)$, i.e. $g_{\bar{s}}(\mathbf{v}_i, \mathbf{v}_j) = \delta_{ij}$. Again, the spectrum of $\mathbf{C} \in \mathbb{R}^{3n\times 3n}$ coincides with the spectrum of Gram's matrix $\mathbf{G} \in \mathbb{R}^{m\times m}$, hence it is advantageous to compute the latter one. The jth principal component is then obtained via the relation $\mathbf{v}_j = (\sqrt{\lambda_j})^{-1} \sum_{i=1,\ldots m} \mathbf{w}_{j,i} u_i$, where $\mathbf{w}_j = (\mathbf{w}_{j,1}, \ldots, \mathbf{w}_{j,m})^T$ denotes the jth eigenvector of Gram's matrix (corresponding to the eigenvalue λ_j).

4.2.4 Interpolation and extrapolation

Computing geodesics enables a straightforward, geometrically sound *interpolation* and *extrapolation* operation. Indeed, given s_A and s_B, we define $\mathcal{I}(s_A, s_B, t) = \mathfrak{s}(t)$ for $t \in \mathbb{R}$, where $\mathfrak{s}(t)_{t\in[0,1]}$ is the minimizer of the path energy Eq. (4.1) subject to $\mathfrak{s}(0) = s_A$ and $\mathfrak{s}(1) = s_B$, and outside $[0, 1]$ we use exponential map to extrapolate the geodesic, i.e. $\mathfrak{s}(t) = \exp_{s_A}(t \log_{s_A}(s_B))$.

In particular, for the case $\mathcal{I}(s_A, s_B, 0.5)$ we obtain the geodesic midpoint (i.e. Fréchet mean) of s_A and s_B. For the case $\mathcal{I}(s_A, s_B, -1) = \exp_{s_A}(-\log_{s_A}(s_B))$ we obtain the Riemannian reflection of s_B about s_A, such that s_A is the geodesic midpoint of $\mathcal{I}(s_A, s_B, -1)$ and s_B. For the case $\mathcal{I}(s_A, s_B, 2)$ we extrapolate beyond s_B, such that s_B is the geodesic midpoint between s_A and $\mathcal{I}(s_A, s_B, 2)$.

4.3 PGA in shell space

In the previous section, we review the classical Principal Geodesic Analysis (PGA). We will now transfer the concepts of Riemannian geometry and statistical PGA to a manifold defined by a physical model such that we could eventually build hybrid physical/statistical models.

We aim at developing a principal geodesic analysis on the space of triangular surfaces considered as a Riemannian manifold reflecting the physics of viscous shells. In particular, we regard a discrete shell as a triangular mesh approximation. For fixed mesh topology a discrete shell can be identified with a vector of vertex positions in $\mathbb{R}^{3n}$, where n is the number of vertices. This subset of $\mathbb{R}^{3n}$ is then equipped with a Riemannian metric which measures the energy dissipation caused by infinitesimal membrane and bending distortion. The subset turns out to be the manifold $\mathcal{M}$ we will be working on. The tangent space at some discrete shell s consists of vector fields $v \in \mathbb{R}^{3n}$ acting on the vertices of the triangular mesh. Next, we will describe the manifold of discrete shells.

4.3.1 The manifold of discrete shells

Riemannian metric on the space of discrete shells. To derive a metric we start with an elastic deformation model for the thin material layer. Let $\mathbf{W}[s, \tilde{s}]$ be the elastic energy that is needed to deform $s \in \mathcal{M}$ into $\tilde{s} \in \mathcal{M}$. To account for the physical properties of thin elastic shells, i.e. to measure membrane and bending distortions, we take into account a splitting of the energy

$$\mathbf{W}[s, \tilde{s}] = \mathbf{W}_{\text{mem}}[s, \tilde{s}] + \mathbf{W}_{\text{bend}}[s, \tilde{s}] \,.$$

On the discrete shell space we make use of the following definitions (Here, quantities with a tilde always refer to the deformed configuration):

$$\begin{aligned} \mathbf{W}_{\text{mem}}[s, \tilde{s}] &= \delta \sum_{t \in \mathcal{T}(s)} a_t W(\mathcal{G}_t) \,, \\ \mathbf{W}_{\text{bend}}[s, \tilde{s}] &= \delta^3 \sum_{e \in \mathcal{E}(s)} \frac{(\theta_e - \tilde{\theta}_e)^2}{a_e} l_e^2 \,, \end{aligned}$$

where $\mathcal{T}(s)$ and $\mathcal{E}(s)$ denote the set of triangles and edges of s. For more details on shell deformation energy, please refer to previous chapter (3.2.1).

Rigid body motion invariance. The key property of elastic energy is that it is strictly rigid body motion invariant, i.e.

$$\mathbf{W}[s, \tilde{s}] = \mathbf{W}[s, Q\tilde{s} + b] \tag{4.4}$$

for $Q \in SO(3)$ (the space of rotation matrices in $\mathbb{R}^3$) and $b \in \mathbb{R}^3$ (3D translation). The elastic energy $\mathbf{W}[s, \tilde{s}]$ can be rewritten in terms of the underlying deformation ϕ of the discrete shell s, i.e. $\mathbf{W}_s^D[\phi] = \mathbf{W}[s, \phi(s)]$. Furthermore, Heeren et al. [82] showed that this leads to a Riemannian metric. The naturally appearing rigid body motion invariance is an indispensable feature of our model which requires a careful treatment from the computational perspective. At first, we no longer treat single discrete shells s but equivalence classes of shells $[s] =$

$SO(3)s+\mathbb{R}^3$. As a consequence our manifold of discrete shells $\mathcal{M}$ is a space of such equivalence classes and tangent vectors are equivalence classes of vector fields. In fact an explicit treatment of equivalence classes of vector fields as tangent vectors of our discrete shell manifold in the principal geodesic analysis would be extremely cumbersome. Our observation that the underlying elastic energy $\mathbf{W}$ is rigid body motion invariant will help us to derive a suitable, strictly rigid body motion invariant algorithmic alternative. Indeed, we will base all components of the algorithm solely on discrete shells and avoid any appearance of vector fields.

A local approximation of the squared distance. It has been shown in [129] that for sufficiently smooth $\mathbf{W}$, there exists the approximation property

$$\mathbf{W}[s,\tilde{s}] = \operatorname{dist}^2(s,\tilde{s}) + O(\operatorname{dist}^3(s,\tilde{s}))\,. \tag{4.5}$$

Hence, we will think of $\mathbf{W}$ as a local approximation of the squared Riemannian distance in the following.

4.3.2 Principal variations and local submanifold

We now look at deriving a principal geodesic analysis (PGA) on the space of discrete shells that is invariant with respect to rigid body motions. However, as discussed above, metric evaluations are based on tangent vectors and thus require to treat (infinitesimal) rigid motions explicitly. *To avoid this explicit treatment we shall derive an alternative representation of PGA which omits any operations in the tangent space but works directly with objects lying on the manifold.* To this end, our formulation will be solely based on the squared distance and hence inherently rigid motion invariant.

We propose to study a Riemannian submanifold of $\mathcal{M}$ which is associated with the J dominant modes of the principal geodesic analysis given m input objects s_i on $\mathcal{M}$. In particular, we introduce an approximate submanifold which relies on an approximate Gram matrix and does not require tangential quantities at all.

Approximation of Gram's matrix. First, we substitute metric evaluations by evaluations of the squared distance as follows. Paths that minimise the path energy are known to have constant absolute velocity, i.e.

$$g_{\mathfrak{s}(t)}(\dot{\mathfrak{s}}(t),\dot{\mathfrak{s}}(t)) = g_{\mathfrak{s}(0)}(\dot{\mathfrak{s}}(0),\dot{\mathfrak{s}}(0)) = \operatorname{dist}^2(\mathfrak{s}(0),\mathfrak{s}(1)) \tag{4.6}$$

for all $t\in[0,1]$. This enables us to write

$$g_{\bar{s}}(u_j,u_j) = \operatorname{dist}^2(\bar{s},s_j) = \epsilon^{-2}\operatorname{dist}^2(\bar{s},\mathfrak{s}_j(\epsilon)) \approx \epsilon^{-2}\mathbf{W}[\bar{s},\mathfrak{s}_j(\epsilon)] \tag{4.7}$$

for some generic scaling factor $\epsilon>0$. Here $u_j=\log_{\bar{s}}(s_j)\in T_{\bar{s}}\mathcal{M}$ are the tangent vectors usually used in PGA whereas $\mathfrak{s}_j(\epsilon)=\mathcal{I}(\bar{s},s_j,\epsilon)\in\mathcal{M}$ are objects on the manifold, lying a distance ϵ

along the geodesic path from $\bar{s}$ to the s_j. In a more general case, metric evaluations between a pair of tangent vectors are obtained by using the polar formula

$$g_{\bar{s}}(u_j, u_i) = \frac{1}{2}\left(g_{\bar{s}}(u_j, u_j) + g_{\bar{s}}(u_i, u_i) - g_{\bar{s}}(u_j - u_i, u_j - u_i)\right) \tag{4.8}$$

and to approximate the last term we use

$$g_{\bar{s}}(u_j - u_i, u_j - u_i) \approx \epsilon^{-2}\mathrm{dist}^2(\mathfrak{s}_j(\epsilon), \mathfrak{s}_i(\epsilon)) \approx \epsilon^{-2}\mathbf{W}[\mathfrak{s}_j(\epsilon), \mathfrak{s}_i(\epsilon)], \tag{4.9}$$

Altogether, we are able to replace evaluations of the metric with squared distances on $\mathcal{M}$ via combining Eq. (4.7), Eq. (4.8) and Eq. (4.9). We define the entries of an approximative Gram's matrix $\mathbf{G}$

$$\mathbf{G}^{\epsilon}_{ij} = \frac{1}{2\epsilon^2}\Big(\mathrm{dist}^2(\bar{s}, s_i(\epsilon)) + \mathrm{dist}^2(\bar{s}, s_j(\epsilon)) - \frac{1}{2}(\mathrm{dist}^2(s_i(\epsilon), s_j(\epsilon)) + \mathrm{dist}^2(s_i(\epsilon), s_j(\epsilon)))\Big). \tag{4.10}$$

for $i, j = 1, \ldots, m$. The additional symmetrisation in the last terms ensures symmetry of $\mathbf{G}$. Again, due to the rigid body motion invariance the resulting $\mathbf{G}$ does not depend on the chosen representation of the equivalence classes of discrete shells. As before we obtain approximate eigenvalues λ^{ϵ}_j and corresponding eigenvectors $\mathbf{w}^{\epsilon}_j \subset \mathbb{R}^m$.

Principal variations versus principal components. Next, we define an approximate (nonlinear) principal variation p^{ϵ}_j on $\mathcal{M}$ as a replacement of the principal component (modes) $\mathbf{v}_j \in T_{\bar{s}}\mathcal{M}$. Given $u = \sum_{i=1,\ldots,m} \alpha_i u_i$ for a coefficient vector $\alpha \in \mathbb{R}^m$ with $\sum_{i=1,\ldots,m} \alpha_i = 1$, u can obviously be characterized as the minimizer of the quadratic energy functional $u \mapsto \sum_{i=1,\ldots,m} \alpha_i g_{\bar{s}}(u - u_i, u - u_i)$. Using Taylor expansion for a given coefficient vector α this implies that for

$$p^{\epsilon} := p^{\epsilon}[\alpha] := \arg\min_{p \in \mathcal{M}} \sum_{i=1,\ldots,m} \alpha_i \mathrm{dist}^2(s^i(\epsilon), p),$$

This is just a weighted elastic average on the $\mathcal{M}$. Here, we have used the smoothness of the metric, in particular that $s \mapsto g_s(\cdot, \cdot)$ is smooth. Hence, $p^{\epsilon} \in \mathcal{M}$ can be considered as a *nonlinear* variation of $\bar{s}$ corresponding to the *linear* infinitesimal variation $u \in T_{\bar{s}}\mathcal{M}$.

Now, we apply this replacement procedure to the (rescaled) principal components $\mathbf{v}^{\epsilon}_j = (\sqrt{\lambda^{\epsilon}_j})^{-1} \sum_{i=1,\ldots,m} \mathbf{w}^{\epsilon}_{j,i} u_i$ for $j = 1, \ldots, J$, where $\{\lambda^{\epsilon}_j\}_j$ are eigenvalues of $\mathbf{G}^{\epsilon}$ with corresponding eigenvectors $\{\mathbf{w}^{\epsilon}_j\}_j \subset \mathbb{R}^m$. Actually, we assume a uniform rescaling of the vector $\mathbf{w}^{\epsilon}_j = (w^{\epsilon}_{j,1}, \ldots, w^{\epsilon}_{j,m})^T$ such that $\sum_i w^{\epsilon}_{j,i} = \sqrt{\lambda^{\epsilon}_j}$, however, we shall make use of the same

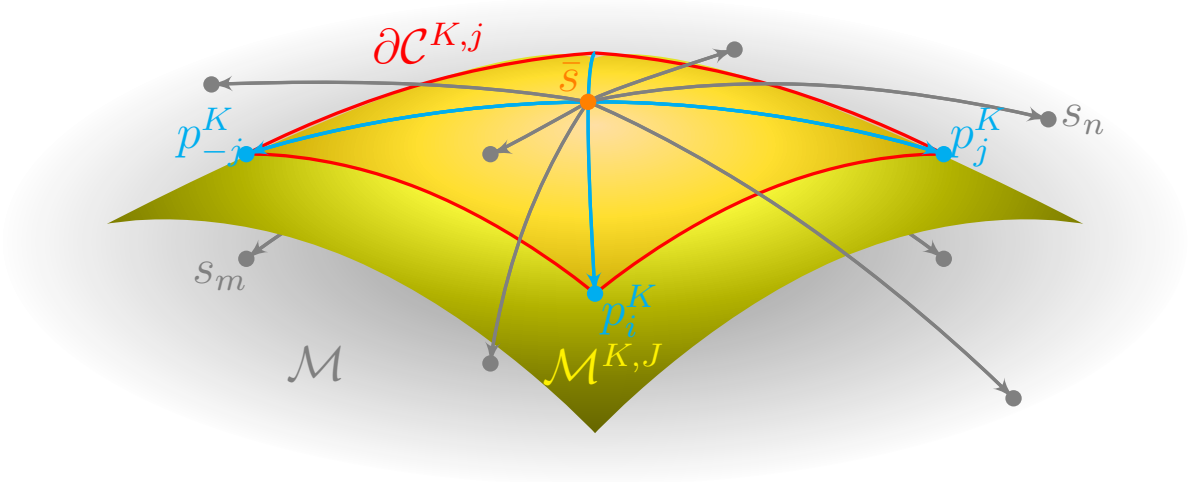


Figure 4.5: Approximative manifold $\mathcal{M}^{\epsilon,J}$ (yellow) spanned by nonlinear combinations of the principal variations $\{p^\epsilon_j\}_j$ and their negative reflections $\{p^\epsilon_{-j}\}_j$, respectively. Note that the input shapes $\{s_n\}_n \subset \mathcal{M}$ do not lie on $\mathcal{M}^{\epsilon,J}$ in general. The boundary of the Riemannian polyhedron defined by the vertices $\{p^\epsilon_j, p^\epsilon_{-j}\}_j$ and positive weights $\alpha \geq 0$ is shown in red.

variable. Then we define the *principal variation* by

$$p^\epsilon_j := p^\epsilon[\mathbf{w}^\epsilon_j] := \arg\min_{p\in\mathcal{M}} \sum_{i=1,\dots,m} w^\epsilon_{j,i} \mathrm{dist}^2[s_i(\epsilon), p], \tag{4.11}$$

for $j = 1, \dots, J$ as a point on $\mathcal{M}$ close to $\bar{s}$ with $\frac{1}{\epsilon}\log_{\bar{s}} p^\epsilon_j \to \mathbf{v}_j$ for $\epsilon \to 0$.

Defining the submanifold. Finally, we are able to define a local submanifold $\mathcal{M}^{\epsilon,J}$ "spanned" by these principal variations. This is analogous to the linear subspace spanned by the principal components in classical PCA. Futhermore we consider Eq. (4.11) for the J dominant principal variations and also their associated reflections $p^\epsilon_{-j} = \mathcal{I}(\bar{s}, p^\epsilon_j, -1)$ (the sign of a principal component is arbitrary so our submanifold includes variations in both directions). Points lying within our submanifold are obtained by taking a Riemannian combination (parameterization) of p^ϵ_j with a vector of weights $\alpha = (\alpha_{-J}, \dots, \alpha_{-1}, \alpha_1, \dots, \alpha_J) \in \mathbb{R}^{2J}$ subject to $\sum_{j=-J,\dots,J} \alpha_j = 1$. Therefore, we define the submanifold as

$$\mathcal{M}^{\epsilon,J} = \left\{ \arg\min_{p\in\mathcal{M}} \sum_{j=-J}^{J} \alpha_j \mathrm{dist}^2(p, p^\epsilon_j) \;\middle|\; \sum_{j=-J}^{J} \alpha_j = 1 \right\},$$

with the notational convention $\alpha_0 = 0$, i.e. a definition of p^ϵ_0 is not necessary. Note that in particular $\bar{s} \in \mathcal{M}^{\epsilon,J}$. Geometrically, $\mathcal{M}^{\epsilon,J}$ is a submanifold of $\mathcal{M}$. If we additionally impose $\alpha_j \geq 0$ for all j, we obtain a convex Riemannian polyhedron with vertices p^ϵ_j for $j = -J, \dots, -1, 1, \dots, J$. In either case, we might allow for non-vanishing α_0 and $p^\epsilon_0 := \bar{s}$, which does not alter the definition of $\mathcal{M}^{\epsilon,J}$. Let us emphasize that we are interested in a small ball on $\mathcal{M}^{\epsilon,J}$ around $\bar{s}$. The tangent space $T_{\bar{s}}\mathcal{M}^{\epsilon,J}$ is spanned by the $\log_{\bar{s}} p^\epsilon_j$ (which converge

to $\mathbf{v}_j$ for $\epsilon \to 0$). Altogether, we get that $\frac{1}{\epsilon}\log_{\bar{s}}\left(\mathcal{M}^{\epsilon,J}\right) \to \operatorname{span}(\{v_1,\dots,v_J\})$ for $\epsilon \to 0$. This approximation property holds even if $\alpha_0 \neq 0$ is allowed. That both principal variations p_j^ϵ and their reflections p_{-j}^ϵ are indispensable to our construction reflects the fact that the infinitesimal counterpart, the principal components $\mathbf{v}_j$, generate one dimensional subspaces rather than rays.

Algorithm 2 Computation of discrete Shell PGA principal variations

Input: $s_1,\dots,s_m \in \mathcal{M}$, $K \in \mathbb{N}$, $J \in \mathbb{N}$, $T \in \mathbb{N}$
Output: geodesic average $\bar{s}$, geodesic paths $\mathfrak{s}_{i=1,\dots,m}$, principal variations $\mathbf{v_1},\dots,\mathbf{v_m}$
initialise elastic average as geodesic average $\bar{s}_0$
// optimise geodesic average and geodesic paths
for $t : 1$ to T **do**
  for $i : 1$ to m **do**
    optimise geodesic path $\mathfrak{s}_i$ to minimise (Eq. 4.12)
  end for
  update geodesic average $\bar{s}_t$ as elastic average of $\mathfrak{s}^1_{i=1,\dots,m}$
end for
$\bar{s} = \bar{s}_T$
// computation of Shell PGA principal variations
compose discrete approximation Gram's matrix $\mathbf{G}^K$ using (Eq. 4.17)
do SVD decomposition of Gram's matrix $\mathbf{G}^K$
compute principal variations using (Eq. 4.18)

4.3.3 Time-discrete principal geodesic analysis

We have derived a representation of PGA which is solely based on the squared distance. In particular, our formation is based on the local approximation property Eq. (4.5). However, each single evaluation of $\operatorname{dist}^2(\cdot,\cdot)$ via Eq. (4.2) requires the minimisation of the path energy, which might be infeasible e.g. from the computational point of view. Following [129] we would introduce a variational time-discretization of the path energy which is computationally advantageous. At its heart, this discretisation is invariant to rigid body motions by construction. Building on this scheme we finally propose a corresponding time-discrete setting for our PGA model.

Discrete geodesic path. For some integer K a time-discrete sampling $(s_0, s_1, \dots, s_K)$ of a given continuous path $\mathfrak{s} = (\mathfrak{s}(t))_{t\in[0,1]}$ on $\mathcal{M}$ is given by $s_k = s(\frac{k}{K})$. Then the path energy Eq. (4.1) satisfies the estimate

$$\mathcal{E}[s] \geq K \sum_{k=1,\dots,K} \operatorname{dist}^2(s_{k-1}, s_k)\,,$$

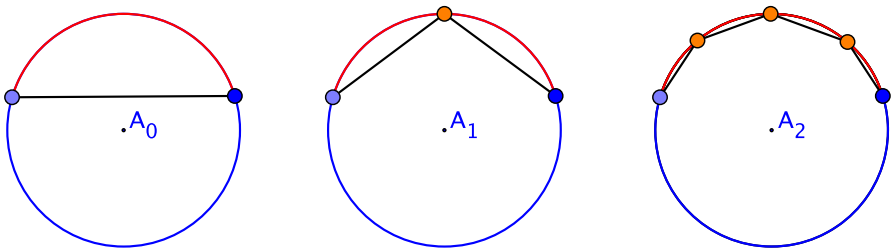


Figure 4.6: Sample geodesic path

where equality holds if and only if s is already a shortest geodesic. This could be easily verified through a circle example shown in Fig. 4.6 where the circle length is estimated using segments. Now, we replace $\mathrm{dist}^2(\cdot,\cdot)$ by a suitable approximation. Since for the elastic energy $\mathbf{W}$ introduced above, the approximation property Eq. (4.5) is valid. This motivates the definition of the *discrete path energy*

$$\mathbf{E}[s_0,\ldots,s_K] = K\sum_{k=1}^{K}\mathbf{W}[s_{k-1},s_k] \tag{4.12}$$

of a *discrete path* $(s_0,\ldots,s_K)$. For given shells s_A and s_B in $\mathcal{M}$ we call the discrete path $(s_0,\ldots,s_K)$ a *discrete shortest geodesic*, if $(s_0 = s_A,\ldots,s_K = s_B)$ is a minimizer of the discrete path energy Eq. (4.12). Physically, we could consider a discrete path as a chain of springs connecting subsequent shells on the manifold. Then, $\mathbf{E}[s_0,\ldots,s_K]$ is the total elastic energy of this chain and a discrete geodesic is obtained when relaxing this energy via adjusting the intermediate shells $s_1,\ldots,s_{K-1}$ on the manifold. A discrete geodesic must satisfy the system of Euler–Lagrange equations given by

$$\mathbf{W}_{,2}[s_{k-1},s_k] + \mathbf{W}_{,1}[s_k,s_{k+1}] = 0 \tag{4.13}$$

for $k = 1,\ldots,K-1$, where $\mathbf{W}_{,i}$ denotes the variation with respect to the ith argument. This implies that for any three-point geodesic s_{k-1}, s_k, s_{k+1}, the mid-point s_k has to be the stationary point.

Discrete Fréchet mean. Next, in analogy to Eq. (4.3) we define the associated *discrete Fréchet mean* $\bar{s}$ as the minimizer of

$$s \mapsto \sum_{\substack{j=1,\ldots,m \\ s_j^0=\bar{s},\, s_j^K=s^j}} \mathbf{E}[s_j^0,\ldots s_j^K]\,, \tag{4.14}$$

which is the sum of all discrete path energies of a spider of discrete paths $(s_j^0, s_j^1,\ldots s_j^K)$ centered at s (cf. Fig. 4.7). We denote s_j^k as kth shell along the discrete path associated with the jth

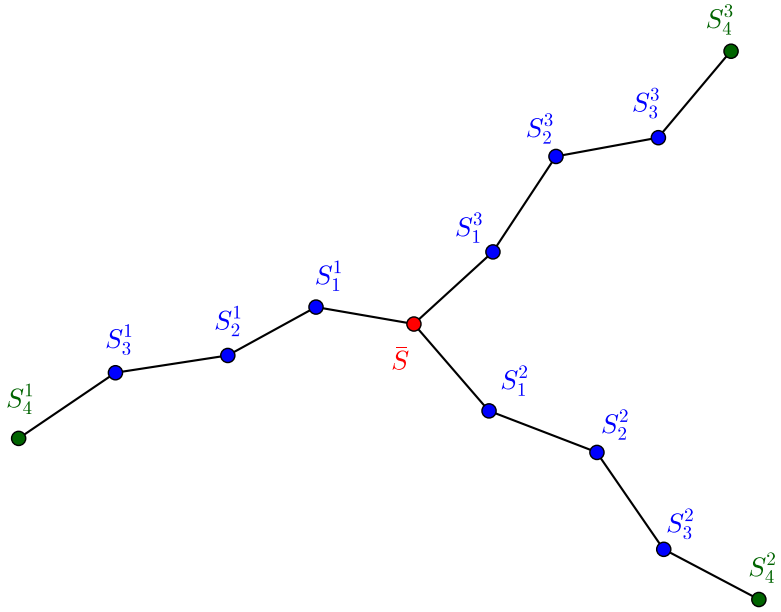


Figure 4.7: An example of geodesic spider centered at $\bar{S}$ with three inputs.

input shape s_j. The degrees of freedom (DOFs) are the shells defining the m polygonal paths $(s_j^0, s_j^1, \dots s_j^K)$ (with $3mn(K-1)$ DOFs) connecting the input shells s_i and $\bar{s}$ (with its $3n$ DOFs). Each arc of the polygonal spider has to solve the system of Euler–Lagrange equations for a single discrete K-geodesic (i.e Eq. (4.13) for $0 < k < K$) and the coupling at the center is described by the Euler–Lagrange equation Eq. (4.14).

Discrete interpolation and extrapolation. Analogous to the continuous setup we require a discrete geodesic interpolation and extrapolation method for given $s_A, s_B \in \mathcal{M}$.

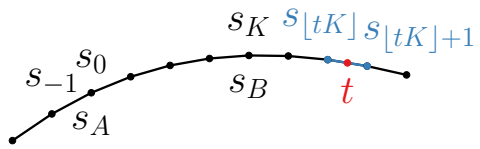
Easy: For $t = k/K$ for some $0 \le k \le K$ we set $\mathbf{I}^K(s_A, s_B, k/K) = s^k$, where $(s_0, \dots, s_K)$ is solution of the system Eq. (4.13) subject to $s_0 = s_A$ and $s_K = s_B$.

Medium: For $t = r/K$ with arbitrary $r \in \mathbb{Z}$ we define a discrete extrapolation by an iterative scheme based on the following induction: Assume $k \ge K$, such that s_{k-1} and s_k are already known, then we define s_{k+1} to be the solution of Eq. (4.13). Likewise, for $k \le 0$, such that s_k and s_{k+1} are already known, we define s_{k-1} to be the solution of Eq. (4.13). Hence for two arbitrary integer $R_1, R_2 \in \mathbb{N}$ we construct a sequence $(s_{-R_1}, \dots, s_{-1}, s_0, s_1, \dots, s_{R_2})$ iteratively and set $\mathbf{I}^K(s_A, s_B, r/K) = s^r$ for $-R_1 \le m \le R_2$. Note that this sequence solves Eq. (4.13) for $-R_1 < k < R_2$, hence represents a discrete geodesic by construction.

Hard: Finally, for general $t \in \mathbb{R}$ we denote $t(K) = tK - \lfloor tK \rfloor$ (where $\lfloor \cdot \rfloor$ is the usual floor function returning the largest integer less than or equal to the argument). We now define $\mathbf{I}^K(s_A, s_B, t)$ as the minimizer of

$$(1 - t(K))\mathbf{W}[s^{\lfloor tK \rfloor}, s] + t(K)\mathbf{W}[s, s^{\lfloor tK \rfloor + 1}], \tag{4.15}$$

where $s^r = \mathbf{I}^K(s_A, s_B, r/K)$ for $r \in \mathbb{Z}$, as described above. Intuitively, Eq. (4.15) is a weighted average of the two shells on the discrete path between which the desired t lies.



In analogy to the continuous case $\mathbf{I}^K(s_A, s_B, -1)$ defines a discrete Riemannian reflection of s_B about s_A. It has been shown in [129] that this discrete interpolation and extrapolation scheme converges to its continuous counterpart—subject to suitable assumptions.

Discrete Gram matrix and principal variations. One could immediately derive a discrete Gram matrix just by replacing the continuous logarithms $u_j \in T_{\bar{s}}\mathcal{M}$ by a corresponding discretisation, i.e. a discrete tangent field. However, this would require us to take explicit care to retain rigid body motion invariance by dealing with equivalence classes of discrete tangent fields. To avoid this, we exploit the concept of principal variations introduced in Sec. 4.3.2. To this end, we adapt Eq. (4.10) by replacing $\mathrm{dist}^2(\cdot,\cdot)$ by $\mathbf{W}[\cdot,\cdot]$, $\bar{s}$ by $\bar{s}^K$ as well as $\mathcal{I}$ by $\mathbf{I}^K$, i.e. s_j^ϵ by $s_j^{\epsilon,K} = \mathbf{I}^K(\bar{s}^K, s_j, \epsilon)$, and obtain

$$\mathbf{G}_{ij}^{\epsilon,K} = \frac{1}{2m\epsilon^2}\left(\mathbf{W}[\bar{s}^K, s_i^{\epsilon,K}] + \mathbf{W}[\bar{s}^K, s_j^{\epsilon,K}] \; - \frac{\mathbf{W}[s_i^{\epsilon,K}, s_j^{\epsilon,K}] + \mathbf{W}[s_j^{\epsilon,K}, s_i^{\epsilon,K}]}{2}\right). \tag{4.16}$$

for $i, j = 1, \ldots, m$. The additional symmetrisation in the last terms ensures symmetry of the resulting approximate Gram matrix $\mathbf{G}^{\epsilon,K}$. Let us emphasize that $\mathbf{W}$ is rigid body motion invariant by Eq. (4.4). Thus $\mathbf{G}^{\epsilon,K}$ can be computed based on any representations of the equivalence classes of discrete shells. Following the discrete calculus developed in [129], we know that $\bar{s}^K$ converges to $\bar{s}$ and $\mathbf{G}^{\epsilon,K}$ to $\mathbf{G}^\epsilon$ for $K \to \infty$ and $\epsilon \to 0$. This convergence is demonstrated empirically in Fig. 4.8.

More concretely, we omit K and replace $s_i^{\epsilon,K}$ with s_i^1, where $\epsilon = \frac{1}{K}$. In such case, the first node along the discrete K-geodesic from $\bar{s}$ to s_i is retrieved. Hence, we have the following

$$\mathbf{G}_{ij} = \frac{K^2}{2m}\left(\mathbf{W}[\bar{s}, s_i^1] + \mathbf{W}[\bar{s}, s_j^1] \; - \frac{\mathbf{W}[s_i^1, s_j^1] + \mathbf{W}[s_j^1, s_i^1]}{2}\right). \tag{4.17}$$

for $i, j = 1, \ldots, m$.

Next, we consider a discrete counterpart of the principal variation p_j^ϵ in Eq. (4.11). The eigenvalues and eigenvectors of $\mathbf{G}^{\epsilon,K}$ are denoted by $(\lambda_j^{\epsilon,K})_j$ and $(\mathbf{w}_j^{\epsilon,K})_j$, respectively, with $\mathbf{w}_j^{\epsilon,K} = (w_{j,1}^{\epsilon,K}, \ldots, w_{j,n}^{\epsilon,K})^T$. We have to proceed with special care when replacing $\mathrm{dist}^2(\cdot,\cdot)$ by $\mathbf{W}[\cdot,\cdot]$ in Eq. (4.11). If $w_{j,i}^{\epsilon,K} < 0$ for some i and j we replace $w_{j,i}^{\epsilon,K}$ by $|w_{j,i}^{\epsilon,K}|$ and $s_i^{\epsilon,K}$ by its discrete geometric reflection at $\bar{s}^K$ using the discrete exponential shooting in negative time direction, i.e. $\mathbf{I}^K(\bar{s}^K, s_i, -\epsilon)$. This is necessary because $\mathbf{W}$ is no longer quadratic (unlike squared distance) and there is no a priori control of the growth of $\mathbf{W}$ for general coefficients $w_{j,i}^{\epsilon,K} \in \mathbb{R}$. Therefore, we have to enforce all coefficients to be non-negative in the objective

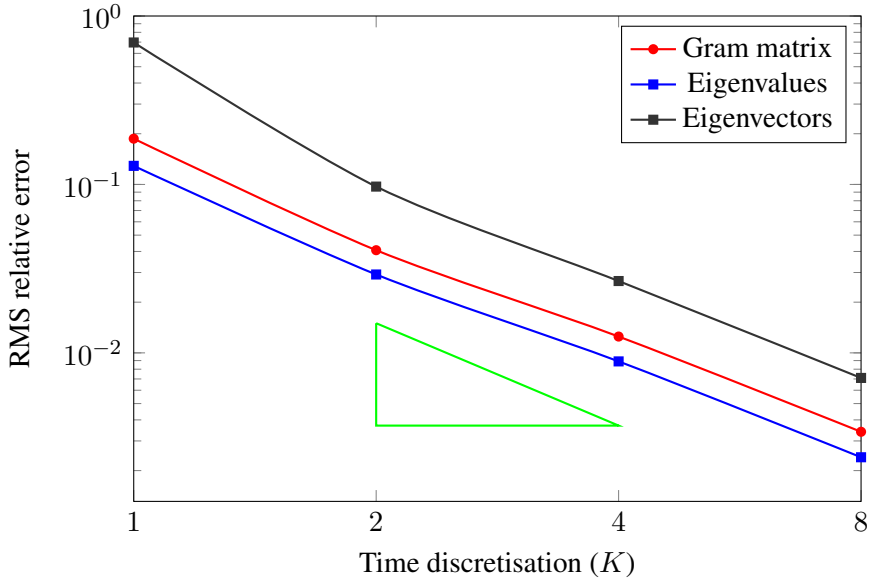


Figure 4.8: Convergence of the discrete Gram matrix and its eigenvectors and eigenvalues as $K \to \infty$ for the SCAPE dataset shown in Fig. 4.11. We show RMS relative error, using $K_{\text{max}} = 16$ as pseudo ground truth. Second order convergence is illustrated by the green triangle.

function. Altogether we obtain *discrete principal variations* by

$$p_j^{\epsilon,K} := \arg\min_{p \in \mathcal{M}} \sum_{i=1,\dots,m} |w_{j,i}^{\epsilon,K}| \, \mathbf{W}\left[\mathbf{I}^K\left(\bar{s}^K, s_i, \operatorname{sgn}(w_{j,i}^{\epsilon,K})\epsilon\right), p\right] \; . \tag{4.18}$$

Let us remark that for $\epsilon = \frac{1}{K}$ the shells $s_i^{\epsilon,K}$ are just an evaluation of the first shell of the discrete K-geodesic connecting $\bar{s}^K$ and s_i. The definition of a discrete Riemannian submanifold $\mathcal{M}^{\epsilon,K,J}$ is straightforward. We just have to replace once more $\operatorname{dist}^2(\cdot,\cdot)$ by $\mathbf{W}[\cdot,\cdot]$ and p_j^ϵ by $p_j^{\epsilon,K}$ in the definition of $\mathcal{M}^{\epsilon,J}$. Finally, discrete geodesic reflections are defined as $p_{-j}^{\epsilon,K} = \mathbf{I}^K(p_j^{\epsilon,K}, \bar{s}^K, 1)$ for $j = 1, \dots, J$.

The geometric average and the principal variations are representations of the underlying equivalence classes. Following the convergence theory in [129] we expect that for an eigenvalue of multiplicity 1 and for $K \to \infty$ and $\epsilon \to 0$ the eigenvalues $\lambda_j^{\epsilon,K}$ converge to their continuous counterpart λ_j and $\epsilon^{-1} \log_{\bar{s}^K} p_j^{\epsilon,K}$ converges (up to scaling) to a representative of the corresponding principal component $\mathbf{v}_j$.

Algorithm 3 Reconstructing an unseen shape using Shell PGA

Input: principal variations $\{p_1, \ldots, p_J\}$, geodesic average $\bar{s}$, $K \in \mathbb{N}$, unseen shape $q \in \mathcal{M}$
Output: reconstruction of shape $\mathcal{P}[q]$
// obtain reflections of principal variations
$p_{-j} = \mathbf{I}(\bar{s}, p_j, 1)$
// compute scaling length so that scaled local input could be well-reconstructed
solve for ρ^ϵ using (Eq. 4.3.4)
// compute geodesic path for input shape q
optimise path $\{\bar{s}, q_1, \ldots, q_K = q\}$
// scaling local variation of input shape
scale q_1 via $q_1^\epsilon = \mathbf{I}(\bar{s}, q_1, \rho^\epsilon)$
// reconstruct scaled local variation (the first shape along the geodesic)
reconstruct q_1^ϵ using (Eq. 4.19)
// reconstructing local input via rescaling
$\mathcal{P}[q_1] = \mathbf{I}(\bar{s}, q_1^\epsilon, 1/\rho^\epsilon)$
// reconstruct input shape by shooting
reconstruct q as $\mathcal{P}[q] = \exp_{\bar{s}}^{K-1} \mathcal{P}[q_1]$

4.3.4 Compression via submanifold projection

In this section we will derive a suitable approximate projection of a given shell $s \in \mathcal{M}$ onto the (approximate) submanifold $\mathcal{M}^{\epsilon,J}$ resp. $\mathcal{M}^{\epsilon,K,J}$ induced by the geometric mean $\bar{s}$ and the set of the J dominant principal variations for a set of input shells.

The classical Riemannian projection given the dominant J principal components $\mathbf{v}_1, \ldots, \mathbf{v}_J$ would work as follows: First compute an infinitesimal representation $v = \log_{\bar{s}} s$ of s in $T_{\bar{s}}\mathcal{M}$, then project v (locally) onto the subspace $\operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$ via the formula $v^J = \sum_{j=1,\ldots,J} g(v, \mathbf{v}_j)\mathbf{v}_j$ and finally compute the projection $\mathcal{P}[s] = \exp_{\bar{s}} v^J$. Note that this closed-form projection identity for v^J only holds if $\{\mathbf{v}_1, \ldots, \mathbf{v}_J\}$ is an orthonormal system.

Again, the incorporation of rigid body motion invariance needs to be delicately handled. Just replacing the scalar products on $T_{\bar{s}}\mathcal{M}$ in the above projection by the approximation used in the definition of the discrete Gram matrix in Eq. (4.16) does not lead to a satisfactory solution. Actually, the expected orthogonality relation of the principal variations holds only approximately and that deteriorates as operations such as scaling and optimisation involve in (4.18). Instead, we propose to perform a nonlinear projection onto the local approximating manifold $\mathcal{M}^{\epsilon,K,J}$. To derive this projection we first rewrite the orthogonal projection onto an affine subspace in Euclidean space.

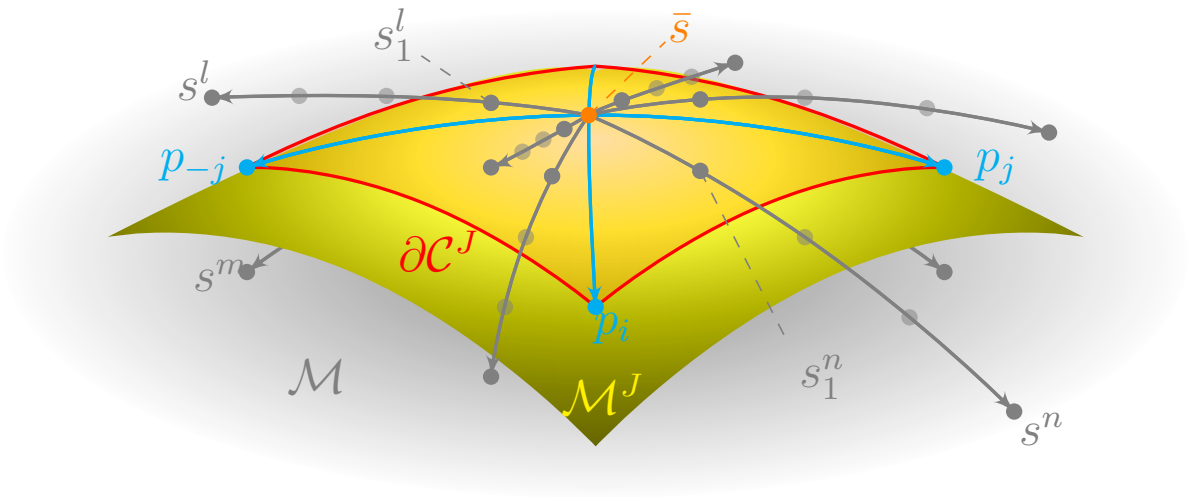


Figure 4.9: Submanifold $\mathcal{M}^J$ (yellow) and polyhedron $\mathcal{C}^J \subset \mathcal{M}^J$ (with red boundary) spanned by nonlinear combinations of principal variations $\{p_j\}_j$. Note that the input shapes $\{s_k\}_k \subset \mathcal{M}$ do not lie on $\mathcal{M}^J$ in general. The polygonal spider connecting input shapes and Fréchet mean is drawn in grey.

Let

$$\mathcal{U} = \left\{ \sum_{j=-J}^{J} \alpha_j q_j \,\middle|\, \sum_{j=-J}^{J} \alpha_j = 1 \right\}$$

be the J dimensional subspace of $\mathbb{R}^N$ containing the vertices q_j with q_{-j} being the reflection of q_j with the notational convention $\alpha_0 = 0$, $q_0 = 0$. Then the orthogonal projection $\mathcal{P}_{Euc}[p] \in \mathcal{U}$ for a point $p \in \mathbb{R}^N$ is given by

$$\mathcal{P}_{Euc}[p] = \arg\min_{q \in \mathcal{U}} \operatorname{dist}^2(p, q)$$

where $\operatorname{dist}^2(\cdot,\cdot)$ is the squared Euclidean distance. As above the constraint $q \in \mathcal{U}$ can be rephrased as $q = \arg\min_{\tilde{q}\in\mathbb{R}^N}\{\sum_j \alpha_j \operatorname{dist}^2(q_j, \tilde{q}) \mid \sum_j \alpha_j = 1\}$.

This formulation translates $1-1$ to the local projection of a shell $s \in \mathcal{M}$ onto the local submanifold $\mathcal{M}^{\epsilon,J}$ "spanned" by the principal variations $\{p_j^\epsilon\}_j$ and their geodesic reflections $\{p_{-j}^\epsilon\}_j$ as illustrated in Fig. 4.9. First, in order to make sure the given shape s could be locally represented within the submanifold, we scale s by defining $s^\epsilon = \mathcal{I}(\bar{s}, s, \rho^\epsilon)$ where

$$\rho^\epsilon := \frac{1}{2} \frac{\min_j \operatorname{dist}(\bar{s}, p_j^\epsilon)}{\operatorname{dist}(\bar{s}, s)} .$$

Note that $\rho^\epsilon \sim \epsilon$ since $\operatorname{dist}(\bar{s}, p_j^\epsilon) \sim \epsilon \operatorname{dist}(\bar{s}, s_i)$. Second, we compute a local projection as the

best approximation of s^ϵ on the submanifold $\mathcal{M}^{\epsilon,J}$, i.e. we define

$$\mathcal{P}^\epsilon_{loc}[s^\epsilon] = \arg\min_{q\in\mathcal{M}^{\epsilon,J}} \mathrm{dist}^2(s^\epsilon, q)\,, \tag{4.19}$$

where $\mathrm{dist}^2(\cdot,\cdot)$ is the Riemannian distance on $\mathcal{M}$. Finally, we rescale the local projection to define an approximate projection $\mathcal{P}^\epsilon[s] = \mathcal{I}(\bar{s}, \mathcal{P}^\epsilon_{loc}[s^\epsilon], 1/\rho^\epsilon)$. Now, we take into account the discrete geodesic calculus. As before, we set $s^{\epsilon,K} = \mathbf{I}^K(\bar{s}^K, s, \rho^\epsilon)$ and replace $\mathrm{dist}^2(\cdot,\cdot)$ by $\mathbf{W}[\cdot,\cdot]$ as well as p^ϵ_j by $p^{\epsilon,K}_j$ in the definition of the local projection. In explicit define

$$\mathcal{P}^{\epsilon,K}_{loc}[s^{\epsilon,K}] = \arg\min_{q\in\mathcal{M}^{\epsilon,K,J}} \mathbf{W}[s^{\epsilon,K}, q]\,, \tag{4.20}$$

which is equivalent to

$$q = \arg\min_{\tilde{q}\in\mathcal{M}} \Big\{ \sum_{j=-J}^{J} \alpha_j \mathbf{W}[p^{\epsilon,K}_j, \tilde{q}] \;\Big|\; \sum_{j=-J}^{J} \alpha_j = 1 \Big\}\,, \tag{4.21}$$

for some $\alpha \in \mathbb{R}^{2J}$ and set the requested approximate projection

$$\mathcal{P}^{\epsilon,K}[s] = \mathbf{I}^K(\bar{s}^K, \mathcal{P}^{\epsilon,K}_{loc}[s^{\epsilon,K}], 1/\rho^\epsilon)\,. \tag{4.22}$$

Again as shown in [129] we obtain a rigorous relation of this nonlinear projection on the local submanifold $\mathcal{M}^{\epsilon,K,J}$ and the classical Riemannian projection, i.e. we obtain

$$\mathcal{P}^{\epsilon,K}[s] \to \mathcal{P}[s]$$

for $\epsilon \to 0$ and $K \to \infty$ (for instance with $\epsilon = N/K$ for a small $N \in \mathbb{N}$. Let us emphasize that the constrained optimisation problem incorporated in the projection $\mathcal{P}^{\epsilon,K}_{loc}$ does not require any treatment of tangent vectors and is built on the rigid body motion invariant functional $\mathbf{W}$.

Quasi Newton method and Lagrangian formulation. We solve the constrained optimisation problem Eq. (4.21) using a Quasi Newton method. As a reminder, s^ϵ is the local version of s to be reconstructed, and $q[\alpha]$ is the projection on the local manifold, represented as the weighted elastic average of principal variations p_j.

To this end, we define the objective functional

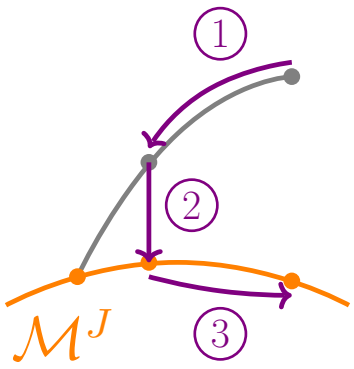
$$\mathcal{J}[\alpha] = \mathbf{W}[s^\epsilon, q[\alpha]]\,, \tag{4.23}$$

for $(\alpha_{-J}, \ldots, \alpha_{-1}, \alpha_1, \ldots, \alpha_J) \in \mathbb{R}^{2J}$ and $q = q[\alpha] \in \mathbb{R}^{3n}$ as the (locally unique) minimiser of

$$q \mapsto \mathcal{A}[\alpha, q] = \sum_{j=-J}^{J} \alpha_j \mathbf{W}[p^{\epsilon,K}_j, q]\,. \tag{4.24}$$

In order to apply a Quasi Newton scheme we have to evaluate the cost functional and its gradient $(\partial_{\alpha_j}\mathcal{J}[\alpha])_{j=-J,\dots,-1,1,\dots,J}$. To this end let us consider the Lagrangian formulation of the constrained optimisation problem Eq. (4.21).

Given



$$\mathcal{G} : \mathbb{R}^{2J} \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}\,, \tag{4.25}$$
$$\mathcal{G}[\alpha, q] := \partial_q \mathcal{A}[\alpha, q] = \sum_{j=-J}^{J} \alpha_j \, \mathbf{W}_{,2}[p_j^{\epsilon,K}, q]\,,$$

the Euler–Lagrange condition on q is given by $\mathcal{G}[\alpha, q] = 0$ for fixed α (with $\alpha_0 = 0$). Hence we define the Lagrangian

$$\mathcal{L} : \mathbb{R}^{3n} \times \mathbb{R}^{2J} \times \mathbb{R}^{3n} \to \mathbb{R} \tag{4.26}$$
$$\mathcal{L}[q, \alpha; \mu] := \mathbf{W}[s^\epsilon, q] + \mathcal{G}[\alpha, q] \cdot \mu$$

where μ is the associated vector of Lagrange multipliers. The solution of Eq. (4.21) is linked to a saddle point of Eq. (4.26) and thus to the nonlinear system of equation

$$0 = D_{(q,\alpha,\mu)}\mathcal{L}[q, \alpha; \mu]\,, \tag{4.27}$$

which leads to

$$0 = D_q\mathcal{L}[q, \alpha; \mu] = \mathbf{W}_{,2}[s^\epsilon, q] + \sum_{j=-J}^{J} \alpha_j \mathbf{W}_{,22}[p_j^{\epsilon,K}, q] \cdot \mu\,, \tag{4.28}$$
$$0 = D_{\alpha_j}\mathcal{L}[q, \alpha; \mu] = \mathbf{W}_{,2}[p_j^{\epsilon,K}, q] \cdot \mu\,,\ j = 1, \dots, J\,, \tag{4.29}$$
$$0 = D_\mu\mathcal{L}[q, \alpha; \mu] = \sum_{j=-J}^{J} \alpha_j \mathbf{W}_{,2}[p_j^{\epsilon,K}, q]\,, \tag{4.30}$$

where $\mathbf{W}_{,22}$ denotes the Hessian with respect to the second argument. Here, Eq. (4.28) defines the dual variable μ, Eq. (4.30) coincides with Euler–Lagrange condition of the constraint Eq. (4.25), and by a classical result of constrained optimisation the right hand side of Eq. (4.29) returns the derivatives of the cost functional $\mathcal{J}$ with respect to actual degrees of freedom α_j. Thus, to evaluate the derivative of the cost functional we first solve the nonlinear equation Eq. (4.30) for q, the linear equation Eq. (4.28) for μ and then apply Eq. (4.29) to obtain

$$\partial_{\alpha_j}\mathcal{J}[\alpha] = \mathbf{W}_{,2}[p_j^{\epsilon,K}, q] \cdot \mu\,.$$

Alternatively, one could directly solve Eq. (4.27) by means of Newton's method with the drawback that one has to compute third derivatives of $\mathbf{W}$.

Due to the proper scaling with ρ^ϵ in the definition of $s^{\epsilon,K}$, the local projection in Eq. (4.20) is supposed to lie in the convex Riemannian polyhedron with vertices $(p_j^{\epsilon,K})_j$. Hence we have $\alpha_j \geq 0$ in practice which renders the minimisation of Eq. (4.24) numerically feasible.

4.4 Experiments

4.4.1 Datasets

To evaluate our proposed method, three datasets of different objects are used. The first one is high resolution "cat" shapes undergoing non-rigid deformations (10 poses in total) taken from [4]. Although cat data exhibit strong non-linear deformations, there might not exist much articulations as seen in human body shapes. Therefore, we look at human subject performing various poses and use totally 71 poses from [3] to study articulation movements at a very high level. Moreover, another 30 high quality human alignments taken from [11] is used for pose reconstruction purpose. Note that all the shapes used in our experiments are in dense correspondence as provided and have the same mesh topology across different poses.

4.4.2 Principal variations

Two time-discrete shell PGA models (Fréchet mean and first five principal variations for $K = 4$) learned on the Cats [4] and the SCAPE dataset[3], respectively, are shown in Fig. 4.11. The principal variations are visualised by using the geodesic interpolation operator $\mathbf{I}(\bar{s}, p_j, \pm t)$ with $t = 3$ to sample along the one dimensional principal geodesic and overlay the resulting shapes with Fréchet mean.

In Fig. 4.12, the model compactness as a function of the number of retained modes for these two models and the one learned on Dyna [11] are shown.

4.4.3 Shape reconstruction

Several qualitative examples of submanifold projection are shown in Fig. 4.13. The input shape (gray) is projected onto the submanifold obtained by building a discrete PGA model using the Dyna dataset. By varying the model dimensionality over $J = 5, 11, 17$, reconstructed shapes are shown in yellow. The subtleties of the shape are correctly recovered as J increases, yielding a smooth residual energy.

We evaluate the generalisation ability of our model in Fig. 4.15. We compare against [6] with 60 dimensions retained, the data-driven approach of [5] using all training shapes and the Shell

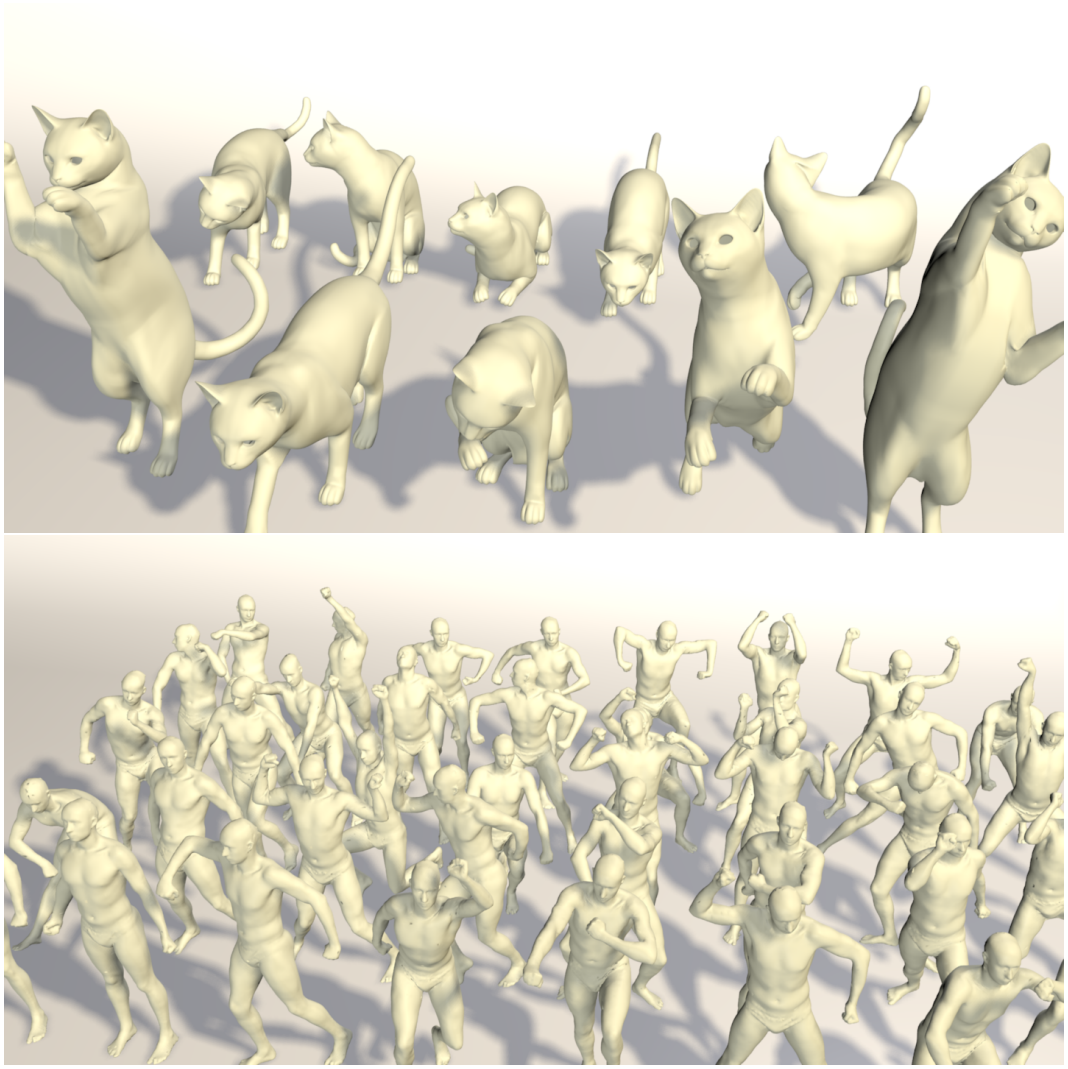


Figure 4.10: TOSCA cats [4] and SCAPE [3] training data.

Figure 4.11: Time-discrete PGA models built on TOSCA cats [4] and SCAPE [3]. Mean shape (orange) and first five principal variations (green).

PCA model (Chapter 3). Using only 20 dimensions, our model generalises almost as well as [5] and outperforms the other two models substantially.

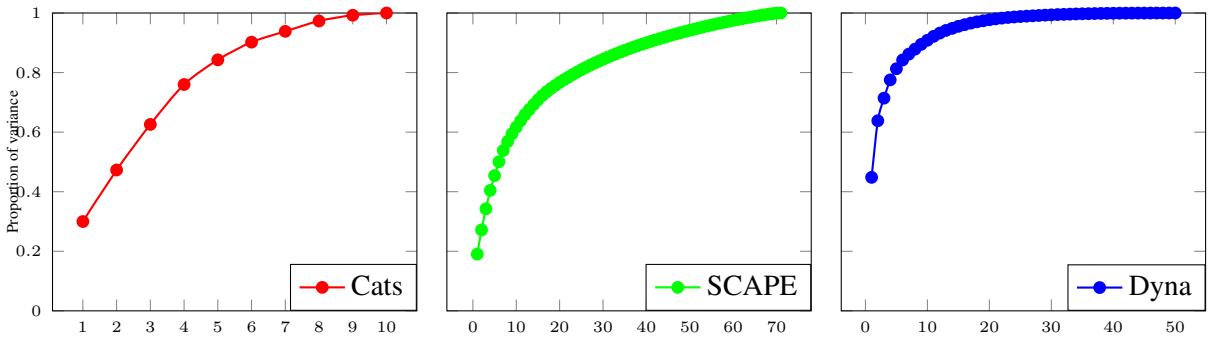


Figure 4.12: Model compactness with respect to $\mathbf{W}$ for models built on the TOSCA cats, the SCAPE dataset and a subset of Dyna dataset. Number of retained model dimensions on x axis, proportion of variance captured on y axis.

	Dataset	n / J	Offline	Online
Fig. 4.11	SCAPE	71 / -	73 m	-
	TOSCA_Cat	10 / -	66 m	-
Fig. 4.13	Dyna_50009	29 / 10	21 m	70 s
Fig. 4.15	SCAPE	70 / 20	72 m	232 s

Table 4.1: Timings for model building and shape reconstruction using fixed $K = 4$, but different numbers of training shapes n and principal variations J.

4.5 Summary

In this chapter, we have shown how to perform principal geodesic analysis in the space of discrete shells. In doing so, we derived an alternate formulation of PGA that avoids performing any operations in the tangent space and works directly with objects lying on the manifold. The whole approach is based on an elastic energy functional measuring membrane and bending distortion, so it is promising to interpolate and extrapolate between sparse sample shapes in a physically-meaningful way (Challenge 3). The result is a physically-guided statistical shape model, that is able to generalise across datasets containing large nonlinear articulations and deformations (Challenge 2). The central tool - the projection onto a submanifold of discrete shells - is well suited as the key ingredient in mesh editing or model fitting. Most importantly, the whole framework does not require any alignment step to get rid of rigid body motion which is notorious in shape modelling (Challenge 4).

In comparison to the original PGA model [79], which deals with a low dimensional medial axis description, we consider high dimensional shape manifolds. Furthermore, we extend PGA to the time-discrete setting and introduce a rigid body motion invariant distance measure. This invariance is also a substantial advantage over the Shell PCA model proposed in the previous chapter, which is based on vertex displacement and hence alignment-dependent. To this end, Shell PCA model only allows for small deformations, i.e. mesh editing and motion tracking

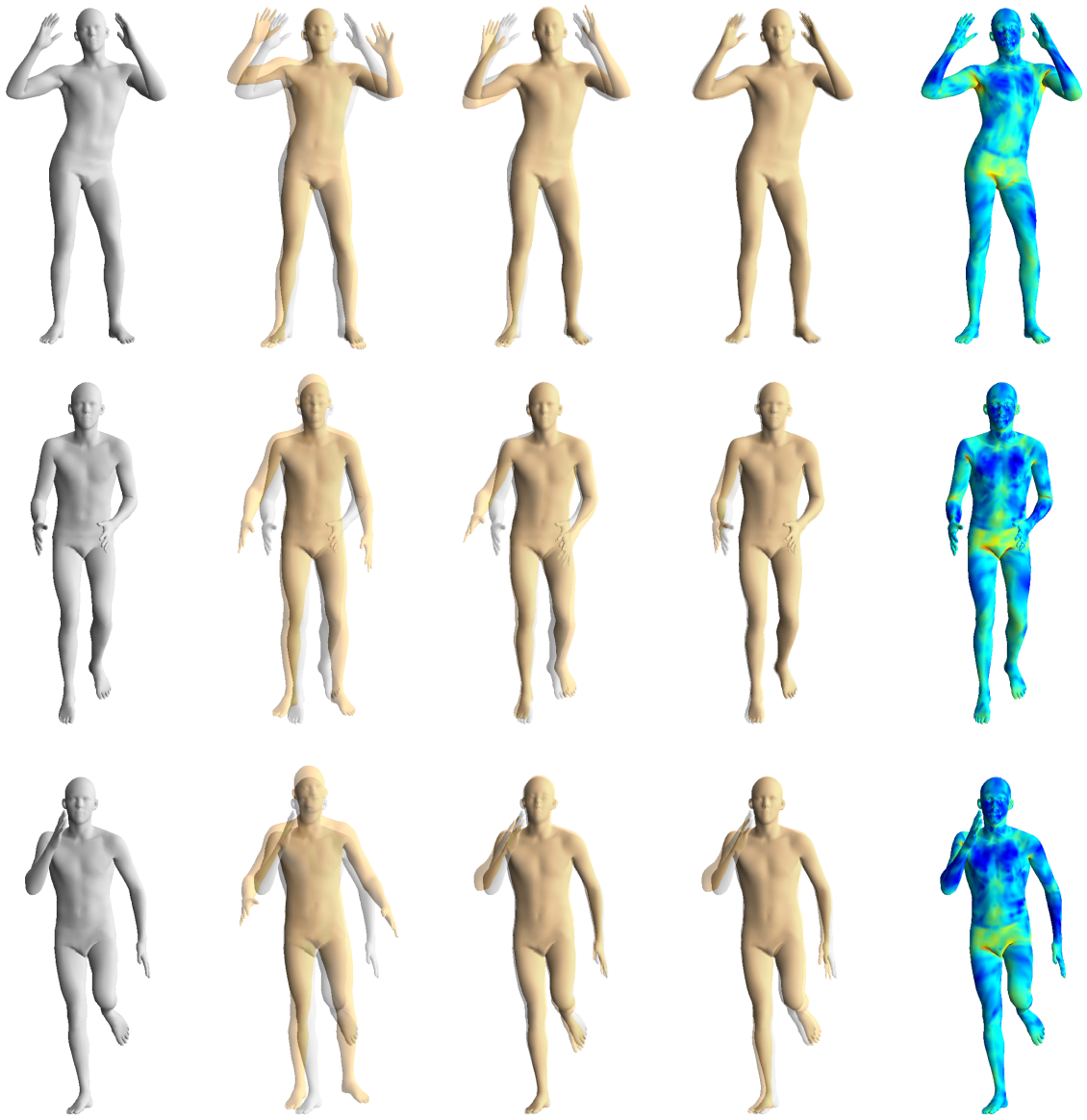


Figure 4.13: Qualitative visualisation of input shape (gray) projected onto model with (cols 2-4) $J = 5, 11, 17$ dimensions. Col 5 shows residual energy of projection with $J = 17$.

applications are out of reach of this purely elastic PCA approach.

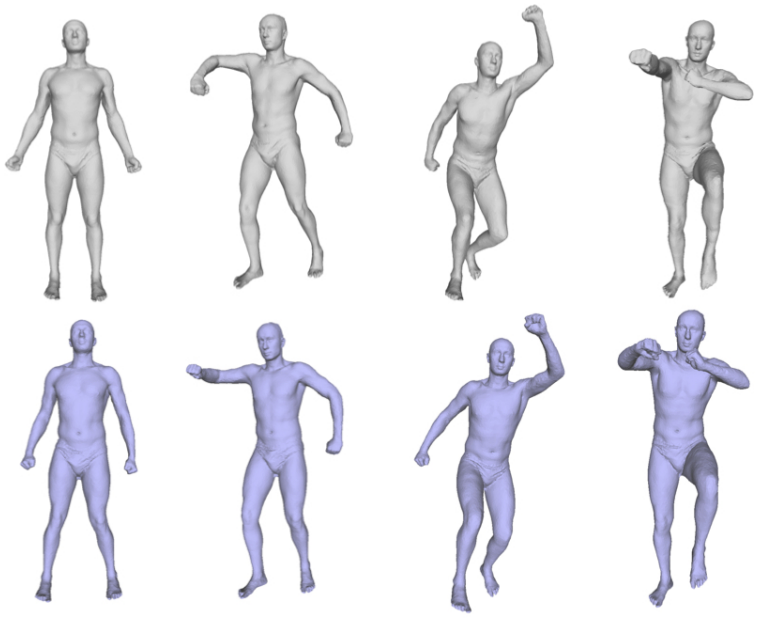


Figure 4.14: Qualitative reconstructions of input shapes from SCAPE [3]. Top: ground truth; bottom: reconstructions using $J = N - 1 = 70$.

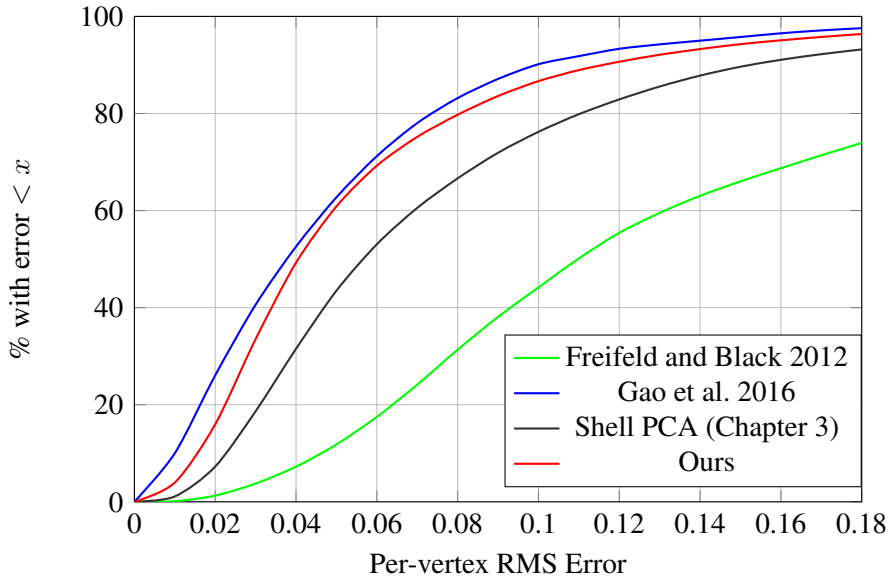


Figure 4.15: Leave-one-out evaluation of generalisation error on the SCAPE data set compared to [5] (using all shapes), Lie body [6] (60 dimensions) and Shell PCA (Chapter 3).

CHAPTER 5

GROUPWISE SHAPE CORRESPONDENCE VIA FUNCTIONAL MAPS

5.1 Introduction

Computing dense correspondence for shapes in a collection (represented as discrete 3D meshes) is a fundamental problem in computer vision research. It arises in a number of applications including statistical shape modelling [24], face morphing [130], motion capture [131], performance driven animation [132] and face transfer [97]. In essence, dense correspondence allows a collection of 3D shapes to be treated as vectors, and facilitates subsequent analysis such as Principal Component Analysis.

There are two distinct classes of dense correspondence problem. The first version of the problem (which we term *within-subject correspondence*) is to compute correspondence between scans of the same subject. This is a non-rigid registration problem which, in principle, has a well defined correct solution. Given enough information, it should be possible to uniquely describe a local region of one scan and find its corresponding region in another. A special case (and more constrained version) of this problem is where the scans come from a motion sequence where the shape non-rigidly deforms over time. In this context, temporal consistency means that dense correspondence can be viewed as a tracking problem.

The second version of the problem (which we term *between-subject correspondence*) is computing dense correspondence between scans of different subjects (i.e. faces with different identities). This is a much harder problem and arguably not well defined. In general, correspondence is a hypothesis of equivalence and defining the objective of the correspondence requires a definition of equivalence. Defining a meaningful notion of equivalence may only be possible in a sparse or low frequency sense. For example, sparse landmark points can be identified across different shapes [133] or it may be meaningful to talk of correspondence between parts or segments [134]. In this case, the correspondence in the remaining regions is interpolated (either

explicitly or implicitly). An alternate view is to impose some external desirable criterion on the correspondence. For example, we may require that the correspondence is smooth [24] or that it is optimal with respect to an information theoretic measure (e.g. minimum description length [64]).

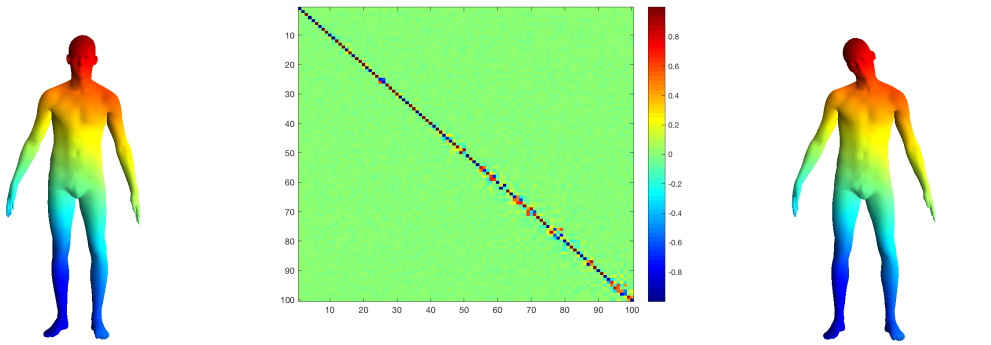
In both within-subject and between-subject correspondence, another important distinction is between *pairwise* and *groupwise* methods. Pairwise methods compute correspondence between each shape in the collection and a reference shape. This includes all template-based methods. To solve the problem in a different way, groupwise methods explicitly optimise such an objective function that measures the quality of the correspondences across the whole set of shapes simultaneously. The advantage of this is that the result is not dependent on a choice of reference shape or the order in which samples are processed. Furthermore, groupwise information can help resolve ambiguities that would be present in pairwise correspondence. For a long time, groupwise approaches to computing correspondence have had limited practical application. This is because the size of the problem space grows very rapidly with the number of samples in the set, leading to a very high dimensional nonlinear optimisation problem.

A recent paradigm shift in non-rigid shape analysis is based on the notion of "functional maps". The idea is to correspond real-valued functions on the mesh rather than points on the mesh directly [46]. A functional map can be converted to a point-to-point correspondence and they have recently been shown to perform very well for point-to-point shape matching [52, 53]. In this chapter, we pick up functional maps as representation to solve the problem of dense correspondence for 3D shapes. Specifically, we propose a groupwise variant of functional maps. The functional map representation overcomes the problem of the computational expense of groupwise methods in two ways. 1). functional maps are of much lower dimensionality than the mesh geometry themselves. We show in our experiments that functional maps of dimension as low as 30 are enough for high quality correspondence between face meshes containing 250k vertices. 2). functional maps can be composed meaning that, in our approach, only a minimal subset of maps need to be optimised with the remainder maps being constructed by compositions of these minimal subset of maps.

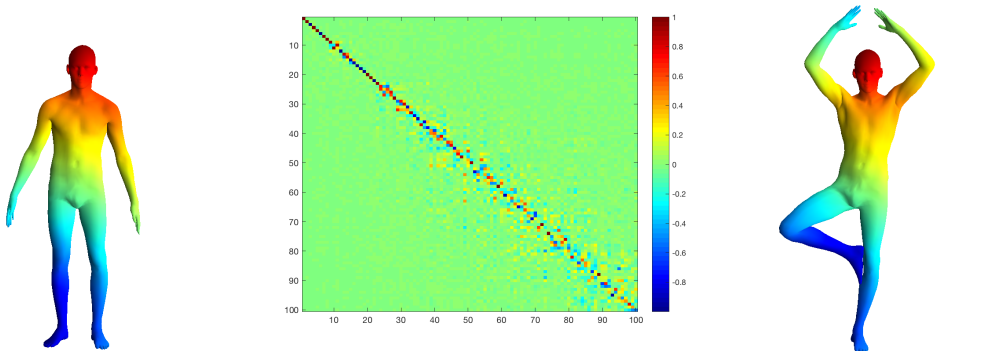
We apply our method to both between-subject and within-subject correspondence problems. This includes high resolution, high quality static facial expression scans and, large non-linearly deformed general objects, such as human bodies.

5.2 Review of functional maps

The eigenfunctions of the Laplace-Beltrami operator (manifold harmanics) of a 3D shape play the role of the Fourier basis in the Euclidean space [135, 136]. Several papers have studied consistent discretization of the Laplace-Beltrami operator [137, 138]. Taubin [135] made the analogy between the classical signal processing theory and manifold harmonics, showing that



(a) source (tr_reg_000) (b) ground-truth map (c) target (tr_reg_005)



(d) source (tr_reg_000) (e) ground-truth map (f) target (tr_reg_007)

Figure 5.1: Sample ground-truth functional maps between meshes on FAUST [7]. Maps are computed using $n = 100$ eigenfunctions of Laplace-Beltrami operator. Top row shows a map between a pair of shapes with small deformation, while bottom row shows a more deformed case.

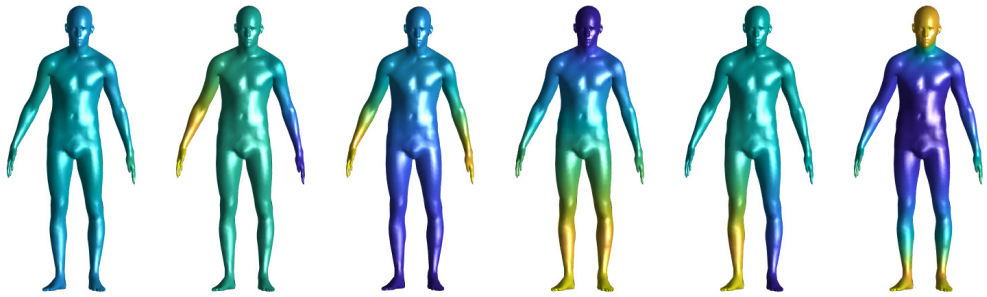


Figure 5.2: First 6 Laplace-Beltrami basis functions computed on a human body shape.

standard tools such as analysis and synthesis of signals in signal processing can be carried out on manifolds.

Let $T : M \to N$ be a bijective mapping between manifolds M and N. Given a scalar

function on M, $f : M \to \mathbb{R}$, we obtain a corresponding function on N, $g : N \to \mathbb{R}$, by composition $g = f \circ T^{-1}$. The induced transformation $T_F : \mathcal{F}(M, \mathbb{R}) \to \mathcal{F}(N, \mathbb{R})$ is the functional representation of the mapping T, where $\mathcal{F}(\cdot, \mathbb{R})$ is a generic space of real valued functions. We remark that knowledge of T_F is equivalent to knowledge of T. And while T may be a complicated mapping between surfaces, T_F acts linearly between function spaces.

Let $\{\phi_i^M\}$ and $\{\phi_i^N\}$ be bases for $\mathcal{F}(M, \mathbb{R})$ and $\mathcal{F}(N, \mathbb{R})$ respectively (in the discrete case, these are basis vectors as shown in Fig. 5.2). We follow Ovsjanikov et al. [46] and use as our basis the eigenfunctions of the Laplace-Beltrami operator. Any function $f : M \to \mathbb{R}$ can be approximated as a linear combination of basis functions $f = \sum_i^n a_i \phi_i^M$, where we use the first n Laplace-Beltrami eigenfunctions as the basis. The functional mapping T_F can be expressed in terms of these bases as:

$$T_F \left(\sum_i a_i \phi_i^M \right) = \sum_j \sum_i a_i c_{ij} \phi_i^N, \tag{5.1}$$

where $\mathbf{C} = c_{ij}$ is a possibly infinite matrix of real coefficients.

Given a pair of functions $f : M \to \mathbb{R}$ and $g : N \to \mathbb{R}$, the correspondence between f and g can be written simply as $\mathbf{Ca} = \mathbf{b}$ with $\mathbf{C}$ being the functional representation of the map, $\mathbf{a}$ and $\mathbf{b}$ the representation of f and g in the chosen bases of M and N. When $\mathbf{C}$ is unknown but a number of corresponding functions on M and N are provided, $\mathbf{C}$ can be found given enough constraints of type $\mathbf{Ca}_i = \mathbf{b}_i$. This is called function preservation constraint.

5.3 Groupwise functional maps

In this section, we replace pairwise map inference with a groupwise objective function that measures the quality of functional correspondence across the whole set of shapes. We show how to optimise all maps simultaneously whilst guaranteeing inversion and transitivity constraints are satisfied. A variant which enforces orthonormality of the maps as a hard constraint is presented. Furthermore, a groupwise refinement step applying ICP in the embedded space is shown to improve the correspondence quality, in particular, in terms of point-to-point correspondence.

5.3.1 Groupwise optimisation

Now consider a collection of shapes, i.e. a set of surfaces M_i $(i \in 1 \dots m)$. If we compute pairwise functional maps between all pairs of shapes, we ignore the context provided by the collection as a whole. Previously, this has been addressed by post-processing the pairwise maps to encourage cycles of compositions of maps to be close to the identity. Instead, we replace pairwise map inference with a groupwise objective function that measures the quality of the functional maps across the whole set of shapes simultaneously. We show how to optimise all

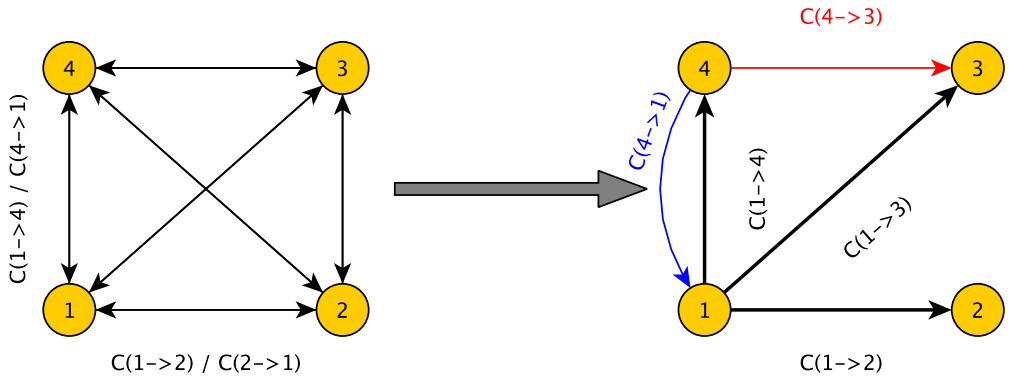


Figure 5.3: Groupwise optimisation of functional maps are done on a minimal set. Initial maps are edges in black on the right side, and unknown maps (in blue or red) could be obtained by either using *map inverse* or *map composition*.

maps simultaneously whilst guaranteeing inversion and transitivity constraints are satisfied (i.e. any cycle of maps will be guaranteed to be equal to the identity). Moreover, we describe a variant which enforces orthonormality of the maps as a hard constraint.

We denote by $\mathbf{C}_{j \leftarrow i}$ (or $\mathbf{C}_{ji}$ for short) the functional map from shape i to j. We denote by $\mathbf{a}_{ip}$ the low-rank representation of the pth function in the chosen basis of M_i with $i \in 1 \dots s$ and $p \in 1 \dots t$. The set of unknown functional map matrices is $\mathcal{C} = \{\mathbf{C}_{ij} | i, j \in [1, \dots, m] \wedge i \neq j\}$.

The groupwise objective function is as follows:

$$\varepsilon(\mathcal{C}) = \sum_i \sum_j \sum_p \|\mathbf{C}_{ij}\mathbf{a}_{jp} - \mathbf{a}_{ip}\|_F^2 \tag{5.2}$$

This is quadratic in the unknown functional map matrices. It could be solved using linear least squares with independent matrix variables. However, in order to strictly satisfy the constraint that any cycle of compositions of maps is identity, we need to impose two constraints between maps leading to a nonlinear optimisation problem.

First, we require that maps in either direction between a pair of views are the inverse of each other:

$$\forall i, j \in [1..s], \mathbf{C}_{ij} = \mathbf{C}_{ji}^{-1}. \tag{5.3}$$

Second, they are subject to a transitivity constraint:

$$\forall i, j, k \in [1..s], \mathbf{C}_{ki} = \mathbf{C}_{kj}\mathbf{C}_{ji} \tag{5.4}$$

This ensures that all 3-cycles are identity and, by construction, all n-cycles are also identity. See Fig. 5.3 for an illustration.

If we optimise over all transformations and explicitly enforce these two constraints, then this leads to a constrained optimisation with quadratic equality constraints, i.e. a non-convex

QCQP optimisation problem. We propose instead to optimise over a minimal subset of the transformations which allows us to express the problem as an unconstrained optimisation yet to guarantee that all constraints are satisfied.

We optimise $\mathbf{C}_{k1}$ for all $k > 1$, where shape 1 is an arbitrarily chosen reference shape. We emphasise that the solution is independent of the chosen reference shape. In order to evaluate the objective function in (5.2), we can compute the map between any pair of shapes i and j, in terms of maps from shape 1 by using the construction:

$$\mathbf{C}_{ji} = \mathbf{C}_{j1}\mathbf{C}_{1i} = \mathbf{C}_{j1}\mathbf{C}_{i1}^{-1}. \tag{5.5}$$

Hence, we can construct the set of maps $\mathcal{C} = \{\mathbf{C}_{ij} | i, j \in [1, \ldots, m] \wedge i \neq j\}$ from the reduced set $\mathcal{R} = \{\mathbf{C}_{k1} | k \in [2, \ldots, m]\}$ by applying the appropriate construction from (5.5) which we denote $\mathcal{C}(\mathcal{R})$. In doing so, we guarantee that the constraints in (5.3) and (5.4) are satisfied.

5.3.2 Hard orthonormality constraint

The Stiefel manifold $V_k(\mathbb{R}^n)$ is the set of all orthonormal k-frames in $\mathbb{R}^n$, i.e. the set of all $n \times k$ orthonormal matrices:

$$V_k(\mathbb{R}^n) = \{\mathbf{X} \in \mathbb{R}^{n \times k} | \mathbf{X}^T\mathbf{X} = \mathbf{I}_k\}. \tag{5.6}$$

If we require our functional map matrices to be orthonormal then $\mathbf{C}_{ij} \in V_t(\mathbb{R}^t)$, $\forall i, j \in [1..s]$.

We can use the same optimisation strategy as in the previous section. Since the product of two orthogonal matrices is still an orthogonal one, the construction in (5.5) ensures that any $\mathbf{C}_{ij}$ will be orthonormal, so long as $\mathbf{C}_{k1}$ for all $k > 1$ is orthonormal. There is also an efficiency saving in evaluating the objective function and its gradient since the matrix inverse can be replaced with a matrix transposition. In order to guarantee that $\mathbf{C}_{k1}$ for all $k > 1$ are orthonormal, we perform optimisation over a product of $m - 1$ Stiefel manifolds:

$$\underset{\mathcal{R} \in \prod_{i=1}^{s-1} V_t(\mathbb{R}^t)}{\arg\min} \quad \varepsilon(\mathcal{C}(\mathcal{R})). \tag{5.7}$$

The objective function could be written as :

$$\varepsilon = \sum_{i,j} \|\mathbf{C}_{ij}\mathbf{P}_j - \mathbf{P}_i\|_F^2 + \alpha \sum_{i,j} \|\mathbf{Q}_j\mathbf{C}_{ij} - \mathbf{C}_{ij}\mathbf{Q}_i\|_F^2 \tag{5.8}$$

where the first term encodes the functional preservation constraint with $\mathbf{P}_i$ the corresponding functions in shape i, and the second term enforces operator commutativity constraint with $\mathbf{Q}_i$ the functional representation of Laplacian operator. The weight α is to balance between the two terms. $\mathbf{C}_{ij} = \mathbf{C}_{i1}\mathbf{C}_{j1}^T$ is the matrix representation of functional maps from shape j to i, constructed from the minimal set of optimisation variables $\mathbf{C}_{k1}$, for all $k > 1$.

Algorithm 4 Groupwise functional maps optimisation

Input: a set of m triangular meshes $\{M_1, \ldots, M_m\}$, dimension of functional map $n \in \mathbb{N}$, number of vertex N
Output: functional maps matrices $\mathbf{C}_{ji} \in \mathbb{R}^{n \times n}$, where $1 \leq i, j \leq m$
// compute Laplacian-Beltrami basis for each shape
for $i := 1$ to m **do**
    $\Phi^{M_i} = get\_LB\_basis(M_i, n)$
end for
// compute shape functions $D^{M_i} \in \mathbb{R}^{p \times \mathbb{N}}$, and project on basis
for $i := 1$ to m **do**
    $D^{M_i} = get\_feature\_functions(M_i)$
    $P^{M_i} = project\_LB\_basis(D^{M_i}, \Phi^{M_i})$
    $Q^{M_i} = get\_diag\_evals(M_i)$
end for
// initialise pairwise functional maps with a chosen reference shape
$\mathbf{C}_{i1} = get\_pair\_map(P^{M_i}, P^{M_1})$
// optimise groupwise functional maps using Manopt
$\mathbf{C}_{ij} = get\_groupwise\_map(\mathbf{C}_{i1}, P^{M_i}, Q^{M_i})$ using (Eq. 5.8), $1 \leq i, j \leq m$

5.3.3 Groupwise refinement

There exists a natural transformation between functional maps and point-to-point map. Therefore, one can consider improving a given functional map with the help of its associated point-to-point map. Ovsjanikov et al. [46] use this observation to propose an iterative refinement scheme to encourage an estimated functional map to be closer to a point-to-point map. The idea is to move from computing functional map matrices based on coefficient vectors representing functions on the surfaces to computing them directly from point correspondences (provided by nearest neighbour matching in the embedded functional space). This procedure can be expressed as classical Iterative Closest Point (ICP) algorithm, after $\mathbf{S}$ is determined via NN search:

$$\varepsilon_{icp}(\mathcal{C}) = \|\mathbf{C}\Phi^{M_j} - \mathbf{S}\Phi^{M_i}\|_{\mathrm{F}}^2 \tag{5.9}$$

This is an orthogonal Procrustes problem, which can be seen as a rigid alignment in the embedded space.

If a shape collection is being analysed and it is believed that there is a point-to-point correspondence between any pair of shapes then we can perform iterative refinement in a groupwise manner. We do so by iterating between computing nearest neighbour correspondences and updating the functional map matrices by optimising a groupwise objective function.

Groupwise functional map ICP:

1. $\forall i, j \in [1..s] \wedge i \neq j$, compute nearest neighbour matches between columns of $\mathbf{C}_{ij}\Phi^{M_j}$ and Φ^{M_i}. Represent matches using a selection matrix $\mathbf{S}_{ij}$ such that the nearest neighbour of a column of $\mathbf{C}_{ij}\Phi^{M_j}$ is found in the same column of $\mathbf{S}_{ij}\Phi^{M_i}$.

2. Recompute functional map matrices by solving the groupwise optimisation problem:

$$\arg\min_{\mathcal{R}} \varepsilon_{icp}(\mathcal{C}(\mathcal{R})) \tag{5.10}$$

   where the objective function is given by:

$$\varepsilon_{icp}(\mathcal{C}) = \sum_i \sum_j \|\mathbf{C}_{ij}\Phi^{M_j} - \mathbf{S}_{ij}\Phi^{M_i}\|_{\mathrm{F}}^2 \tag{5.11}$$

3. Return to 1 until convergence.

Note that in step 2, the optimisation can be initialised using the previous estimates of the functional map matrices (at first, these are the estimates using function preservation constraints).

5.3.4 Point-to-point correspondence

Ultimately, for applications such as face morphing, statistical modelling or motion capture, we require a dense point-to-point correspondence. Functional correspondence can be converted to a dense point-to-point correspondence in a straightforward manner. The naive approach is to pass an indicator function through the map and select the point with maximal value. Concretely, to find the point $y \in N$ that corresponds to $x \in M$, we chose y as the point at which $g(y)$ obtains a maximum, where $g = T_F(\delta_x)$ and δ_x the delta function around $x \in M$. This is computationally expensive. A more efficient alternative is to perform nearest neighbour (NN) matching in the embedded functional space.

Observe that the delta function δ_x has coefficients $a_i = \phi_i^M(x)$ in the basis of M. Hence, given a matrix $\mathbf{\Phi}^M \in \mathbb{R}^{t \times v}$, where v is the number of vertices, and a functional map matrix C, the image of all delta functions centred at points of M is given by $\mathbf{C}\mathbf{\Phi}^M$. Finding corresponding maxima amounts to performing NN matching between the columns of $\mathbf{C}\mathbf{\Phi}^M$ and $\mathbf{\Phi}^N$, i.e. in t-dimensional space.

This approach has been proposed in the work of Ovsjanikov et al. [46] and we use the same method for point-to-point maps for between-subject correspondence. We choose one of the meshes to provide the reference topology, apply NN matching between each mesh and the reference in the functional embedding space and use this correspondence to consistently remesh each input mesh. The drawback of this approach is that correspondence obtained will not be bijection. In other words, each point from one shape may not have a unique counterpart. In addition, matches found in terms of nearest neighbour search in the embedded space is not reliable.

5.3.5 Implementation and optimisation

Optimisation on a product manifold of Stiefel manifolds is implemented with the Manopt toolbox [139]. Specifically, groupwise cost objective function and the gradient w.r.t. the minimal set of matrix variables are provided. The trust-regions solver is used to solve the manifold optimisation problem. The weight α is set to be 1.0 in all cases. In our experiment, the optimisation problem involving two 30×30 matrices takes less than 50 seconds to converge on an Intel Core 3.40GHz. Note that the optimisation problem is non-convex since the Stiefel manifold is not a convex set. Hence, a good initialisation is important.

To initialise, we begin by solving the problem without orthogonality constraints, transform each map $\mathbf{C}_{k1}$ to its closest orthogonal matrix and then refine by performing manifold optimisation of the groupwise objective function. Again, an alternative unbiased initialisation would be to use identity matrices. A further alternative is to use each shape in turn as the reference shape and then take the solution which gives lowest error overall (in other words, use a number of different initialisations and pick the best result).

5.4 Choice of functions

Key to the functional map framework is the a set of functions that are assumed to be approximately preserved between shapes. Depending on the applications, it is quite flexible to choose the functions to be used. In this section, we first list several feasible candidate functions for general shape matching, and then take faces as an example to design some face-specific functions. Basically, each function can be seen to capture a different notion of equivalence between shapes. The functional map framework attempts to satisfy all of these notions via the function preservation constraint. In general, we aim to choose smooth functions that will be well approximated by the low frequency basis provided by the truncated eigenfunctions of the Laplace-Beltrami.

5.4.1 General functions

Prior work on functional maps has almost exclusively used functions derived from intrinsic shape properties. Many of these functions are appropriate for shape matching with deformations being known to be approximate isometries [4].

Curvature We use mean (g) and Gaussian (h) curvature and some functions derived from these intrinsic curvatures. Specifically, we follow [48] and use the logarithm of the absolute value of the mean (i) and Gaussian (j) curvature. Shape index [140] is a continuous characterisation of local surface shape and previous work [141] has shown that it is useful for face shape matching problems. Shape index is derived from the principal curvatures and is shown in (k).

Shape descriptor Following the original work of Ovsjanikov et al. [46], we also use the Wave Kernel Signature (l) and Heat Kernel Signature (m) to provide shape-dependent functions.

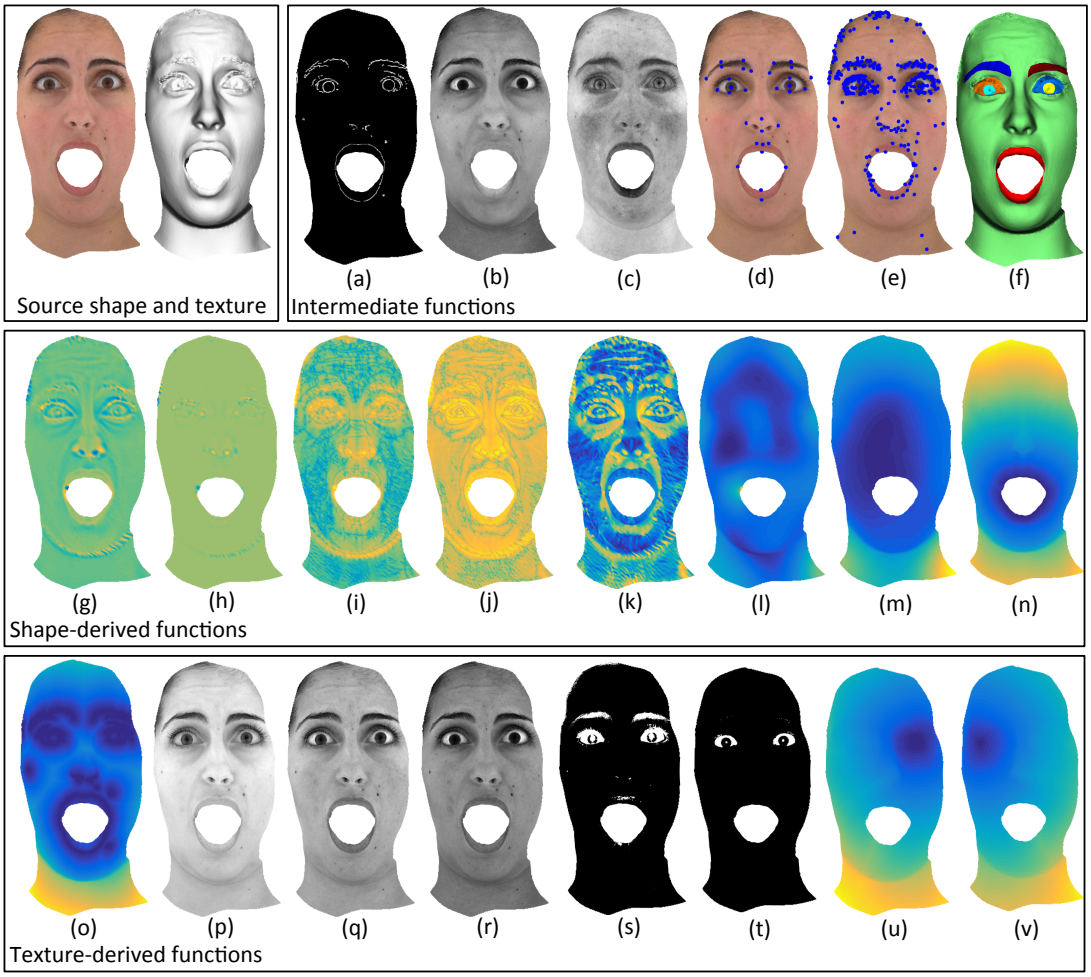


Figure 5.4: Candidate functions for face matching using functional maps framework. (See text for details)

By varying the time parameter, these signatures provide a sequence of functions.

Landmarks If we are given landmark point correspondences, we can translate this knowledge as functional constraints by considering distance functions to the landmarks or normally distributed functions around landmarks.

Segmentations Similarly to landmarks, correspondences between parts or segments can also induce functional correspondence in terms of distance or segment descriptors.

5.4.2 Face-specific functions

A prominent advantage of shape matching for faces over other shapes is the availability of high-quality texture maps. We are not aware of any previous attempts to use texture-derived functions for functional map estimation between meshes. For dense face correspondence, texture provides a rich source of intrinsic face properties and correspondence cues.

Texture and edges First, there is the raw texture itself, i.e. we use the raw RGB channels (p)-(r) as scalar functions on the mesh. For within-person correspondence, we expect this to

remain exactly constant apart from changes in appearance due to lighting. The 3DRFE dataset that we use was captured in a lightstage meaning that texture is diffuse albedo - an intrinsic property of the surface. Between-person, we do not expect texture to remain constant although there are still useful cues. For example, lips will be redder than the rest of the skin and the whites of the eyes have a consistent colour between people. Nevertheless, it may be more appropriate to preserve texture *edges* rather than texture itself. Intuitively, this will encourage features such as the lips and eyebrows to have consistent boundaries. To do so, we apply an edge detector to the texture in UV space and transfer the binary edges to mesh (a). To represent the edges using a smooth function, we apply a geodesic distance transform to the edges (o) so that each vertex is assigned a value corresponding to the geodesic distance to the closest texture edge vertex.

Facial segmentations In the same vein, texture provides a useful cue for segmenting a face into semantically meaningful regions (f). One robust way to do this for faces is to use a biophysically inspired colour transformation to identify non-skin regions. Tsumura et al. [142] have shown that Independent Components Analysis can be used to estimate melanin and haemoglobin maps from face textures. We show the two ICA channels in (b) and (c) corresponding to melanin and haemoglobin respectively. The haemoglobin map can be used to segment the lips. Further, by thresholding the relative error between the original colour values and their reconstruction using two ICA channels, we can compute binary segmentations that highlight eyebrows and eyes (i.e. non-skin regions). Another simple segmentation is threshold the image saturation which allows the white of the eyes to be located (t). Segmentations can either be used directly as indicator functions or transformed into smooth functions by computing geodesic distance maps. For example, (u) shows geodesic from the right iris segment.

Fiducial landmarks Perhaps the most powerful use for texture is in the accurate detection and matching of landmark points. Between-person, landmarks are fiducial points with anthropometric meaning (manually or automatically labelled). The automatic labelling of such points in images has recently received a lot of attention [133]. Applying a facial landmarking algorithm in UV texture space or to a rendered image of the textured mesh, provides fiducial points that can be mapped to the corresponding vertex (d). Since each landmark has the same meaning they are, by design, groupwise consistent. For within-person correspondence, landmarks can be provided by any local feature detector (e.g. SIFT). In (e), we show SIFT features detected in UV space and mapped to the mesh.

Boundaries During face shape capture, certain regions of the face may not be captured. For example, many capture methods do not accurately capture the interior of the mouth, interior of the nostrils or the highly specular eye surface. These regions may be missing from the scan or manually removed. The internal boundaries caused by these holes provides a useful cue for correspondence. For example, in (n) we show a function derived from the mouth boundary (geodesic distance from the closest mouth boundary point).

5.5 Experiments

5.5.1 Evaluation on Face Datasets

Evaluation of dense face correspondence is difficult. Since there is no meaningful source of "ground truth" correspondence, quantitative evaluation can only indirectly measure the quality of the correspondence. For this reason, we focus on qualitative evaluation, although we also provide a meaningful quantitative evaluation for within-person correspondence.

We use several datasets for our evaluation. For between-person correspondence we use the 10 out-of-sample meshes provided with the Basel Face Model [8]. These meshes have been put into dense correspondence using non-rigid ICP fitting of a template mesh, allowing us to qualitatively compare our correspondence to the original. We subsample the meshes to 6K vertices. For within-person correspondence we use a subset of the 3DRFE [7] dataset. These models were captured using a lightstage and are of very high quality. The resolution of each mesh is over 250K vertices and the texture is diffuse albedo, obtained using cross-polarised illumination. These meshes have genus 1 as the mouth has been manually removed. We choose 6 scans of a same person from BU3D [143] for a quantitative comparison. In all cases, we use functional maps of dimension 30×30.

Between-person correspondence. In order to evaluate between-person correspondence we visualise texture transfer results allowing qualitative assessment of the correspondence quality. In Figure 5.5 we compare texture transfer results for the pairwise version of our proposed method, the groupwise version and non-rigid ICP [9]. Non-rigid ICP introduces some artefacts around the eyes and eyebrows where the result lacks symmetry. The functional map results are improved in the eye regions but show some artefacts around the lips, though the groupwise variant slightly reduces these. We believe that the source of these errors is the reliance on nearest neighbour matching in the functional embedding space, particularly as the deformations are non-isometric in this case.

Within-person correspondence. We now evaluate within-person correspondence. Again, in Figure 5.6 we provide qualitative evaluation via texture transfer and provide comparison with non-rigid ICP [9]. Under large non-rigid deformations, non-rigid ICP introduces large artefacts in the eyebrows and lips. Groupwise functional maps with simple nearest neighbour matching improves in the eyebrow region but still shows problems in the lip region. Groupwise functional maps with our proposed point-to-point conversion using filtered feature matches provides a convincing texture transfer. Following [132] we can evaluate this correspondence quantitatively under the assumption that texture remains constant under expression changes. By measuring the variance of the colour at each vertex over the expression changes, we get a measure of correspondence quality. We expect the variance to be zero under perfect correspondence. In other words, a small value of the variance indicates a better correspondence. The variances

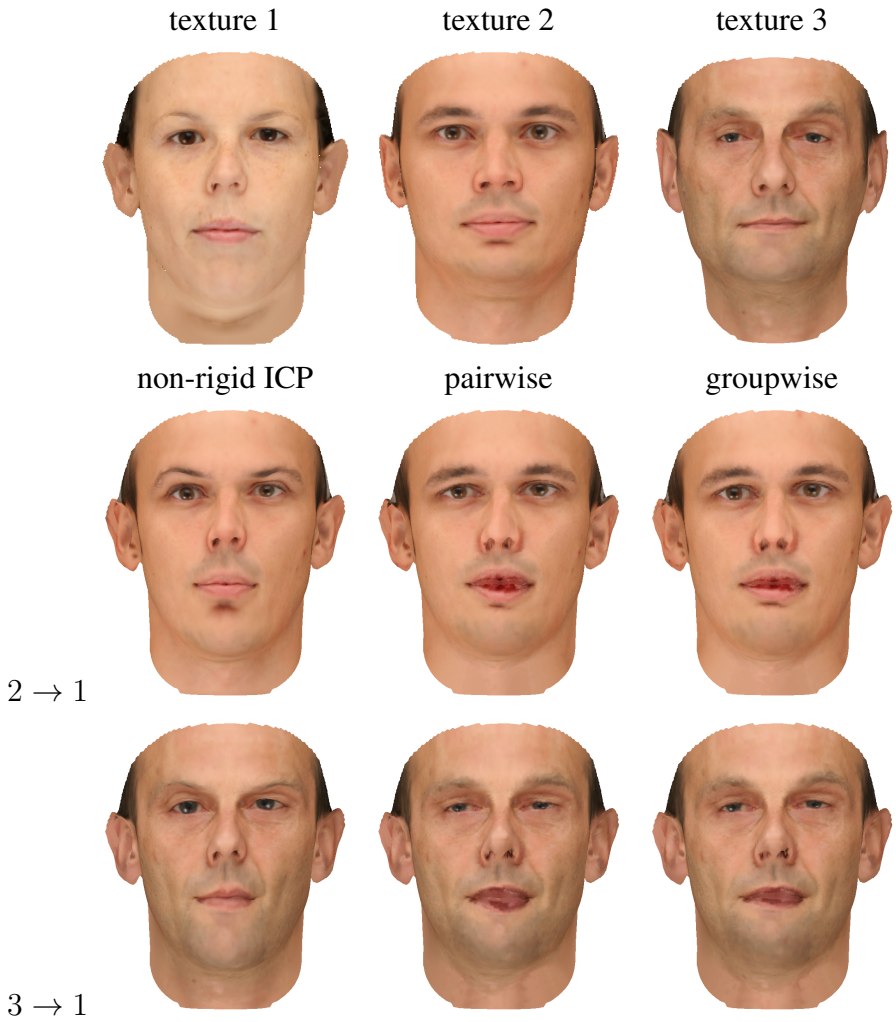


Figure 5.5: Texture transfer results on BFM meshes [8]. Row 1: original textures. Rows 2 and 3: textures transferred from shape 2 and 3 to shape 1 respectively. Col. 1: non-rigid ICP [9]; col. 2: pairwise functional maps; col. 3: groupwise functional maps.

averaged over colour channels and vertices are reported in Table 5.1. The functional map methods outperform non-rigid ICP, while requiring several orders of magnitude less running time. The qualitative improvement of our feature match warping method is evident again in the quantitative results.

Method	Texture consistency	Running time
Non-rigid ICP[9]	.0103	6 hours
Ours (with NN)	.0101	55 seconds
Ours (Sec.5.3.3)	.0099	120 seconds

Table 5.1: Texture consistency and running time on 3DRFE subset

Visualising functional maps error. We visualise the error of correspondence by using a SVD decomposition of the functional maps [144]. Three kinds of functional maps are compared: ground truth map, pairwise map and groupwise map. Only the first two modes are shown. Facial

Figure 5.6: Texture transfer results on 3DRFE [7]. Row 1: original textures. Rows 2 and 3: texture transfer results from shape 2 and 3 to shape 1 respectively. Col. 1: Non-rigid ICP [9]; col 2: groupwise functional map with nearest neighbour point-to-point; col 3. groupwise functional map with feature match warping.

regions undergoing large error are highlighted in the Fig. 5.7.

Comparing to pairwise map, the errors of groupwise variant are more local and more close to ground truth error in locations. Note, groupwise method is applied on shapes of multiple

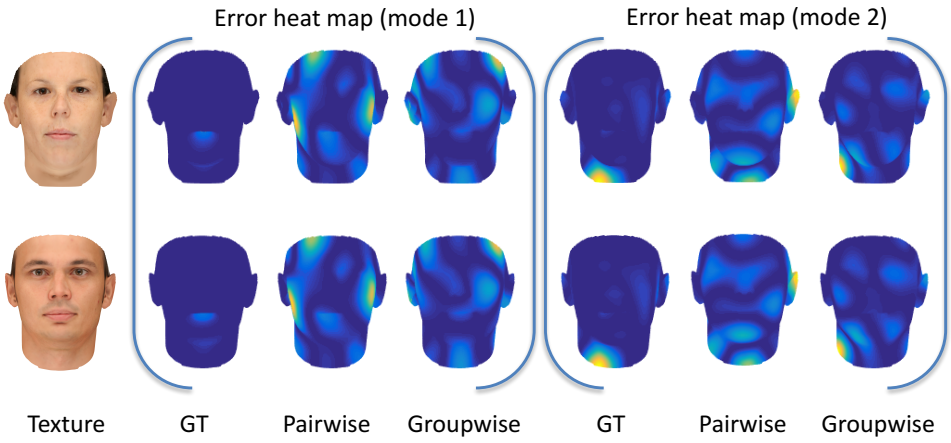


Figure 5.7: Visualising correspondence errors via SVD decomposition of the maps.

identities from [8], while only two of them used by the pairwise method are shown.

5.5.2 Evaluation on TOSCA dataset

In addition to shapes of human faces, our method can also be applied on general shapes such as TOSCA [4] dataset. TOSCA contains 80 meshes representing humans and animals in a variety of poses. The meshes appear in 9 categories with same topology. Per-vertex ground truth map for each class is provided. We show the performance of our method with comparisons to other methods in this section, both qualitatively and quantitatively.

Candidate descriptors We use both point descriptors and landmarks for the functional maps computation. These include: Wave Kernel Signature (step = 100), and Wave Kernel Map for landmarks (step = 130). For landmarks, we manually choose 3 landmarks for each class of objects.

Evaluation measures. We provide a quantitative evaluation of our method based on the predicted map, $f : M_1 \to M_2$ with respect to a "ground truth" map, $f_{true} : M_1 \to M_2$. We compute for every point, p on M_1 in the ground truth correspondence the geodesic distance, $d_{M_2}(f(p), f_{true}(p))$, between its image $f(p)$ and its true correspondence $f_{true}(p)$. We aggregate the geodesic distances into a error measure:

$$Err(f, f_{true}) = \sum_{p \in M_1} d_{M_2}(f(p), f_{true}(p))$$

where $d_{M_2}(f(p), f_{true}(p))$ is normalized by $\sqrt{(Area(M_2))}$

Distributions of errors are plotted as well to show the performance of different methods. X-axis represents a varying geodesic distance threshold, D, and the y-axis shows the average percentage of points for which smaller than D (or called correspondence percentage).

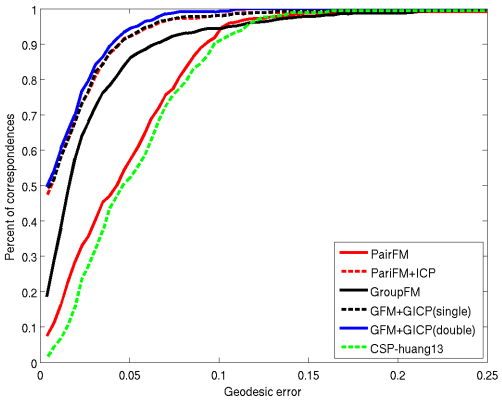
Quantitative results. We show the performance of correspondence mapping in 5.8 and

5.9. Our groupwise functional map outperforms pairwise functional map substantially. It also outperforms baseline method CSP [65] by a large margin. Instead of recovering dense correspondence directly, CSP first estimates the correspondence of a small number of sampled landmarks using Farthest Point Sampling, and then interpolated them to dense ones. This is why CSP does not perform well. Furthermore, when ICP refinement step is applied, both pairwise and groupwise variant gain performance improvement. In particular, double-directional ICP performs slightly better than single-directional ICP in groupwise refinement. Generally speaking, groupwise ICP has similar performance than pairwise ICP, except for the case "Michael".

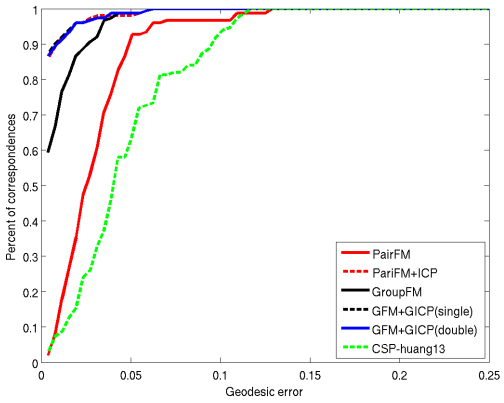
Qualitative results. In Figure 5.10, we show the dense correspondence (same color meaning correspondence) of several pairs of shapes from TOSCA. In each row, the source shape is shown at the beginning. From left to right, correspondence of the target shape found by Pairwise Functional Map ([46]), Groupwise Functional Map, and with additional ICP step are presented. Our method significantly improves over pairwise version of functional maps, although the difference of variants using ICP is very small.

5.6 Summary

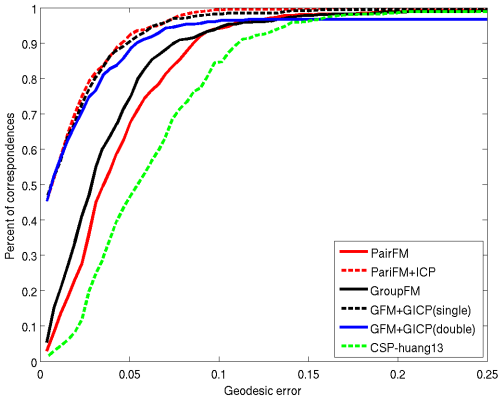
In this chapter, we developed a groupwise extension to the *functional maps* paradigm. The goal is to computing dense correspondence for a given set of 3D meshes. The method is particularly suitable for non-rigid shapes. The deformations between them are expected to be close to near-isometric ones, including facial expressions as an example. The map-based representation overcomes many drawbacks of classical point-based methods. For example, point-based representation easily leads to combinatorial NP-hard problem when the size of meshes increases. Also, point-to-point correspondence may not exist in many scenarios and hence soft or fuzzy correspondence were introduced. There are also many advantages when comparing groupwise to pairwise ones. First, pairwise methods only make sense when two similar shapes are considered. Second, results of pairwise methods are not consistent for multiple shapes. In all, the proposition of groupwise functional maps alleviate these limits. The method uses functional maps as a multi-scale, flexible, low-dimensional representation. The optimisation improves the correspondence accuracy because it's objective is groupwise. It also guarantees cycle-consistency by map-construction which is computational cheap. To solve the optimisation, Manopt toolbox is utilised. It also allows one to implement ICP-based refinement step easily within the same framework.



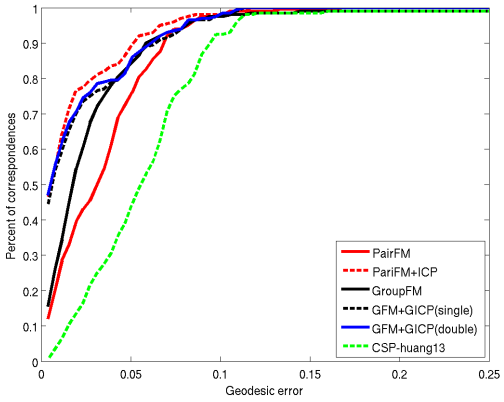
(a) Cat



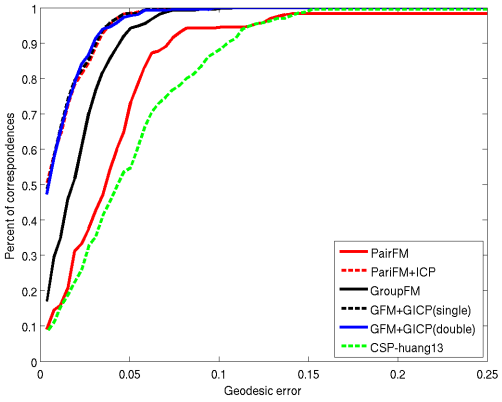
(b) Wolf



(c) Dog

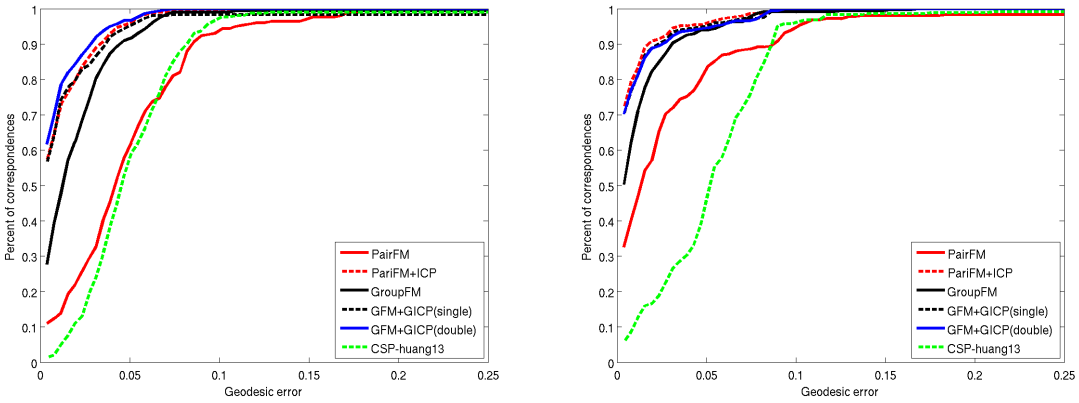


(d) Gorilla



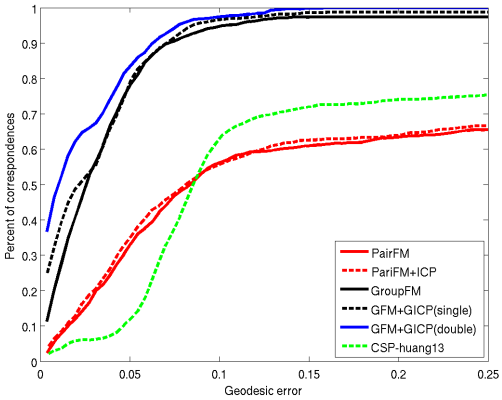
(e) Horse

Figure 5.8: Comparison between the proposed method and the baseline approaches on TOSCA (Part 1)

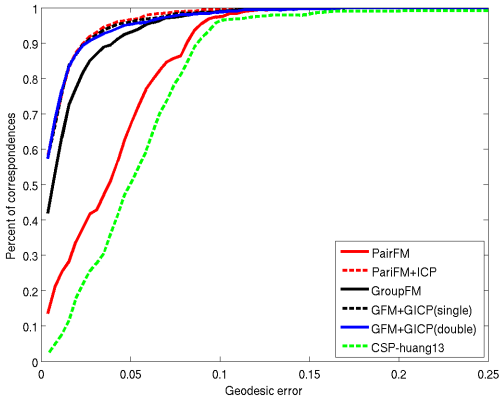


(a) Centaut

(b) David



(c) Michael



(d) Victoria

Figure 5.9: Comparison between the proposed method and the baseline approaches on TOSCA (Part 2)

Figure 5.10: Qualitative comparison between the proposed method and the baseline approaches on TOSCA

CHAPTER 6

APPLICATIONS OF STATISTICAL HUMAN BODY MODELS

6.1 Introduction

As a variety of statistical human body models presented high quality results in literature, many applications followed with successful performance in computer vision and computer graphics area. For example, motion capture using MoCap markers and mesh editing applications are highlighted in Sec. 6.2 and Sec. 6.3. Furthermore, we demonstrate a novel application of estimating detailed human body shape and pose from 3D scan sequences in Sec. 6.4. Note also that two different statistical models are used in this chapter: Shell PGA developed in Chapter 4 for Sec. 6.2 and Sec. 6.3, state-of-the-art vertex-based SMPL[26] model for Sec. 6.4.

6.2 Model Fitting via Soft Constraints

Marker-based motion capture (MoCap) is widely used to animate human characters in films and games. In this section, an application of fitting the Shell PGA model to motion capture data is presented by means of a soft penalty approach. In particular, this allows us to reconstruct a discrete shell from (potentially noisy) input data from a motion capture device. In this case, the input data is given as a vector of L sparse marker positions, i.e. $x = (x_\ell)_{\ell=1,\dots,L}$, corresponding to vertex positions $\mathcal{X}_\ell(s)$ on the mesh s. Knowing these correspondences, we measure the mismatch of some discrete shell $s \in \mathcal{M}$ and the given landmarks by $\mathcal{F}_x[s] = \sum_{\ell=1}^{L} \|\mathcal{X}_\ell(s) - x_\ell\|^2_{\mathbb{R}^3}$. Following Chapter 4, we consider shell deformation energy $\mathbf{W}[s, \mathcal{P}[s]]$ as a prior for the identification of a reconstructed discrete shell. Hence, we seek a minimizer s of the model fitting energy given by

$$\mathbf{W}[s, \mathcal{P}[s]] + \gamma \mathcal{F}_x[s] \tag{6.1}$$

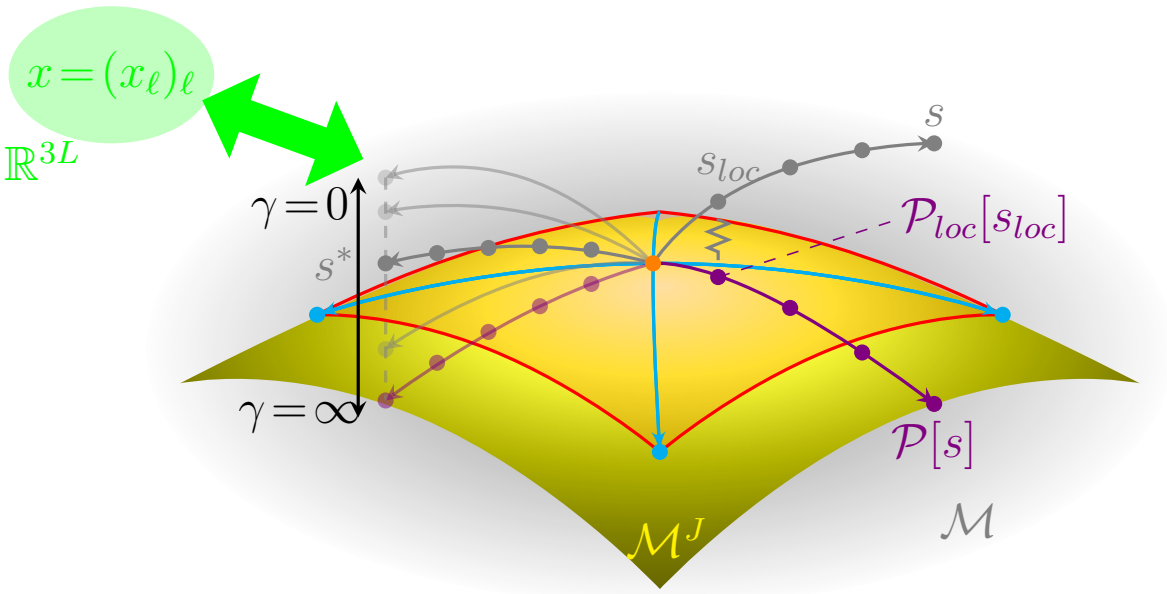


Figure 6.1: Right: Projection of an unseen shape s onto the model space $\mathcal{M}^J$: scale s to s_{loc}, project s_{loc} locally to $\mathcal{P}_{loc}[s_{loc}] \in \mathcal{C}^J$, and finally rescale to get $\mathcal{P}[s] \in \mathcal{M}^J$. Left: Model fitting of s^* driven by sparse landmarks $X \in \mathbb{R}^{3L}$ depending on fitting parameter $\gamma > 0$.

for some weight $\gamma > 0$, which controls the strength of markers fitting. In other words, a large value of γ will encourage exact matching of the markers and their correspondence of s. Please refer to Fig. 6.1 for the visual illustration of model fitting.

Algorithm 5 Motion capture using Shell PGA model

1: **Input:** MoCap markers $x^{f=1,\dots,F} = (x_\ell)_{\ell=1,\dots,L} \in \mathbb{R}^3$, average $\bar{s}$, principal variations $p_{-J,\dots,J}$, marker weight γ
2: **Output:** reconstructed dense shape $s^{f=1,\dots,F}$
3: initialise $\mathcal{P}[s^{f=1}]$ with average shape $\bar{s}$
4: // optimise s for frames $f = 2, \dots, F$
5: **for** $f := 2$ to F **do**
6: minimising $\mathbf{W}[s^f, \mathcal{P}[s^{f-1}]] + \gamma \mathcal{F}_x[s^f]$
7: update $\mathcal{P}[s^f]$ by projecting s^f on $p_{-J,\dots,J}$
8: re-estimate s^f using (Eq. 6.1)
9: **end for**

For the numerical solution of this problem, we make use of the following alternating scheme (based on the initial guess $\mathcal{P}[s] = \bar{s}$ or using temporal information): First, we minimise Eq. (6.1) in s for fixed $\mathcal{P}[s]$. If necessary, we re-compute $\mathcal{P}[s]$ (see Sec. 4.3.4) and go back to the first step. In our experiments this scheme quickly converges and only very few iterations already give very satisfactory fitting results. In practice, we use two iterations for the results shown.

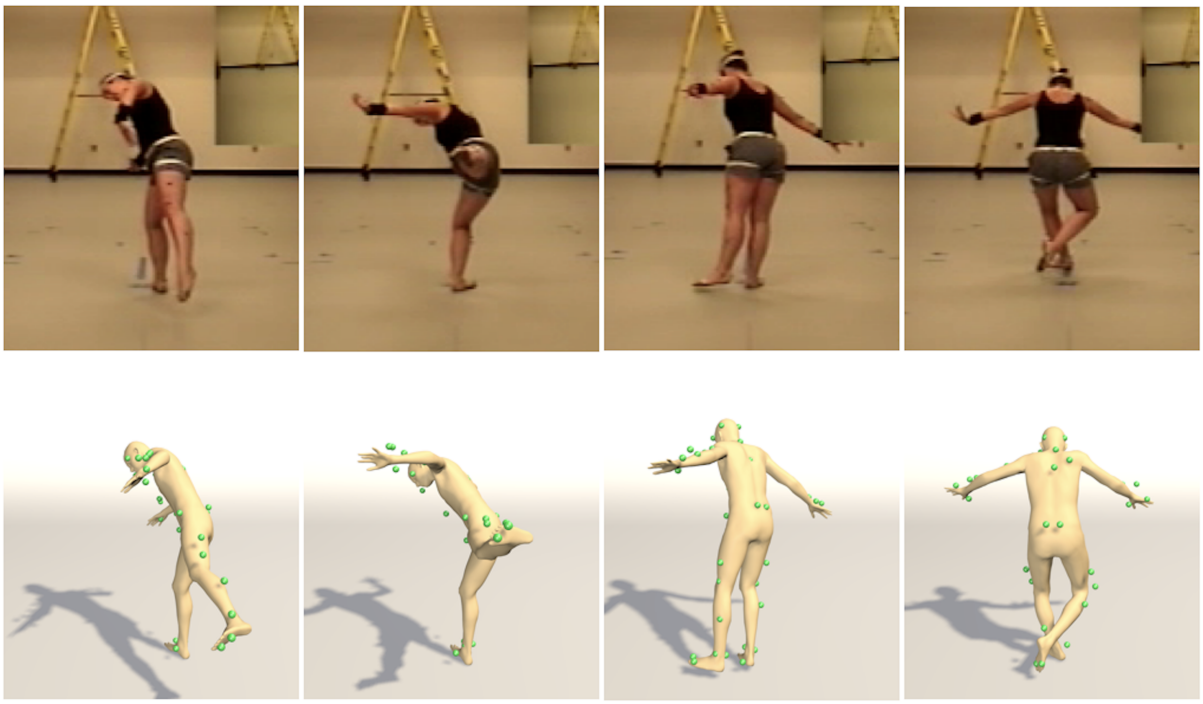


Figure 6.2: Qualitative results of fitting to motion capture data. Frames from original sequence (top) shown with corresponding reconstruction (bottom).

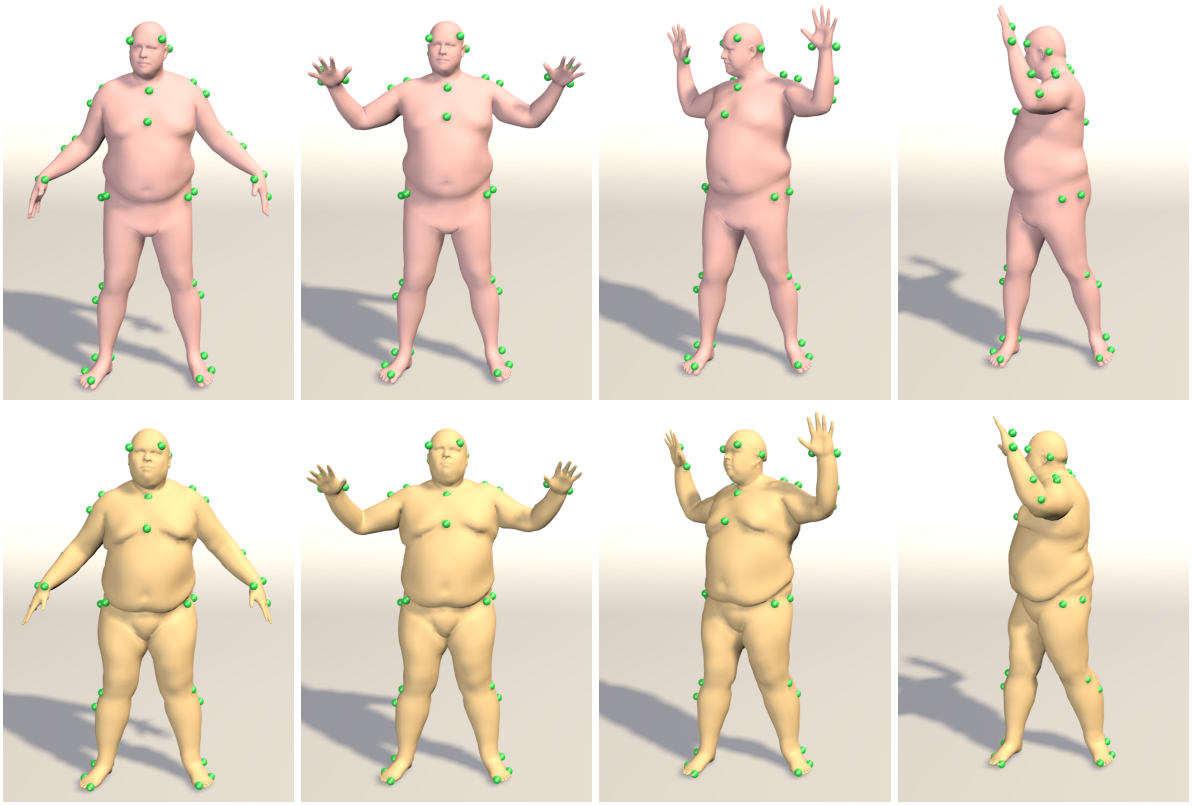


Figure 6.3: Comparison of reconstruction from motion capture data with the MoSh model [10]. Although MoSh (top) is trained on more than 5,000 scans and uses an additional skeleton model, our method with $K = 4$ (bottom) obtains similar results using 10 principal variations only.

Qualitative results of model fitting. In Figures 6.2 and 6.3 we show qualitative results of fitting to 41 markers in sequences from the CMU mocap dataset and 89 markers from MPI MoSh dataset [10]. Fig. 6.2 shows the result in which the learnt body model has quite different geometry to that of the performer. Note that the video frames are just shown for comparison - we use only the 3D markers positions as input. Our fitted model is still able to capture the dynamic poses of the performance.

In Fig. 6.3 we compare against the method [10]. It should be noted that their method uses a model of substantially higher complexity than ours. In particular, it is trained on 3,803 body scans in neutral pose and 1,832 body scans in dynamic poses and uses a 19 parameter skeleton model and retains up to 300 dimensions of the statistical deformation model (10 used in Fig. 6.3). Our result is obtained using a model trained on 20 scans of a single person (chosen to match the body shape of the performer), is entirely mesh-based (we have no articulation model) and we also use only 10 principal variations ($J = 10$). Nevertheless, our results are qualitatively very similar to [10].

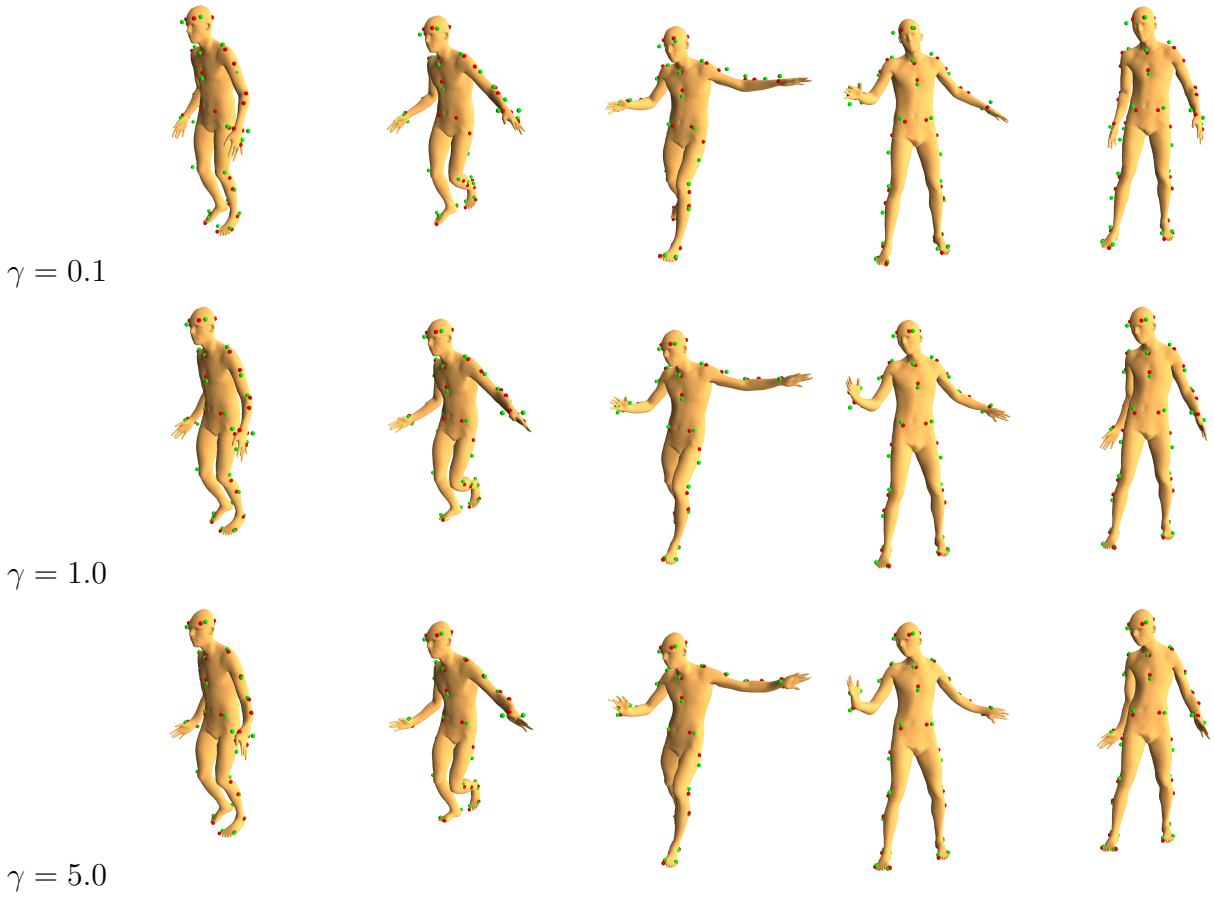


Figure 6.4: Selected frames to demonstrate the effect of marker weight (Top: 0.1; middle: 1; bottom: 5). Red: model correspondence; Green: MoCap markers.

Effect of parameter γ. To analyse the effect of γ which controls the strength of markers fitting term in our application, a series of values for γ are tested in the model fitting process.

Several frames are selected and shown in Fig. 6.4 for illustrations. As γ gets bigger, we can see that MoCap markers (in red) and the model correspondence (in green) are getting closer, while the downside is that it can also distort the model. For example, in face and arm elbows areas we can observe some undesired artefacts. Therefore, it is vital to choose a suitable γ for nice results. In our model fitting experiments, we empirically choose $\gamma = 0.5$ to have a trade-off between marker closeness and overall model quality.

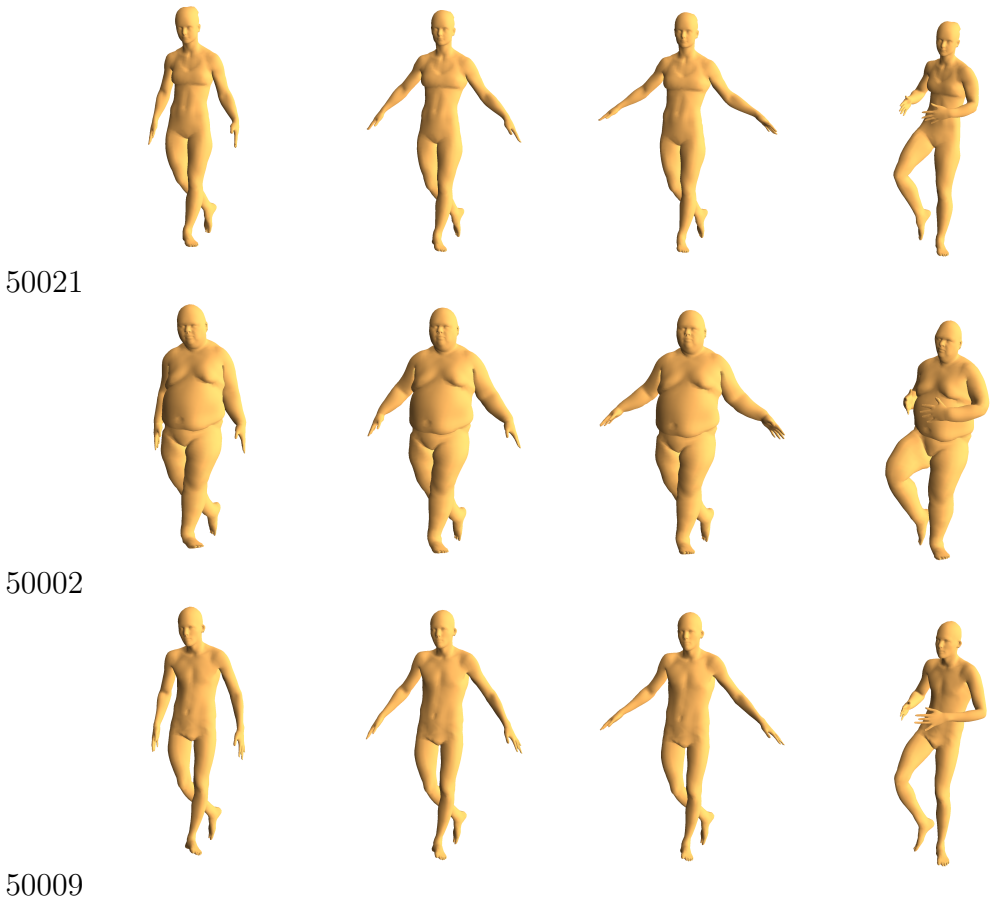


Figure 6.5: Selected frames to show the results of fitting different characters to a "Dance" sequence. Characters from top to bottom correspond: 50021, 50002, and 50009, from Dyna [11] dataset.

Fitting with different characters. Given some motion capture data, the model fitting process is quite robust and not restricted to the underlying identities of the performer. Theoretically, we can fit the model learned from an arbitrary character to the marker data, provided that the correspondence between model mesh and markers are determined in a meaningful way. Indeed, we consider this to be a very nice property for animating motion capture data with a chosen character.

In Fig. 6.5, we show such a comparison of fitting three different characters to the same "dance" sequence. The reconstructed meshes in each case look physically valid and retain the character details of a high level thanks to the efficient multi-resolution optimisation scheme.

6.3 Mesh Editing via Hard Constraints

The model we learned can also be used for physically plausible model-constrained mesh editing. Assume we are given a discrete PGA model and a set of handle vertex positions. Now, one positions the handle vertices manually and asks for a shell satisfying the new handle positions strictly or approximately while being a physically valid deformation of a shell lying on the statistical submanifold. Using the submanifold projection introduced in Section 4.3.4, we define this shell as the minimiser s of the energy

$$\mathbf{W}[s, \mathcal{P}[s]]$$

subject to the constraint positions of the deformed handle vertices. Thus, we ask for the "closest" (in terms of the elastic energy functional $\mathbf{W}$) discrete shell s to the nonlinear submanifold associated with the dominant J modes of variation of our training data. Taking into account the elastic energy functional as a dissimilarity measures ensures once again that membrane and bending distortion are taken into account and is consistent with the involved local projection. Depending on the application one can either regard s or $\mathcal{P}[s]$ as a solution. Indeed, s exactly obeys the prescribed handle vertex positions but $s \notin \mathcal{M}^J$ in general, whereas $\mathcal{P}[s] \in \mathcal{M}^J$ and can be represented by the $2J$ weights α_j but the constraint of the prescribed handle vertex positions is usually fulfilled only approximately. Note that this mesh editing too comes with a selection of a particular representative s from its equivalence class $[s]$, which is determined by the handle vertex positions (as long as there are at least 3 handle vertices not lying on a line).

We emphasise that the proposed mesh editing method is different from those classical editing methods, where only initial mesh and a set of target handles positions are provided. More like the work of [5], the edited mesh is dependent on the feasible deformation space learned from the training data. Concretely, the space is well-captured with our submanifold in terms of a set of J principal variations. Fig. 6.6 shows a comparison of our approach to four classical mesh editing methods and a very recent data-driven approach [5]. The challenging configuration of handles causes classical methods to fail. Gao et al. obtain a plausible body pose but lose details, cause the arms to thin and the back to curve and deform the head. Our result preserves details and retains plausible arms and head and a straight back. Note though that the thickening of the left foot is an artefact.

In Fig. 6.7, we show an example of mesh editing using the SCAPE dataset. The methods [14] and [15] produce visible distortions due to large rotations. Results of [5] and [16] show plausible mesh deformations, but do not capture sufficient details. Our result outperforms these methods in both producing physically valid deformation and retaining high quality details.

To obtain the desired result of the edit, it might be necessary to take into account sufficiently many handles as indicated in Fig. 6.8. Here, we consider the cat model first with five handles and fit to modified handle positions in which the tail tip is moved. To minimise in particular the

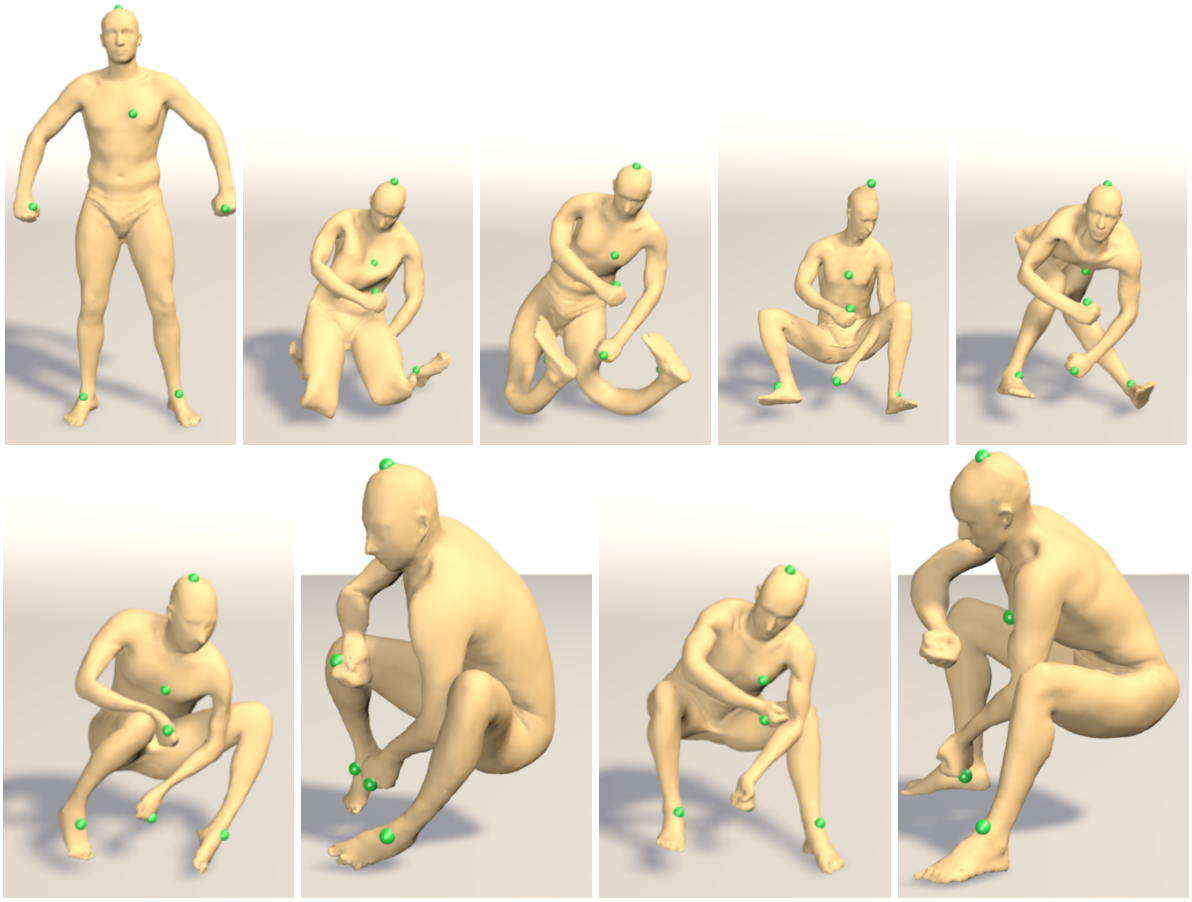


Figure 6.6: Comparison of mesh editing results. Row 1, cols 2-5: [12], [13], [14], [15]. Row 2, cols 1-2: [5]. Row 2, cols 3-4: Ours (with $K = 4$).

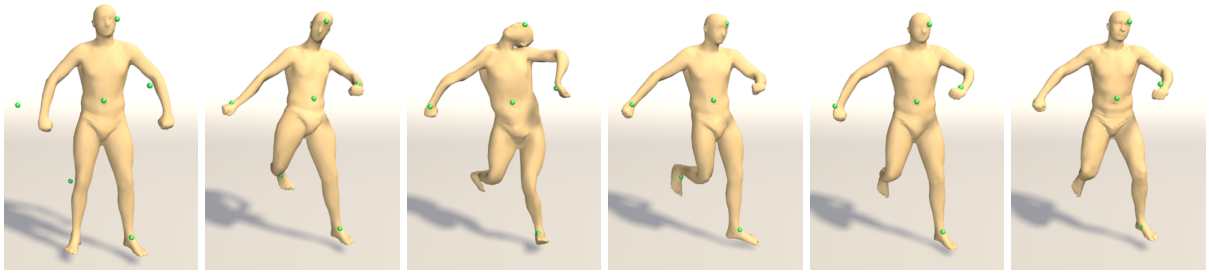


Figure 6.7: Comparison of deformation results using the SCAPE dataset. (a) input shape, (b) [14], (c) [15], (d) [5], (e) [16], (f) Ours.

bending energy our method significantly bends the whole body. This can easily be prevented by adding a sixth handle on the back of the cat (cf Fig. 6.8, col. 3-4).

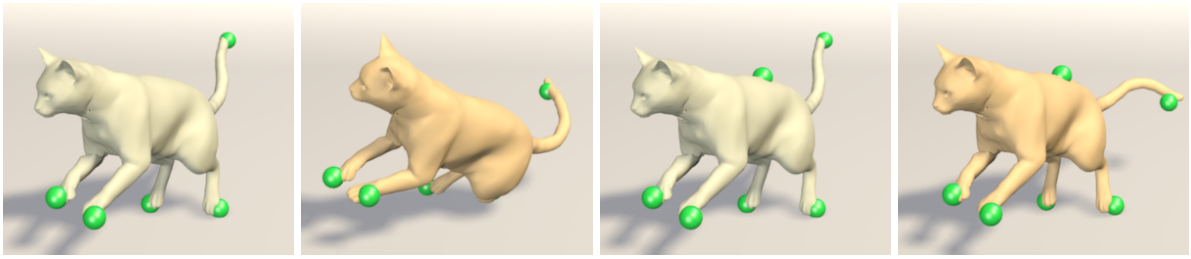


Figure 6.8: Mesh editing with five (col. 1-2) vs. six (col. 3-4) handle positions to be fitted, where the handle at the tail is shifted.

	Dataset	n / J	Offline	Online
Fig. 6.6	SCAPE	71 / 20	73 m	919 s
Fig. 6.8	TOSCA_Cat	10 / 5	66 m	154 s
Fig. 6.2	Dyna_50009	29 / 10	21 m	100 s
Fig. 6.3	Dyna_50021	20 / 10	99 m	321 s

Table 6.1: Timings for motion capture and mesh editing with $K = 4$, but different numbers of training shapes n and principal variations J.

6.4 Human Body Shape Estimation from Clothed Sequence

6.4.1 Introduction

We address the problem of estimating the body shape of a person wearing clothing from 3D scan sequences or visual hulls computed from multi-view images. Reliably estimating the shape under clothing is useful for many applications including virtual try-on, biometrics, and fitness. It is also a key component for virtual clothing and cloth simulation where garments need to be synthesised on top of the minimally-clothed body. Furthermore, most digital recordings of humans are done wearing clothes and therefore automatic methods to extract biometric information from such data are needed. While clothes occlude the minimally-clothed shape (MCS) of the human and make the task challenging, different poses of the person provide different constraints on the shape under the clothes. Previous work [17, 145] exploits this fact by optimising shape using different poses. They use the statistical shape model SCAPE [3] that factorises human shape into subject identity and pose. The main limitation of such approaches is that only the parameters of the statistical model are optimised and so the solutions are constrained to lie on the model space. While statistical models provide powerful constraints on the human shape, they are typically overly-smooth and important identity details such as face features are lost. More importantly, constraints such as "the cloth garment should lie outside the body shape surface" are difficult to satisfy when optimising model parameters. This is because shape deformations in most statistical body models are global, so a step in model space that, for example, shrinks the belly might have the "side effect" of making the person shorter. Therefore, we propose a novel method to estimate the MCS, that recovers accurate global body shape as well as important local shape identity

details. Our hypothesis is that several poses of a person wearing the same clothes provide enough constraints for detailed body shape capture. Moreover, if identity details are visible, e.g. the face, the method should capture them.

To do so, we propose to minimise a single-frame objective function that (i) enforces the scan cloth vertices to remain outside of the MCS, (ii) makes the MCS tightly fit the visible skin parts, and (iii) uses a robust function that snaps MCS to close-by cloth vertices and ignores far away cloth points.

In contrast to previous work, where only model shape parameters are optimised, we directly optimise the $N = 6890$ vertices of a template in a canonical "T" pose (unposed template). This allows us to capture local shape details by satisfying the objective constraints. To satisfy anthropometric constraints, we regularise the optimisation vertices to remain close to a statistical body model. We use SMPL [26], a publicly available vertex-based model that is compatible with standard graphics pipelines. While this formulation has a larger number of variables to optimise, we show that it leads to more accurate and more detailed results.

While simple, the proposed single-frame objective is powerful, as it can be adapted to different tasks. To leverage the temporal information one would like to optimise all scans in the sequence at once. However, given high resolution scans, this is computationally very expensive and memory intensive. Hence, we first register/align all scans by deforming one template to explain both, skin and cloth scan points. These *cloth alignments* are obtained by minimising a special case of the single-frame objective treating all scan vertices as skin. Since the model factors pose and shape, all cloth alignment templates live in a common unposed space; we call the union of these unposed alignments the *fusion scan*. Since the cloth should lie outside the body for all frames we minimise the single-frame objective using the fusion scan as input and obtain an accurate shape template (*fusion shape*) for the person. Finally, to obtain the pose and the time varying shape details, we optimise again the single objective function using the fusion shape as a regulariser. The overview of the method is described in Fig. 6.9. The result is a numerically and visually accurate estimation of the body shape under clothing and its pose that fits the clothed scans.

To validate our approach we use an existing data set [17] and collected a new data set (BUFF: Bodies Under Flowing Fashion) that includes high resolution 3D scan sequences of 3 males and 3 females in different clothing styles. We make BUFF publicly available for research purposes at `http://buff.is.tue.mpg.de/`. BUFF contains in total $11,054$ high resolution clothed scans with ground truth naked shape for each subject. Qualitative as well as quantitative results demonstrate that our method outperforms previous state of the art methods.

6.4.2 SMPL Body Model

SMPL [26] is a body model that uses a learned rigged template $\mathbf{T}$ with $N = 6890$ vertices. The vertex positions of SMPL are adapted according to identity-dependent shape parameters and

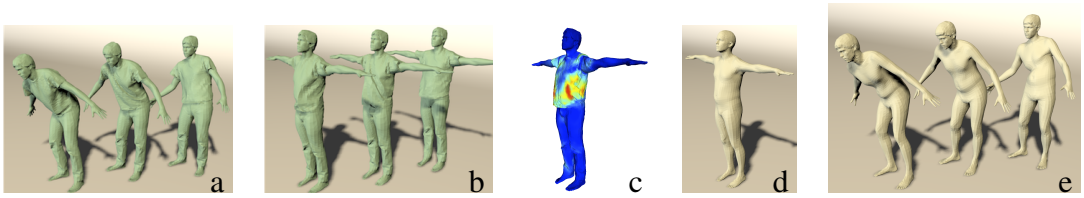


Figure 6.9: a) Cloth alignments b) Unposed alignments c) Fusion scan d) Fusion shape e) Posed and tracked shape. Overview: three example frames are shown. Notice the match in the cloth wrinkles between posed a) and unposed b) alignments. Different time frames provide different constraints in the unposed space. The fusion scan is the union of the frame wise unposed alignments. Colour code indicates variance for that region. From the fusion scan c) we obtain the fusion shape d).

the skeleton pose. The skeletal structure of the human body is modelled with a kinematic chain consisting of rigid bone segments linked by $n = 24$ joints. Each joint is modelled as a ball joint with 3 rotational Degrees of Freedom (DoF), parametried with exponential coordinates ω. Including translation, the pose $\boldsymbol{\theta}$ is determined by a pose vector of $3 \times 24 + 3 = 75$ parameters.

To model shape and pose dependent deformations SMPL modifies the template in an additive way and predicts the joint locations from the deformed template. The model, $M(\boldsymbol{\beta}, \boldsymbol{\theta})$ is then

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) \quad (6.2)$$

$$T(\boldsymbol{\beta}, \boldsymbol{\theta}) = \mathbf{T}_\mu + B_s(\boldsymbol{\beta}) + B_p(\boldsymbol{\theta}) \quad (6.3)$$

where $W(\mathbf{T}_\mu, \boldsymbol{\theta}, \mathbf{J}) : \mathbb{R}^{3N} \times \mathbb{R}^{|\theta|} \times \mathbb{R}^{3K} \mapsto \mathbb{R}^{3N}$ is a linear blend skinning function that takes vertices in the rest pose $\mathbf{T}_\mu$, joint locations $\mathbf{J}$, a pose $\boldsymbol{\theta}$, and the blend weights $\mathbf{W}$, and returns the posed vertices. The parameters $B_s(\boldsymbol{\beta})$ and $B_p(\boldsymbol{\theta})$ are vectors of vertex offsets from the template. We refer to these as shape and pose blend shapes respectively. We use $\mathcal{M}$ to refer to the mesh produced by SMPL. Note that this is different from M, which only refers to the vertices. See [26] for more details.

6.4.3 Method

Our goal is to estimate the naked shape and pose of a subject from a sequence of clothed scans $\{\mathcal{S}\}_k$. If the scans have color information, we use it to split the scan vertices into two sets: the skin ($\mathcal{S}_{\text{skin}}$) and the cloth ($\mathcal{S}_{\text{cloth}}$), otherwise we consider all vertices as cloth ($\mathcal{S}_{\text{cloth}} = \mathcal{S}$). Here we use the segmentation method in [146]. The outputs of our method are: a personalised static template shape $\mathbf{T}_{\text{Fu}}$, the per frame poses θ_k, and the per frame detailed template shapes $\mathbf{T}^k_{\text{Est}}$. Ideally, pose dependent shape changes should be explained by $\mathbf{T}_{\text{Fu}}$ and the pose deformation model; however, in practice models deviate from real data. Therefore, we allow our result $\mathbf{T}^k_{\text{Est}}$ to slightly vary over time. This allows us to capture time changing details, e.g. facial details, present in the data, which the model can not represent.

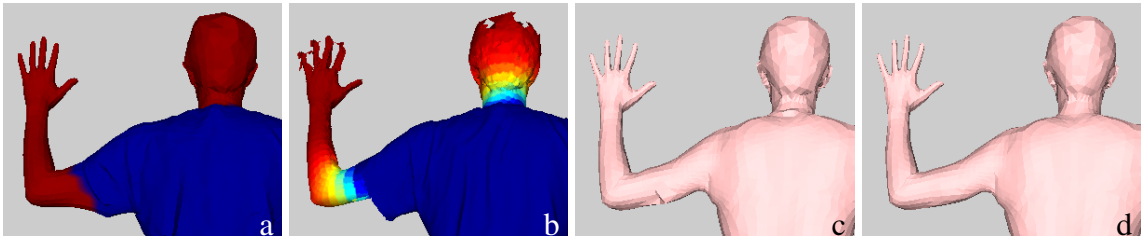


Figure 6.10: Skin term weights. a) alignment segmentation (red: skin, blue: cloth) b) geodesic distance to the closest cloth vertex on the alignment c) broken result with unsmooth neck and arms d) smooth result.

Given a single scan we obtain the shape by minimising a *single-frame objective* function that constrains the scan cloth points to be outside of the body, and penalises deviations from the body to skin parts. However, estimating the shape from a single scan is an under-constrained problem. Fortunately, when all the information in a sequence is considered, the underlying shape is more constrained, as different poses will make the cloth tight to the body in different parts. In order to exploit such rich temporal information we first bring all input scans into correspondence. As a result we obtain a set of posed registrations and unposed template registrations (see Fig. 6.9 a and b). The union of the unposed templates creates the *fusion scan* (Fig. 6.9 c). We use it to estimate a single shape, that we call the *fusion shape* (Fig. 6.9 d). Since all temporal information is fused into a single fusion scan, we can estimate the fusion shape using the same single-frame objective function. Using the fusion shape template as a prior, we can accurately estimate the pose and shape of the sequence. In Fig. 6.9 we show the different steps of the method. The results of each stage are obtained using variants of the same single-frame objective.

We define the single-frame objective function as:

$$E(\mathbf{T}_{\text{Est}}, M(\boldsymbol{\beta}, 0), \boldsymbol{\theta}; \mathcal{S}) = \lambda_{\text{skin}} E_{\text{skin}} + E_{\text{cloth}} + \lambda_{\text{cpl}} E_{\text{cpl}} + \lambda_{\text{prior}} E_{\text{prior}}, \tag{6.4}$$

where E_{skin} is the skin term, E_{cloth} is the cloth term, E_{cpl} is the model coupling term and E_{prior} includes prior terms for pose, shape, and translation. $M(\boldsymbol{\beta}, 0) = \mathbf{T}_{\mu} + B_s(\boldsymbol{\beta})$; $\mathbf{T}_{\mu}$ is the default template of the SMPL model, and $\boldsymbol{\beta}$ are the coefficients of the shape space, see Eq. (6.3). Next we describe each of the terms.

Skin term: We penalize deviations from the model to scan points labelled as skin $\mathbf{s}_i \in \mathcal{S}_{\text{skin}}$ (see Fig. 6.10). A straightforward penalty applied to only skin points creates a discontinuity at the boundaries, which leads to poor results (Fig. 6.10 c). In order to make the cost function smooth, we first compute the geodesic distance of a point in the alignment to the closest cloth point, and we apply a logistic function to map geodesic distance values between 0 and 1 (Fig. 6.10 b). We name this function $g(\mathbf{x}) : \mathbb{R}^3 \mapsto [0, 1]$. The resulting value is propagated to the scan points by nearest distance, and used to weight each scan residual. This way, points close to skin-cloth boundaries have a smooth decreasing weight. This effectively makes the function smooth and

robust to inaccurate segmentations (Fig. 6.10 d).

$$E_{\text{skin}}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta}; \mathcal{S}) = \sum_{\mathbf{s_i} \in \mathcal{S}_{\text{skin}}} g(\mathbf{s}_i)\rho(\text{dist}(\mathbf{s}_i, \mathcal{M}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta}))), \tag{6.5}$$

where dist is a point to surface distance, and $\rho(\cdot)$ is Geman-McClure penalty function. Note that $\text{dist}()$ is computing the closest primitive on the mesh $\mathcal{M}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta})$, triangle, edge or point; analytic derivatives are computed accordingly in each case.

Cloth term: The cloth objective consists of two terms: $E_{\text{cloth}} = \lambda_{\text{outside}} E_{\text{outside}} + \lambda_{\text{fit}} E_{\text{fit}}$. The *outside* term penalises cloth points penetrating the mesh and the *fit* term encourages the mesh to remain close to the cloth surface. This is in contrast to previous work [17] that assumes a closed scan and pushes the model inside. Since scans are not closed surfaces we just penalise cloth points penetrating our closed registration surface. Therefore, the approach is general to point clouds. The outside term is mathematically the sum of penalties for every scan point labelled as cloth $\mathbf{s} \in \mathcal{S}_{\text{cloth}}$ that penetrates the shape mesh:

$$E_{\text{outside}}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta}; \mathcal{S}) = \sum_{\mathbf{s}_i \in \mathcal{S}_{\text{cloth}}} \delta_i \text{dist}(\mathbf{s}_i, \mathcal{M}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta}))^2, \tag{6.6}$$

where δ_i is an indicator function returning 1 if the scan point $\mathbf{s}_i$ lies inside the mesh and 0 otherwise. The activation δ_i is easily obtained by computing the angle between the mesh surface normal and the vector connecting the scan vertex and the closest point in the mesh. Minimisation of the outside term alone can make the shape excessively thin. Hence, the fit term E_{fit} is used to maintain the volume of the naked model. Every cloth scan vertex pays a penalty if it deviates from the body. Since we want to be robust to wide clothing, we define E_{fit} as a Geman-McClure cost function. With this robust cost function, points far away (eg. points in skirt or wide jacket) pay a small nearly-constant penalty. The resulting cloth term is illustrated in the left part of Fig. 6.11.

Coupling term: Optimising only E_{skin} and E_{cloth} results in very unstable results because no human anthropometric constraints are enforced. Therefore, we constrain the template $\mathbf{T}_{\text{Est}}$ to remain close to the statistical shape body model

$$E_{\text{cpl}}(\mathbf{T}_{\text{Est}}, M(0, \boldsymbol{\beta})) = \|\text{diag}(\mathbf{w})(\mathbf{T}_{\text{Est,e}} - M(0, \boldsymbol{\beta})_e)\|^2 \tag{6.7}$$

where the diagonal matrix $\text{diag}(\mathbf{w})$ simply increases the coupling strength for parts like hands and feet where the scans are noisier. Coupling is performed on edges indicated by underscript e. Since we are jointly optimising $\mathbf{T}_{\text{Est}}$, and $\boldsymbol{\beta}$, the model of the shape is pulled towards $\mathbf{T}_{\text{Est}}$ and vice versa. The result of the optimisation is a detailed estimation $\mathbf{T}_{\text{Est}}$ and a model representation of the shape $\boldsymbol{\beta}$.

Prior term: The pose is regularised with a Gaussian prior computed from the pose training

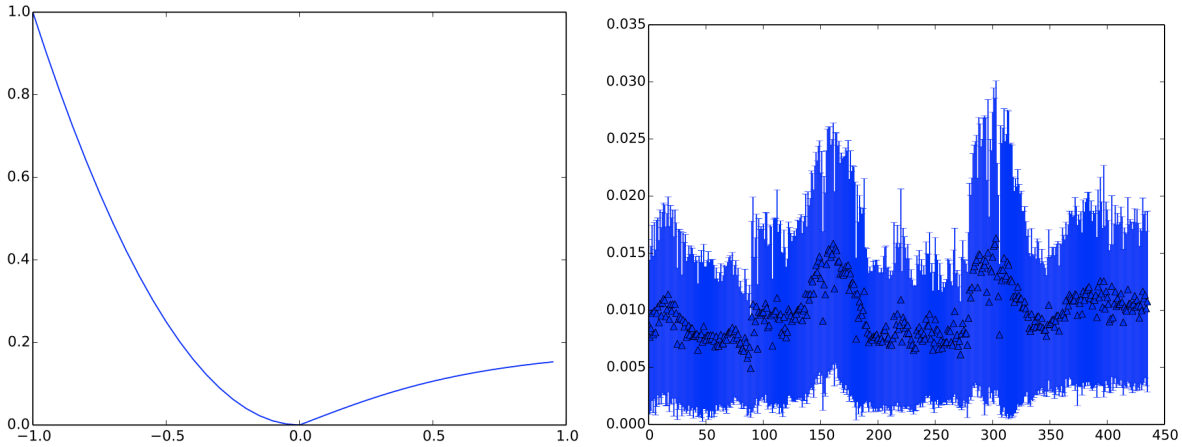


Figure 6.11: Left: Cloth term. The x-axis is the signed distance between $\mathbf{s} \in \mathcal{S}_{\text{cloth}}$ and $\mathcal{M}(\mathbf{T}_{\text{Est}}, \boldsymbol{\theta})$. Points inside (negative) have a quadratic penalty, while points outside are penalised using a robust Geman-McClure function. Right: Root mean squared error and standard deviation between single-frame estimations and the ground truth. Results have significant dispersion depending on pose. (Results for subject 00005, motion "hips" and clothing style "soccer".)

set of [26]. Specifically we enforce a Mahalanobis distance prior on the pose:

$$E_{\text{prior}}(\theta) = (\boldsymbol{\theta} - \mu_{\boldsymbol{\theta}})^T \Sigma_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \mu_{\boldsymbol{\theta}}) \tag{6.8}$$

where the mean $\mu_{\boldsymbol{\theta}}$ and covariance $\Sigma_{\boldsymbol{\theta}}$ are computed from the pose training set. A similar prior can be enforced on the shape space coefficients $\boldsymbol{\beta}$ but we found it did not make a significant difference.

To optimise the single-frame objective we compute the derivatives using the auto-differentiation tool Chumpy [147]. We use the "dogleg" gradient-based descent minimisation method [148].

The problem with the single-frame objective is two fold: the temporal information is disregarded and the frame wise shape changes over time depending on the pose. This can be seen in the right part of Fig. 6.11. The straightforward approach is to extend the single-frame objective to multiple frames and optimise jointly a single $\mathbf{T}_{\text{Est}}$, $\boldsymbol{\beta}$ and the N_{frames} poses $\{\boldsymbol{\theta}^k\}_{k=1}^{N_{\text{frames}}}$. Unfortunately, our scans have around $150,000$ points, and optimising all poses jointly makes the optimisation highly inefficient and memory intensive. Furthermore, slight miss-alignments in pose make the shape shrink too much. Hence, we propose an effective and more efficient solution. We first sequentially register all the scans to a single clothed template. For registration we use the single-frame objective function with no cloth term. From this we obtain a template clothed per frame $\mathbf{T}_{\text{cloth}}^k$. The interesting thing is that the set of $\mathbf{T}_{\text{cloth}}^k$ templates contain the non-rigid cloth motion with the motion due to pose factored out, see Fig. 6.9. The naked shape should lie inside all the clothed templates. Hence we gather all templates and treat them as a single point cloud that we call the fusion scan $\mathcal{S}_{\text{Fu}} = \{\mathbf{T}_{\text{cloth}}^k\}_{k=1}^{N_{\text{frames}}}$. Hence, we can easily obtain

a single shape estimate by using again the single-frame objective

$$\mathbf{T}_{\text{Fu}} = \underset{\mathbf{T}_{\text{Est}},\boldsymbol{\beta}}{\arg\min} \quad E(\mathbf{T}_{\text{Est}}, M(\boldsymbol{\beta}, 0), 0; \mathcal{S}_{\text{Fu}}). \tag{6.9}$$

The obtained fusion shapes are already quite accurate because the fusion scan carves the volume where the naked shape should lie.

Finally we use the fusion shape to perform tracking regularising the estimated shapes to remain close to the fusion shape. We achieve that by coupling the estimations towards the fusion shape instead of the SMPL model shape space. So the coupling term is now $E_{\text{cpl}}(\mathbf{T}_{\text{Est}}, M(0, \boldsymbol{\beta})) \mapsto E_{\text{cpl}}(\mathbf{T}_{\text{Est}}, \mathbf{T}_{\text{Fu}})$. Detailed shapes are obtained minimising

$$\mathbf{T}^k_{\text{Est}} = \underset{\mathbf{T}_{\text{Est}},\boldsymbol{\theta}}{\arg\min} \quad E(\mathbf{T}_{\text{Est}}, \mathbf{T}_{\text{Fu}}, \boldsymbol{\theta}; \mathcal{S}^k). \tag{6.10}$$

6.4.4 Datasets

In this section we present our new BUFF dataset. We start by introducing the previous dataset.

INRIA dataset. The INRIA dataset [17] consists of sequences of meshes obtained by applying a visual hull reconstruction to a 68-color-camera (4M pixels) system at 30fps. The dataset includes sparse motion capture (MoCap) data of 6 subjects (3 female, 3 male) captured in 3 different motions and 3 clothing styles each. The texture information of the scans is not available. Fig. 6.12 a) and b) show frames from the dataset. As shown in Fig. 6.12 c) and d), their statistical body model does not capture the individual details of the human shape. The main drawback of this "ground truth shape" is that it biases the evaluation to the model space. All recovered details, that fall outside the model, will be penalised in the quantitative evaluation. Alternatively, one could compare the obtained shape directly with the visual hull. Unfortunately, visual hulls are not very accurate, sometimes over-estimating, sometimes under-estimating the true shape. While relevant for qualitative evaluation of the shape estimates, we believe that this dataset is limited for quantitative evaluation. This motivated us to create BUFF, which preserves details and allows quantitative evaluation of the shape estimation.

BUFF. To create BUFF, we use a custom-built multi-camera active stereo system (3dMD LLC, Atlanta, GA) to capture temporal sequences of full-body 3D scans at 60 frames per second. The system uses 22 pairs of stereo cameras, 22 colour cameras, 34 speckle projectors and arrays of white-light LED panels. The projectors and LEDs flash at 120fps to alternate between stereo capture and colour capture. The projected texture pattern makes stereo matching more accurate, dense, and reliable compared with passive stereo methods. The stereo pairs are arranged to give full body capture for a range of activities, enabling us to capture people in motion. The system outputs 3D meshes with approximately 150K vertices on average. BUFF consists of 6 subjects, 3 male and 3 female wearing 2 clothing styles: a) t-shirt and long pants and b) a soccer outfit, see Fig. 6.13. The sequence lengths range between 4 to 9 seconds (200-500 frames) totaling 13,632

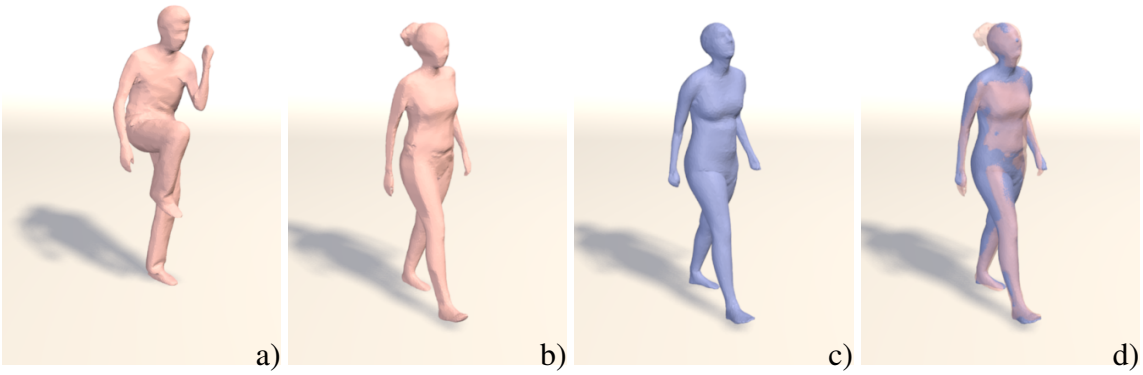


Figure 6.12: INRIA Dataset: a) and b) scan samples; c) estimated "ground truth" shape for b); d) overlay of b) and c).

Figure 6.13: BUFF Dataset: To validate our method we captured a new dataset including 6 subjects wearing different clothing styles and different motion patterns.

3D scans. As shown by previous state of the art methods [145], skin colour is a rich source of information. We thus include texture data in our dataset. All subjects gave informed written consent before participating in the study. One subject did not give permission to release their data for research purposes. Consequently, the public BUFF dataset consists of 11,054 scans.

In order to estimate the "ground truth" shapes in our dataset we capture the subjects in "minimal clothing" (tight fitting sports underwear). Participants performed an "A-T-U-Squat" motion (first row of Fig. 6.14). For all frames, we use our method to fit the data considering all vertices as "skin" . We obtain N template meshes $\mathbf{T}_{\mu}^{i}$, which do not perfectly match, because the pose and the shape are not perfectly factorised in the SMPL model [26]. We define the $\mathbf{T}_{\mathrm{GT}}$ as the mean of the estimates of all frames.

We quantitatively estimated the accuracy of our "ground truth" MCS estimations. More than half of the scan points are within 1.5mm distance of $\mathbf{T}_{\mathrm{GT}}$ and 80% closer than 3mm. Because the scan point cloud has some noise (e.g. points of the scanning platform, poorly reconstructed hands, hair,...), we believe the computed $\mathbf{T}_{\mathrm{GT}}$ provides an accurate explanation of the subjects "minimally clothed shape". In the bottom row of Fig. 6.14 we qualitatively show the visual

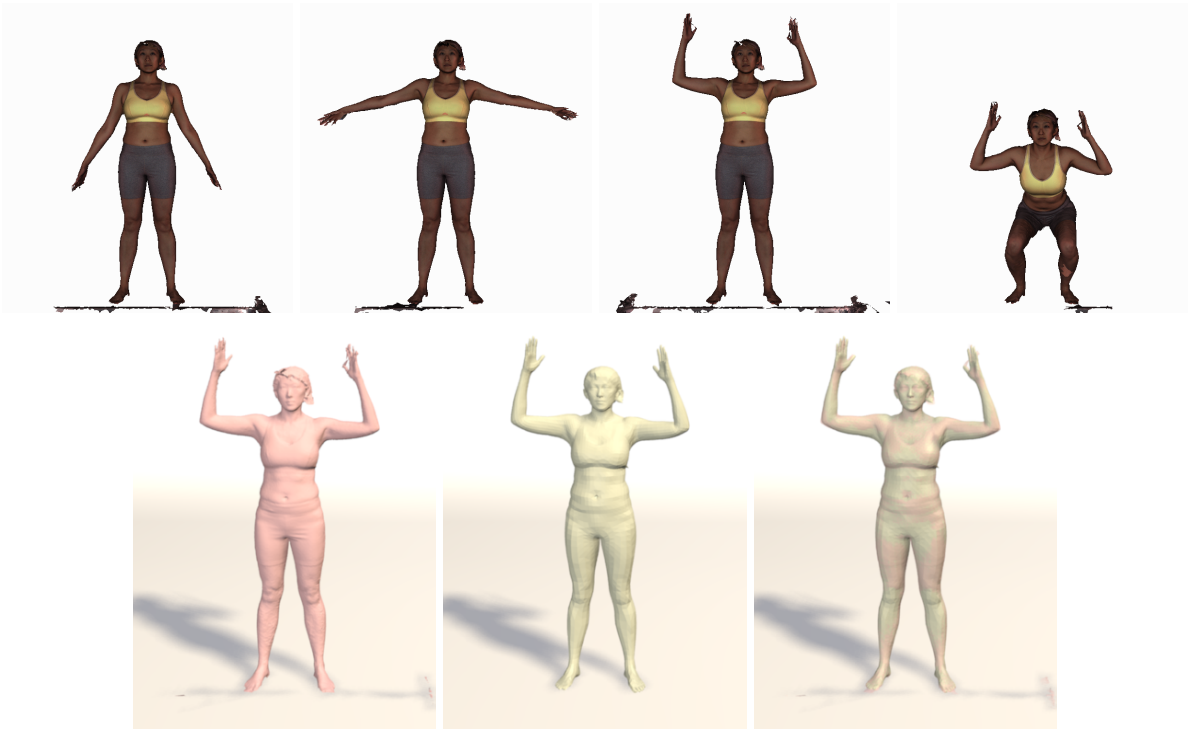


Figure 6.14: Top row: Subject 03223 performing the "A-T-U-Squat" motion in "minimal clothing". These scans are used to compute the ground truth MCS $\mathbf{T}_{\mathrm{GT}}$. Bottom row: triplet of scan, estimated ground truth model and both overlayed (frame 000150). The proposed ground truth shape captures the details present in the scan point cloud.

accuracy of the computed ground truth MCS.

6.4.5 Experiments

In this section we present the evaluation measures and the obtained qualitative and quantitative results.

Evaluation on INRIA dataset. We evaluate our results quantitatively on pose estimation, and qualitatively on shape estimation in the INRIA dataset [17]. We estimated the shape for all tight clothes sequences. To initialised the pose we use the automatically computed landmarks of [149]. We compare the MoCap marker locations to the corresponding vertices of our results and [17]. 10 frames sampled evenly from the first 50 frames of each sequence are used to obtain 10 correspondence sets. In Fig. 6.15 we report the average errors for all frames and correspondence sets; our method achieves state of the art results in pose estimation. In the first row of Fig. 6.16 we present qualitative results for the INRIA dataset. Our results are plausible estimates of minimally-clothed shapes. In the second row of Fig. 6.16 we qualitatively compare our results to previous work on the dancer sequence from [18]. Our results visually outperform previous state of the art. Results are best seen in the video included at `http://buff.is.tue.mpg.de/`.

Evaluation on BUFF dataset. To quantitatively evaluate the results in BUFF, we compare the estimated body shapes with the computed ground truth meshes. We define the "registration

	t-shirt, long pants						soccer outfit					Avrg.
tilt twist left	00005	00096	00032	00057	03223	00114	00005	00032	00057	03223	00114	Avrg.
[17]	17.29	18.68	13.76	17.94	17.90	15.42	16.77	16.96	18.52	20.41	16.40	17.27
fusion mesh	2.58	2.89	2.39	2.53	2.43	2.38	2.50	2.63	2.37	2.28	2.28	2.47
detailed	**2.52**	**2.83**	**2.36**	**2.44**	**2.27**	**2.31**	**2.44**	**2.59**	**2.28**	**2.17**	**2.23**	**2.40**
hips	00005	00096	00032	00057	03223	00114	00005	00032	00057	03223	00114	Avrg.
[17]	21.02	21.66	15.77	17.87	21.84	18.05	22.52	16.81	19.55	22.03	17.54	19.51
fusion mesh	2.81	2.71	2.66	2.66	2.54	2.65	2.65	2.63	2.58	2.50	2.57	2.63
detailed	**2.75**	**2.64**	**2.63**	**2.55**	**2.40**	**2.56**	**2.58**	**2.59**	**2.50**	**2.38**	**2.51**	**2.55**
shoulders mill	00005	00096	00032	00057	03223	00114	00005	00032	00057	03223	00114	Avrg.
[17]	18.77	19.02	18.02	16.50	18.15	14.78	18.74	17.88	15.80	19.47	16.37	17.59
fusion mesh	2.56	2.92	2.74	2.46	2.42	2.69	2.89	2.87	2.37	2.44	2.58	2.63
detailed	**2.49**	**2.85**	**2.72**	**2.37**	**2.26**	**2.59**	**2.83**	**2.82**	**2.28**	**2.33**	**2.51**	**2.55**

Table 6.2: Numerical results for the estimated naked shapes. We report the root mean squared error in millimeters of point to surface distance between the posed GT mesh and the method result. The best value is highlighted in bold.

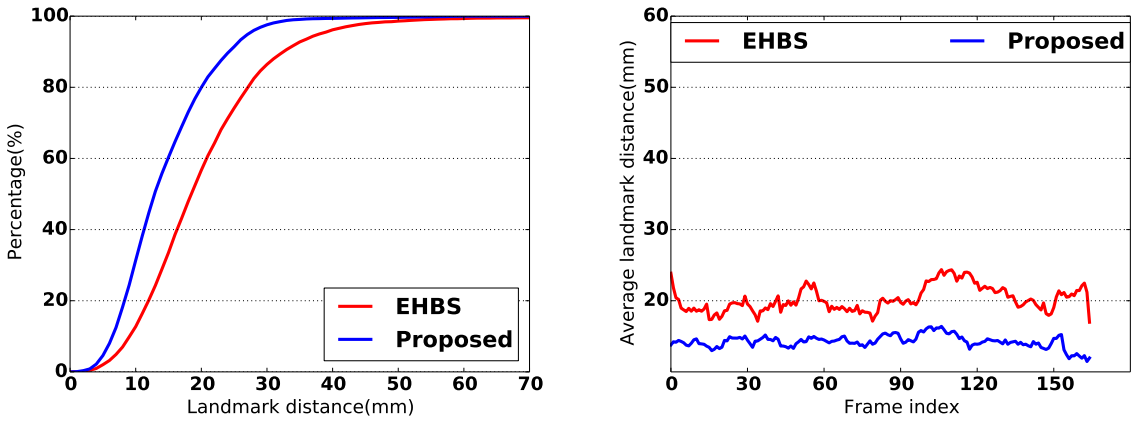


Figure 6.15: Pose estimation accuracy on INRIA dataset. Left: Percentage of landmarks with error less than a given distance (horizontal axis) in mm. Right: per frame average landmark error. EHBS is [17].

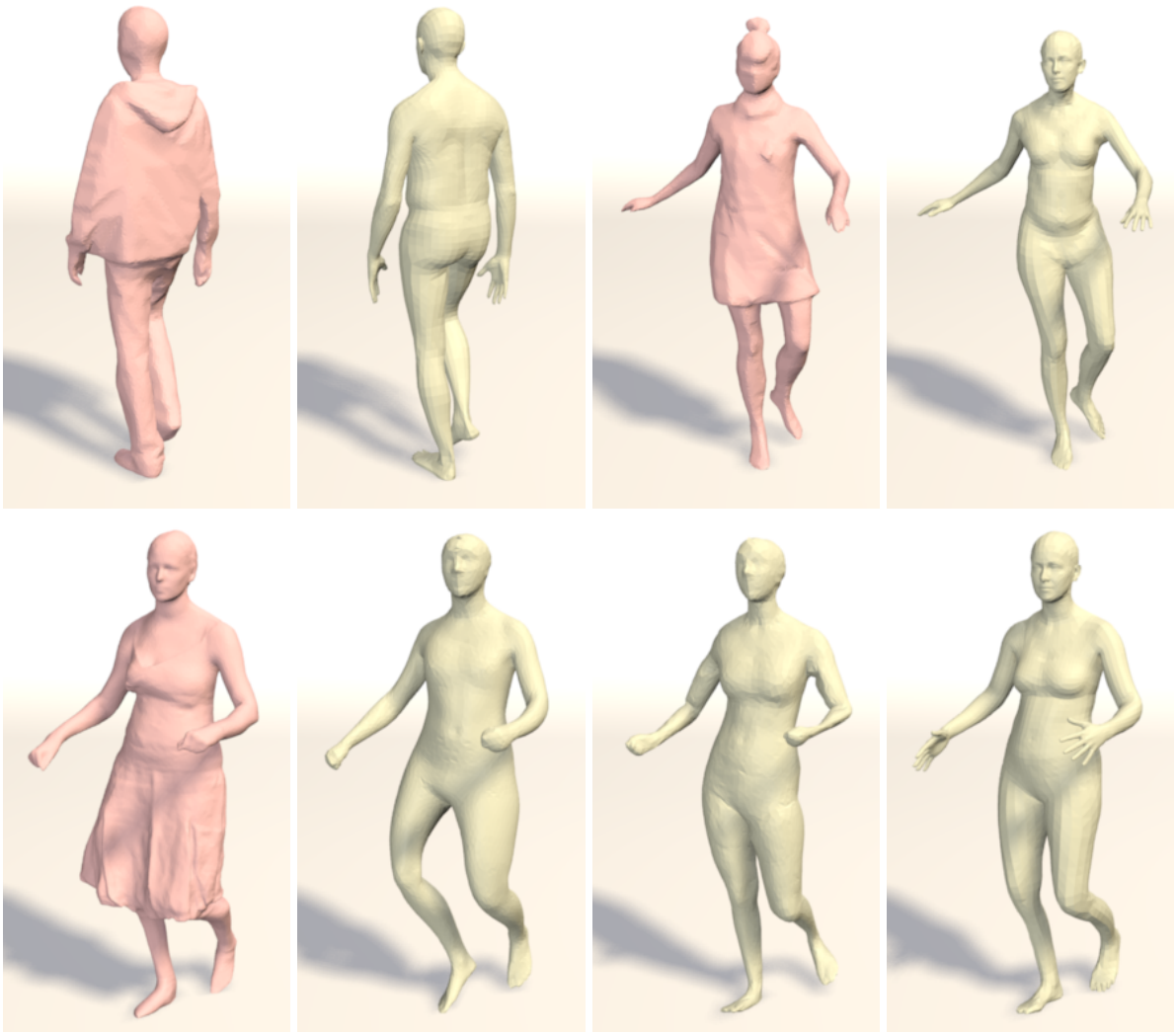


Figure 6.16: Top: Qualitative results on the INRIA dataset; scan (pink), our result. Bottom: Qualitative comparison on Dancer sequence [18]. From left to right: scan, Wuhrer et al. [19], Yang et al. [17], our result.

error" of the estimated body shape as the scan-to-model distance with respect to the ground-truth MCS. Given a result mesh S, we optimise for pose $\boldsymbol{\theta}$ so that the posed $\mathbf{T}_{\mathrm{GT}}$ best fits S. Then the error between S and the posed $\mathbf{T}_{\mathrm{GT}}$ is computed as the Euclidean distance between each vertex in $\mathbf{T}_{\mathrm{GT}}$ and its closest point on the surface S. In Tab. 6.2 we show the numerical results obtained by [17], our fusion mesh, and our detailed mesh. The results obtained with our method systematically outperform the best state of the art method. In Fig. 6.17 we show qualitative results on the pose estimations. Our method properly recovers the scan pose, and visually outperforms [17], especially in elbow and shoulder estimations. In Fig. 6.18 we show qualitative results of the shape estimations. The proposed fusion shape accurately recovers the body shape, while the detailed shape is capable of capturing the missing details. While the detailed shape is visually closer to the ground truth, quantitatively, both results are very similar, see Tab. 6.2. In order to evaluate the robustness of the method when skin/cloth segmentation is not available we evaluate our method labelling the scans of BUFF as *all cloth*. While the obtained shapes are less detailed, they are still accurate. The obtained mean error is $\approx$ 3mm (all cloth) compared to ≈ 2.5mm (detailed) when using our proposed full method.

Computation time and parameters. The single-frame objective computation takes $\sim$10 seconds per frame, fusion mesh is computed in $\sim$200 seconds. The detail refinement takes $\sim$40 seconds per frame. Sequences are computed in parallel and computations are executed on an 3GHz 8-Core Intel Xeon E5. Shapes on BUFF were estimated using $\lambda_{\text{skin}} = 100$, $\lambda_{\text{outside}} = 100$, $\lambda_{\text{fit}} = 3$ and $\lambda_{\text{cpl}} = 1$. For INRIA data we decreased the fit term $\lambda_{\text{fit}} = 1$ to be more robust to wide clothing.

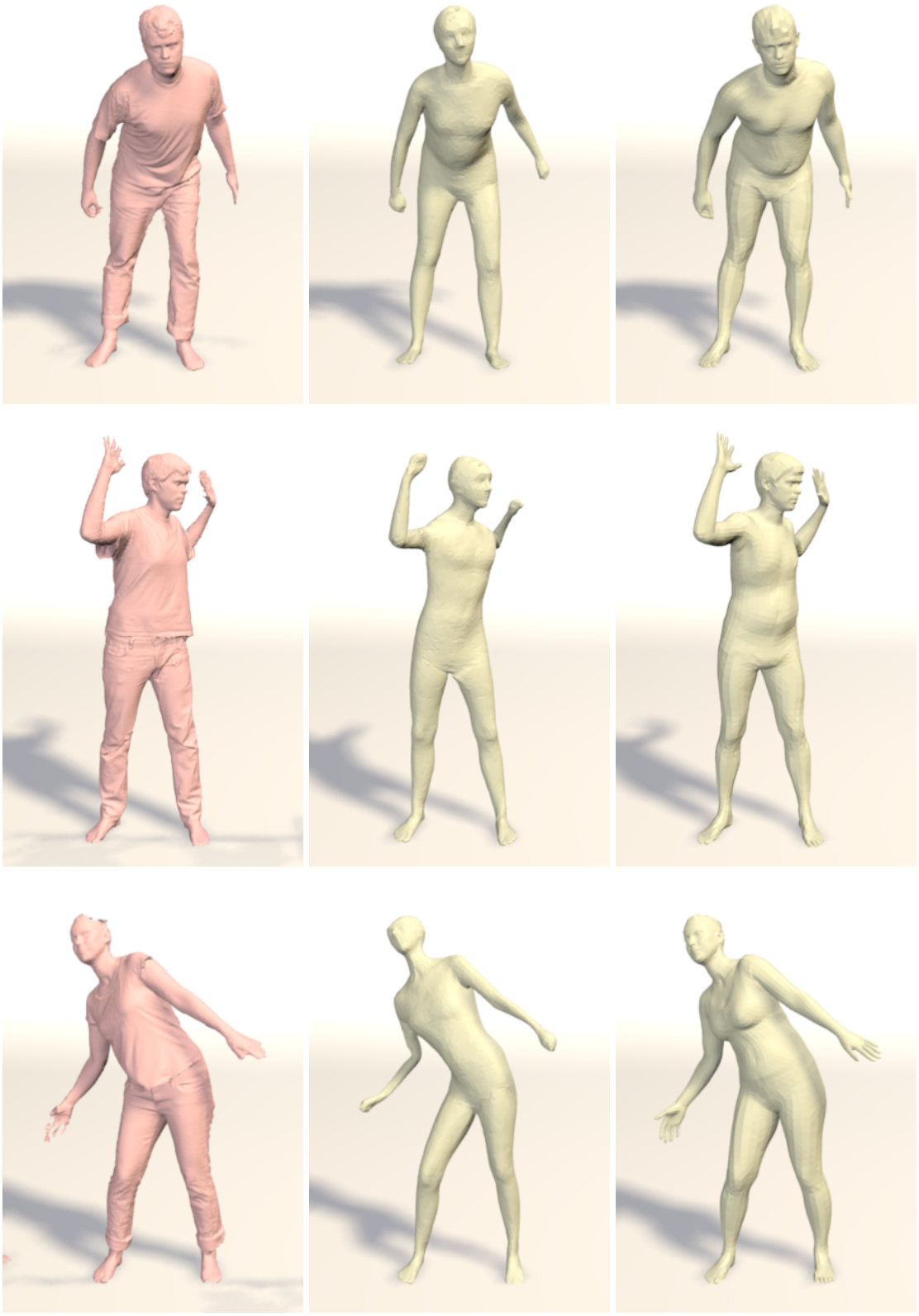


Figure 6.17: Qualitative pose estimation results on BUFF dataset. Left to right: scan, Yang et al. [17], our result.

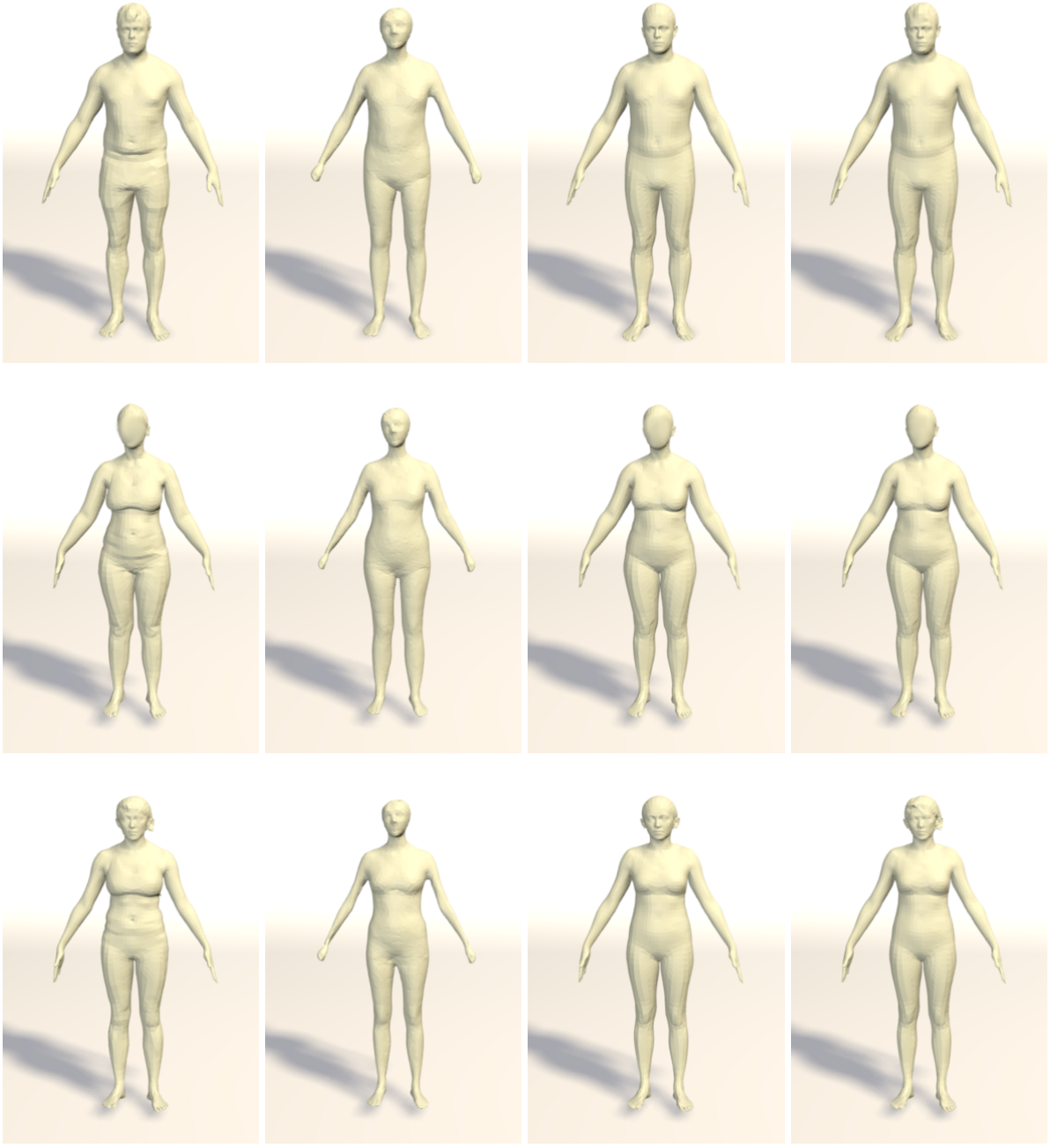


Figure 6.18: Qualitative shape estimation results on BUFF dataset. Left to right: ground truth shape, Yang et al. [17], fusion shape (ours), detailed shape (ours).

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summaries

This thesis explores "unconventional" spaces for 3D shape modelling. We perform statistical shape analysis in the *space of discrete shells*. The approach is based on a physically-inspired energy functional measuring membrane and bending distortion. The result is a physically sound statistical shape model, which is able to handle large nonlinear articulations and deformations. As of shape correspondence, we provide a method to optimise correspondence across a whole shape collection in the *space of functional maps*. This is more flexible and efficient than point-wise or segment-wise map representations. We now revisit the contributions of each work in turn.

The contribution of Shell PCA in Chapter 3 is to show that an extension of PCA in shell space is better to capture nonlinear deformations even when the training data is very sparse. This is our first attempt to provide a hybrid between physical and statistical model. The squared distance is replaced by the elastic energy required to deform one shape to another. Shape reconstruction results are shown for a given set of facial meshes of various expression. This method is limited in that the learned statistics are based on vertex-displacement, therefore it is alignment-dependent and would not work with data that exhibits large deformations.

In Chapter 4, we propose another method Shell PGA in the same shell space. Unlike Shell PCA, this method directly works with shapes rather than displacement between input shapes and the average. One advantage of using elastic energy is that it is rigid body motion invariant. Therefore, the whole approach is carefully crafted to maintain this property and eventually becomes an alignment-free approach. This is vital in shape modelling because a choice of the "best" alignment may not exist in many cases, especially for articulated shapes with large nonlinear deformations.

A groupwise functional maps approach presented in Chapter 5 is applied to optimise the minimal set of functional maps between one reference shape and each of other shapes. The functional maps between arbitrary two shapes can be composed using the minimal set maps. In

particular, a set of candidate functions are proposed for obtaining correspondence for 3D face shapes. Besides the theoretical contribution, we believe that our work was the first to apply the functional maps framework in a practical setting using real scanned data.

Finally, several applications are demonstrated in Chapter 6. We show that the human body model learned as described in Chapter 4 is well suited for model-constrained mesh editing, and mesh reconstruction from sparse landmarks from motion capture data. The performance of our model learned from only 20 scans is qualitatively comparable to the method that trained on thousands of scans, with an explicit Skeleton model. In addition, we contribute a new algorithm for estimating human body shape and poses from clothed 3D sequence. Based on SMPL model, the state-of-the-art vertex-based human body model, we achieve detailed personalised human body shape estimation.

All the these works are accompanied by online materials: Groupwise Functional Maps and Shell PCA are released and publicly available in GitHub, Shell PGA will become available in GitHub shortly after the associated paper becomes available online. "BUFF" dataset is freely available online for research purpose.

7.2 Conclusions

We can draw a number of general conclusions after exploring novel spaces for both 3D shape modelling and matching problems:

1. Introduction of a physical model allows highly plausible interpolation between and extrapolation beyond a very sparse data sample.

2. Given any measure that locally approximates Riemannian distance, time-discrete statistics can be used to build a Riemannian model.

3. Functional maps are applicable to real data with typical systematic noise.

4. The reduced dimensionality of functional maps makes groupwise optimisation feasible and brings a lot of benefits.

5. A good statistical model solves very challenging applications that are not addressable with deep learning methods that would require much larger samples of training data.

7.3 Future work

Integration of Functional Maps and Shell space. Intuitively, shell energy provides a richer shape difference measure in terms of meaningful deformation energy. The question is whether we can bring functional maps and shell space together. Jointly matching and modelling a collection

of shapes would be a very interesting direction for future work. Current groupwise shape matching methods do not output point-to-point correspondence directly, and a post-processing conversion step is needed. To this end, a robust groupwise point-to-point conversion method is worth more research so that the output could be picked up by existing modelling approaches.

Computational efficiency. The current Shell PGA model implemented using MATLAB could not provide reconstruction or model fitting within interactive timing requirements. To improve this, a more efficient language such as C++ could be considered for re-implementing this. Alternatively, the derivative could be computed using GPU array to gain more speed up. At a deeper level, approximations that work in a transformed space of triangle edge lengths, areas and angles could be applied to our framework for a substantial speed up.

Machine learning. It is notable that the current framework of shell PGA employs a discrete version of elastic deformation energy as an approximation of Riemannian distance. Theoretically, any other metric could also be plugged in as well. Any invariance property of the chosen metric would be transferred to the learnt models. In a broader view, not limited to 3D shape modelling, extension of our method to other modalities such as images is also a sensible direction for future work. In addition, only PCA is exploited as an example in shell modelling, there exist much more sophisticated manifold learning methods that we could apply in shell space, e.g. Principal Component Regression and Linear Discriminant Analysis.

Applications. Based on the current shell PCA/PGA models, there are many potentially interesting follow-up applications. *First*, model transport is a solution to effectively achieving "rig personalisation". For example, expression data captured from one character could be easily transferred to a different person while at the same time adapting blendshapes. *Second*, the shell submanifold is perfect for blendshape fitting and non-linear blendshape learning where rigid body motion invariance becomes a built-in property. However, this also poses a problem that the reconstructed shapes are only determined up to a rigid body motion, a.k.a. motion stabilisation. *Third*, non-rigid alignment combining global physical constraint and local feature descriptors is practically a very useful technique. Shell space provides an effective physical constraint allowing only minimal distortion, while local feature matching would be critical especially in the early stages. *Finally*, can we reduce data requirements for deep learning based applications? As shown in our experiments, shell model requires much less data to learn the principal variations than traditional Euclidean space model. Therefore, we can either plug in the shell model or use the model to generate realistic, continuous shapes given sparse training data. The latter is guaranteed to be working thanks to the robust interpolation and extrapolation capacity of shell space.

APPENDIX A

APPENDIX

A.1 Computation of geodesic mean and geodesic paths

Notation convention: As before, input shapes are denoted by s^i, $i = 1, \dots, m$, the geodesic average is denoted by $\bar{s}$ and discrete (geodesic) paths of length $K+1$ connecting $\bar{s}$ and s^i are given by $s_0^i, s_1^i, \dots, s_K^i$. In particular, we have $s_0^i = \bar{s}$ and $s_K^i = s^i$ for all i.

A.1.1 Pseudo code

The core ingredient is the `update` function defined below. Here either the geodesic mean or the geodesic segments are updated. We assume that references to all free shapes (i.e. $\bar{s}$ and $(s_k^i)_{i,k}$) are accessible within the main Algo. 8.

$$\texttt{update}(k, \Delta_l): \quad \begin{cases} \bar{s} \leftarrow \arg\min_s \mathbf{F}_{\Delta_l}^K[s] & \text{if } k = 0 \\ \\ \forall i: \ s_k^i \leftarrow \mathbf{Av}\left(s_{k-\Delta_l}^i,\, s_{k+\Delta_l}^i;\, \frac{1}{2}\right) & \text{if } k > 0 \end{cases}$$

Algorithm 6 v-cycle

1: **Input:** $\max \in \mathbb{N},\ \Delta_l \in \mathbb{N}$
2: **Output:** some error $\delta > 0$
3: **for** $k := 0$, $k < \max$, $k \mathrel{+}= \Delta_l$ **do**
4: `update`(k, Δ_l);
5: **end for**

Algorithm 7 w-cycle

Input: $\max \in \mathbb{N},\ \Delta_l \in \mathbb{N}$
Output: some error $\delta > 0$
for $j := \Delta_l$, $j \leq \max$, $j \mathrel{+}= \Delta_l$ **do**
    `v-cycle`(j, Δ_l);
end for

Algorithm 8 Geodesic mean by cascadic Gauss-Seidel updates

Input: $L \in \mathbb{N},\ s^1, \ldots, s^n \in \mathcal{M}$
Output: $\bar{s} \in \mathcal{M}$ (and s_k^i for $0 < k < K$ and $\forall i$)

Set $K := 2^L$;

// initialize $\bar{s}$ as elastic average
`update`$(0, K)$;
// initialize $(s_k^i)_{i,k}$ for all $i = 0, \ldots, n$ in parallel:
for $k := 1$ **to** $K - 1$ **do**
    $s_k^i \leftarrow \mathbf{Av}\left(\bar{s},\ s^i;\ \frac{k}{K}\right)$
end for

// cascadic approach in time
for $l := 1$ **to** L **do**
    set $\Delta_l := 2^{L-l}$;
    $\delta := \infty$;
    // Gauss-Seidel updates
    while $\delta > \epsilon_l$ **do**
        δ = `cycle`(K, Δ_l); // either `v-cycle` or `w-cycle`:
    end while
end for

A.2 Derivatives of groupwise functional maps objectives

The core part to our groupwise functional maps lies in solving for the optimisation problem in matrix manifold space. More specifically, our optimisation objective is composed of two terms: preservation of descriptors and basis commutativity. Next, we describe the deviation of the gradient of both terms, as used by the Trust-region solver.

A.2.1 Solving for descriptor preservation constraints

Let $\mathbf{P}_i \in \mathbb{R}^{n,N}$ and $\mathbf{P}_j \in \mathbb{R}^{n,N}$ be the descriptors coefficients of shape i and j, respectively. Here, n denotes the dimension of Laplacian-Beltrami basis, and N denotes the number of descriptors. Let $\mathbf{T}_i$ denote the matrix representation of the map from shape i to the reference shape (without loss of generality, this could be the first shape). To get the matrix mapping from shape j to shape i, we have $\mathbf{C}_{i,j} = \mathbf{T}_i^{-1}\mathbf{T}_j$.

The objective for this term is

$$O = \sum_{i,j} \|\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j\|_F^2 \tag{A.1}$$

we then introduce a dummy index k:

$$O_k = \sum_{i,k} \|\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_k\mathbf{P}_k\|_F^2 + \sum_{k,j} \|\mathbf{P}_k - \mathbf{T}_k^{-1}\mathbf{T}_j\mathbf{P}_j\|_F^2 \tag{A.2}$$

then replace i with j:

$$O_= \sum_{j,k} \|\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_k\mathbf{P}_k\|_F^2 + \sum_{k,j} \|\mathbf{P}_k - \mathbf{T}_k^{-1}\mathbf{T}_j\mathbf{P}_j\|_F^2 \tag{A.3}$$

then replace k back with i:

$$O_k = \sum_{j,i} \|\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i\|_F^2 + \sum_{i,j} \|\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j\|_F^2 \tag{A.4}$$

As matrix frobenius norm could be written as dot product form, we could rewrite the above:

$$\begin{aligned} O &= O_{ji} + O_{ij} \\ &= \mathrm{Tr}\,(\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i)^T(\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i) + \mathrm{Tr}\,(\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j)^T(\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j) \end{aligned} \tag{A.5}$$

Then we proceed to get the gradient of the above term with respect to $\mathbf{T}_i$s which are our variables.

$$\begin{aligned}\frac{\partial O}{\partial \mathbf{T}_i} &= \frac{\partial O_{ji}}{\partial \mathbf{T}_i} + \frac{\partial O_{ij}}{\partial \mathbf{T}_i} \\ &=< \mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i, \mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i > + < \mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j, \mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j > \end{aligned} \tag{A.6}$$

We first show how to get the gradient w.r.t. $\mathbf{T}_i$ for the former part, then it applies to the latter part as well.

$$\begin{aligned} & < \mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i, \mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i > \\ =& < \mathbf{P}_j - \mathbf{T}_j^{-1}(\mathbf{T}_i + \mathbf{H})\mathbf{P}_i, \mathbf{P}_j - \mathbf{T}_j^{-1}(\mathbf{T}_i + \mathbf{H})\mathbf{P}_i > \\ =& -2 < \mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i, \mathbf{T}_j^{-1}\mathbf{H}\mathbf{P}_i > \\ =& -2\,\mathrm{Tr}((\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i)^T\mathbf{T}_j^{-1}\mathbf{H}\mathbf{P}_i) \\ =& -2\,\mathrm{Tr}(\mathbf{P}_i(\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i)^T\mathbf{T}_j^{-1}\mathbf{H}) \\ =& -2 < \mathbf{T}_j^{-T}(\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i)\mathbf{P}_i^T, \mathbf{H} > \end{aligned} \tag{A.7}$$

Thus,

$$\frac{\partial O_{ji}}{\mathbf{T}_i} = -2\mathbf{T}_j^{-T}(\mathbf{P}_j - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{P}_i)\mathbf{P}_i^T \tag{A.8}$$

Note, the latter term involves the inverse of $\mathbf{T}_i$, the trick $(A+H)^{-1} = A^{-1} - A^{-1}HA^{-1} + O(H)$ is then used. The gradient w.r.t. $\mathbf{T}_i$ of O_{ij} is then obtained as

$$\frac{\partial O_{ij}}{T_i} = 2\mathbf{T}_i^{-T}(\mathbf{P}_i - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{P}_j)\mathbf{P}_j^T\mathbf{T}_j^T\mathbf{T}_i^{-T} \tag{A.9}$$

A.2.2 Solving for basis commutativity constraints

Let $\mathbf{C}_{j,i}$ be the functional mapping matrix from shape i to shape j, and $\mathbf{Q}_i$ be the low-rank representation of Laplacian-Beltrami basis of shape i. LB basis commutate under functional maps, i.e. $\|\mathbf{Q}_i\mathbf{C}_{j,i} - \mathbf{C}_{j,i}\mathbf{Q}_j\|_F^2 = 0$. In the groupwise functional map, the objective of this constraint could be written as:

$$\begin{aligned} O &= \sum_{i,j} \|\mathbf{Q}_j(\mathbf{T}_i^{-1}\mathbf{T}_j) - (\mathbf{T}_i^{-1}\mathbf{T}_j)\mathbf{Q}_i\|_F^2 \\ &= \sum_{i,j} \|\mathbf{Q}_i(\mathbf{T}_j^{-1}\mathbf{T}_i) - (\mathbf{T}_j^{-1}\mathbf{T}_i)\mathbf{Q}_j\|_F^2 + \sum_{i,j} \|\mathbf{Q}_j(\mathbf{T}_i^{-1}\mathbf{T}_j) - (\mathbf{T}_i^{-1}\mathbf{T}_j)\mathbf{Q}_i\|_F^2 \\ &= O_{ji} + O_{ij} \end{aligned} \tag{A.10}$$

Now, we separately deviate the gradient of O_{ji} and O_{ij}. The objective is:

$$\begin{aligned} O_{ji} &= \mathrm{Tr}\,(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j)^T(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j) \\ &=< \mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j, \mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j > \end{aligned} \tag{A.11}$$

The gradient is:

$$\begin{aligned} &\frac{\partial O_{ji}}{\partial \mathbf{T}_i} \\ &=< \mathbf{Q}_i\mathbf{T}_j^{-1}(\mathbf{T}_i+\mathbf{H}) - \mathbf{T}_j^{-1}(\mathbf{T}_i+\mathbf{H})\mathbf{Q}_j, \mathbf{Q}_i\mathbf{T}_j^{-1}(\mathbf{T}_i+\mathbf{H}) - \mathbf{T}_j^{-1}(\mathbf{T}_i+\mathbf{H})\mathbf{Q}_j > \\ &= 2\,\mathrm{Tr}\,(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j)^T(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j) \\ &= 2< \mathbf{T}_j^{-1}\mathbf{Q}_i^T(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j) - \mathbf{T}_j^{-T}(\mathbf{Q}_i\mathbf{T}_j^{-1}\mathbf{T}_i - \mathbf{T}_j^{-1}\mathbf{T}_i\mathbf{Q}_j)\mathbf{Q}_j^T, H > \end{aligned} \tag{A.12}$$

Similarly, we get $\frac{\partial O_{ij}}{\mathbf{T}_i}$ as follows:

$$\begin{aligned} O_{ij} &= \mathrm{Tr}\,(\mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i)^T(\mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i) \\ &=< \mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i, \mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i > \end{aligned} \tag{A.13}$$

The gradient is:

$$\begin{aligned} \frac{\partial O_{ij}}{\partial \mathbf{T}_i} &=< \mathbf{Q}_j(\mathbf{T}_i+\mathbf{H})^{-1}\mathbf{T}_j - (\mathbf{T}_i+\mathbf{H})^{-1}\mathbf{T}_j\mathbf{Q}_i, \mathbf{Q}_j(\mathbf{T}_i+\mathbf{H})^{-1}\mathbf{T}_j - (\mathbf{T}_i+\mathbf{H})^{-1}\mathbf{T}_j\mathbf{Q}_i > \\ &= 2 < \mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i, \mathbf{T}_i^{-1}\mathbf{H}\mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i - \mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{H}\mathbf{T}_i^{-1}\mathbf{T}_j > \\ &= 2 < T_i^{-T}(\mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i)\mathbf{Q}_i^T\mathbf{T}_j^T\mathbf{T}_i^{-T} - \mathbf{T}_i^{-T}\mathbf{Q}_j^T(\mathbf{Q}_j\mathbf{T}_i^{-1}\mathbf{T}_j - \mathbf{T}_i^{-1}\mathbf{T}_j\mathbf{Q}_i)\mathbf{T}_j^{-T}\mathbf{T}_i^{-T}, \mathbf{H} > \end{aligned} \tag{A.14}$$

BIBLIOGRAPHY

[1] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth, "Time-discrete geodesics in the space of shells," *Computer Graphics Forum (Proceedings of SGP)*, vol. 31, no. 5, pp. 1755–1764, 2012.

[2] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3794–3801.

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," in *ACM Trans. Graph.*, vol. 24, no. 3, 2005, pp. 408–416.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[5] L. Gao, Y.-K. Lai, D. Liang, S.-Y. Chen, and S. Xia, "Efficient and flexible deformation representation for data-driven surface modeling," *ACM Trans. Graph.*, vol. 35, no. 5, p. 158, 2016.

[6] O. Freifeld and M. J. Black, "Lie bodies: A manifold representation of 3d human shape," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 1–14.

[7] G. Stratou, A. Ghosh, P. Debevec, and L. Morency, "Effect of illumination on automatic expression recognition: a novel 3D relightable facial database," in *Proc. Face and Gesture*, 2011, pp. 611–618.

[8] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. AVSS*. IEEE, 2009, pp. 296–301.

[9] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[10] M. Loper, N. Mahmood, and M. J. Black, "MoSh: Motion and shape capture from sparse markers," *ACM Trans. Graph.*, vol. 33, no. 6, p. 220, 2014.

[11] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black, "Dyna: A model of dynamic human shape in motion," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 120:1–120:14, 2015.

[12] O. Sorkine and M. Alexa, "As-rigid-as-possible surface modeling," in *Proc. Eurographics Symposium on Geometry Processing*, 2007, pp. 109–116.

[13] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Trans. Graph.*, vol. 26, no. 3, p. 80, 2007.

[14] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović, "Mesh-based inverse kinematics," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 488–495, 2005.

[15] S. Fröhlich and M. Botsch, "Example-driven deformations based on discrete shells," *Computer Graphics Forum 30*, vol. 8, pp. 2246–2257, 2011.

[16] L. Gao, Y.-K. Lai, J. Yang, L.-X. Zhang, L. Kobbelt, and S. Xia, "Sparse data driven mesh deformation," *arXiv preprint arXiv:1709.01250*, 2017.

[17] J. Yang, J.-S. Franco, F. Hétroy-Wheeler, and S. Wuhrer, "Estimation of Human Body Shape in Motion with Wide Clothing," in *European Conference on Computer Vision 2016*, Amsterdam, Netherlands, Oct. 2016. [Online]. Available: https://hal.inria.fr/hal-01344795

[18] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 98, 2008.

[19] S. Wuhrer, L. Pishchulin, A. Brunton, C. Shu, and J. Lang, "Estimation of human body shape and posture under clothing," *Computer Vision and Image Understanding*, vol. 127, pp. 31–42, 2014.

[20] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[21] T. Cootes, E. Baldock, and J. Graham, "An introduction to active shape models," *Image processing and analysis*, pp. 223–248, 2000.

[22] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," in *ACM Siggraph 2006 Courses*. ACM, 2006, p. 17.

[23] C. Theobalt, M. A. Magnor, P. Schüler, and H.-P. Seidel, "Combining 2d feature tracking and volume reconstruction for online video-based human motion capture," *International Journal of Image and Graphics*, vol. 4, no. 04, pp. 563–583, 2004.

[24] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[25] J. C. Gower, "Generalized Procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[26] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[27] V. Kraevoy, A. Sheffer, and C. Gotsman, *Matchmaker: constructing constrained texture maps*. ACM, 2003, vol. 22, no. 3.

[28] M. Alexa, "Recent advances in mesh morphing," in *Computer graphics forum*, vol. 21, no. 2. Wiley Online Library, 2002, pp. 173–198.

[29] V. Jain and H. Zhang, "Robust 3d shape correspondence in the spectral domain," in *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*. IEEE, 2006, pp. 19–19.

[30] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," in *Computer Graphics Forum*, vol. 30, no. 6. Wiley Online Library, 2011, pp. 1681–1707.

[31] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching," *Proceedings of the National Academy of Sciences*, vol. 103, no. 5, pp. 1168–1172, 2006.

[32] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas, "Non-rigid registration under isometric deformations," in *Computer Graphics Forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1449–1457.

[33] Y. Lipman and T. Funkhouser, "Möbius voting for surface correspondence," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 72, 2009.

[34] O. Kin-Chung Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, and H. Fu, "Electors voting for fast automatic shape correspondence," in *Computer Graphics Forum*, vol. 29, no. 2. Wiley Online Library, 2010, pp. 645–654.

[35] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel," in *Computer Graphics Forum*, vol. 29, no. 5. Wiley Online Library, 2010, pp. 1555–1564.

[36] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 79.

[37] A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel, "Intrinsic shape matching by planned landmark sampling," in *Computer Graphics Forum*, vol. 30, no. 2. Wiley Online Library, 2011, pp. 543–552.

[38] Y. Sahillioğlu and Y. Yemez, "Coarse-to-fine combinatorial matching for dense isometric shape correspondence," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1461–1470, 2011.

[39] A. Golovinskiy and T. Funkhouser, "Consistent segmentation of 3d models," *Computers & Graphics*, vol. 33, no. 3, pp. 262–269, 2009.

[40] K. Xu, H. Li, H. Zhang, D. Cohen-Or, Y. Xiong, and Z.-Q. Cheng, "Style-content separation by anisotropic part scales," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6. ACM, 2010, p. 184.

[41] J. Pokrass, A. M. Bronstein, and M. M. Bronstein, "A correspondence-less approach to matching of deformable shapes," in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2011, pp. 592–603.

[42] Q. Huang, V. Koltun, and L. Guibas, "Joint shape segmentation with linear programming," in *ACM transactions on graphics (TOG)*, vol. 30, no. 6. ACM, 2011, p. 125.

[43] O. Van Kaick, A. Tagliasacchi, O. Sidi, H. Zhang, D. Cohen-Or, L. Wolf, and G. Hamarneh, "Prior knowledge for part correspondence," in *Computer Graphics Forum*, vol. 30, no. 2. Wiley Online Library, 2011, pp. 553–562.

[44] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. Van Kaick, and A. Tagliasacchi, "Deformation-driven shape correspondence," in *Computer Graphics Forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1431–1439.

[45] I.-C. Yeh, C.-H. Lin, O. Sorkine, and T.-Y. Lee, "Template-based 3d model fitting using dual-domain relaxation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 8, pp. 1178–1190, 2011.

[46] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 30, 2012.

[47] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel, "Coupled quasi-harmonic bases," in *Computer Graphics Forum*, vol. 32, no. 2pt4. Wiley Online Library, 2013, pp. 439–448.

[48] É. Corman, M. Ovsjanikov, and A. Chambolle, "Supervised descriptor learning for non-rigid shape matching," in *Computer Vision-ECCV 2014 Workshops*. Springer, 2014, pp. 283–298.

[49] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro, "Sparse modeling of intrinsic correspondences," in *Computer Graphics Forum*, vol. 32, no. 2pt4. Wiley Online Library, 2013, pp. 459–468.

[50] Q. Huang, F. Wang, and L. Guibas, "Functional map networks for analyzing and exploring large shape collections," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 36, 2014.

[51] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers, "Partial functional correspondence," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 222–236.

[52] E. Rodola, M. Moeller, and D. Cremers, "Point-wise map recovery and refinement from functional correspondence," *arXiv preprint arXiv:1506.05603*, 2015.

[53] E. Rodolà, M. Moeller, and D. Cremers, "Regularized pointwise map recovery from functional correspondence," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 700–711.

[54] O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, and D. Cremers, "Non-rigid puzzles," in *Computer Graphics Forum*, vol. 35, no. 5. Wiley Online Library, 2016, pp. 135–143.

[55] D. Nogneng and M. Ovsjanikov, "Informative descriptor preservation via commutativity for shape matching," in *Computer Graphics Forum*, vol. 36, no. 2. Wiley Online Library, 2017, pp. 259–267.

[56] D. Boscaini, D. Eynard, D. Kourounis, and M. M. Bronstein, "Shape-from-operator: Recovering shapes from intrinsic operators," in *Computer Graphics Forum*, vol. 34, no. 2. Wiley Online Library, 2015, pp. 265–274.

[57] R. M. Rustamov, M. Ovsjanikov, O. Azencot, M. Ben-Chen, F. Chazal, and L. Guibas, "Map-based exploration of intrinsic shape differences and variability," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 72, 2013.

[58] D. F. Huber and M. Hebert, "Automatic three-dimensional modeling from reality," Ph.D. dissertation, Citeseer, 2002.

[59] D. Pachauri, R. Kondor, and V. Singh, "Solving the multi-way matching problem by permutation synchronization," in *Advances in neural information processing systems*, 2013, pp. 1860–1868.

[60] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 569–578, 2006.

[61] C. Zach, M. Klopschitz, and M. Pollefeys, "Disambiguating visual relations using loop constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1426–1433.

[62] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas, "An optimization approach to improving collections of shape maps," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1481–1491, 2011.

[63] Q.-X. Huang, G.-X. Zhang, L. Gao, S.-M. Hu, A. Butscher, and L. Guibas, "An optimization approach for extracting and encoding consistent maps in a shape collection," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 167, 2012.

[64] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor, "A minimum description length approach to statistical shape modeling," *IEEE Trans. Medical Imaging*, vol. 21, no. 5, pp. 525–537, 2002.

[65] Q.-X. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," *Computer Graphics Forum*, vol. 32, no. 5, pp. 177–186, 2013.

[66] Y. Sahillioğlu and Y. Yemez, "Multiple shape correspondence by dynamic programming," in *Computer Graphics Forum*, vol. 33, no. 7. Wiley Online Library, 2014, pp. 121–130.

[67] L. Cosmo, E. Rodolà, A. Albarelli, F. Mémoli, and D. Cremers, "Consistent partial matching of shape collections via sparse modeling," in *Computer Graphics Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 209–221.

[68] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proc. SIGGRAPH*, vol. 21, no. 4, 1987, pp. 205–214.

[69] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder, "Discrete shells," in *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.

[70] P. von Radziewsky, E. Eisemann, H.-P. Seidel, and K. Hildebrandt, "Optimized subspaces for deformation-based modeling and shape interpolation," *Computers & Graphics*, vol. 58, pp. 128–138, 2016.

[71] C. von Tycowicz, C. Schulz, H.-P. Seidel, and K. Hildebrandt, "Real-time nonlinear shape interpolation," *ACM Trans. Graph.*, vol. 34, no. 3, pp. 34:1–34:10, 2015.

[72] T. Heap and D. Hogg, "Extending the point distribution model using polar coordinates," *Image Vis. Comput.*, vol. 14, no. 8, pp. 589 – 599, 1996.

[73] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172.

[74] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, 2004.

[75] B. Allen, B. Curless, Z. Popović, and A. Hertzmann, "Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis," in *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, pp. 147–156.

[76] D. Hirshberg, M. Loper, E. Rachlin, and M. Black, "Coregistration: Simultaneous alignment and modeling of articulated 3d shape," in *Proc. ECCV*, 2012, pp. 242–255.

[77] Y. Chen, Z. Liu, and Z. Zhang, "Tensor-based human body modeling," in *Proc. CVPR*, 2013, pp. 105–112.

[78] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel, "A statistical model of human pose and body shape," in *Comput. Graph. Forum*, vol. 28, no. 2, 2009, pp. 337–346.

[79] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 8, pp. 995–1005, 2004.

[80] M. Kilian, N. J. Mitra, and H. Pottmann, "Geometric modeling in shape space," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 64, 2007.

[81] X. Pennec, "Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 1, pp. 127–154, 2006.

[82] B. Heeren, M. Rumpf, P. Schröder, M. Wardetzky, and B. Wirth, "Exploring the geometry of the space of shells," *Computer Graphics Forum (Proceedings of SGP)*, vol. 33, no. 5, pp. 247–256, 2014.

[83] M. Tournier, X. Wu, N. Courty, E. Arnaud, and L. Reveret, "Motion compression using principal geodesics analysis," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 355–364.

[84] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen, "Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 43–56.

[85] S. Huckemann, T. Hotz, and A. Munk, "Intrinsic shape analysis: Geodesic PCA for riemannian manifolds modulo isometric lie group actions," *Statistica Sinica*, pp. 1–58, 2010.

[86] M. Zhang and P. T. Fletcher, "Probabilistic principal geodesic analysis," in *Advances in Neural Information Processing Systems*, 2013, pp. 1178–1186.

[87] S. Hauberg, "Principal curves on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1915–1921, 2016.

[88] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in euclidean spaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 7, pp. 1415–1428, 2011.

[89] S. Kurtek, E. Klassen, J. C. Gore, Z. Ding, and A. Srivastava, "Elastic geodesic paths in shape space of parameterized surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1717–1730, 2012.

[90] G. G. Demisse, D. Aouada, and B. Ottersten, "Deformation based curved shape representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[91] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter, "Reconstructing high quality face-surfaces using model based stereo," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[92] B. Amberg, R. Knothe, and T. Vetter, "Expression invariant 3d face recognition with a morphable model," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 2008, pp. 1–6.

[93] F. B. ter Haar and R. C. Veltkamp, "3d face model fitting for recognition," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 652–664.

[94] C. Basso, A. Verri, and J. Herder, "Fitting 3d morphable models using implicit representations," *Journal of Virtual Reality and Broadcasting*, vol. 4, no. 18, pp. 1–10, 2007.

[95] I. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, T. Theoharis *et al.*, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 4, pp. 640–649, 2007.

[96] M. De Smet and L. Van Gool, "Optimal regions for linear model-based 3d face reconstruction," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 276–289.

[97] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 426–433.

[98] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, p. 130, 2011.

[99] I. Mpiperis, S. Malassiotis, and M. G. Strintzis, "Bilinear models for 3-d face and facial expression recognition," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 498–511, 2008.

[100] T. Bolkart and S. Wuhrer, "Statistical analysis of 3d faces in motion," in *3D Vision-3DV 2013, 2013 International Conference on*. IEEE, 2013, pp. 103–110.

[101] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3d shape regression for real-time facial animation," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 41, 2013.

[102] A. Brunton, T. Bolkart, and S. Wuhrer, "Multilinear wavelets: A statistical shape space for human faces," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[103] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1063–1074, 2003.

[104] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt, "Moviereshape: Tracking and reshaping of humans in videos," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6. ACM, 2010, p. 148.

[105] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele, "Building statistical shape spaces for 3d human modeling," *Pattern Recognition*, vol. 67, pp. 276–286, 2017.

[106] S. I. Park and J. K. Hodgins, "Data-driven modeling of skin and muscle deformation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 96.

[107] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *European Conference on Computer Vision*. Springer, 2012, pp. 640–653.

[108] D. Tzionas and J. Gall, "3d object reconstruction from hand-object interactions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 729–737.

[109] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 143, 2016.

[110] S. Zuffi, A. Kanazawa, D. Jacobs, and M. J. Black, "3d menagerie: Modeling the 3d shape and pose of animals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6365–6373.

[111] E. Corman, J. Solomon, M. Ben-Chen, L. Guibas, and M. Ovsjanikov, "Functional characterization of intrinsic and extrinsic geometry," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 2, p. 14, 2017.

[112] H. LeDret and A. Raoult, "The nonlinear membrane model as a variational limit of nonlinear three-dimensional elasticity," *J. Math. Pures Appl.*, vol. 73, pp. 549–578, 1995.

[113] G. Friesecke, R. James, M. G. Mora, and S. Müller, "Derivation of nonlinear bending theory for shells from three-dimensional nonlinear elasticity by Gamma-convergence," *C. R. Math. Acad. Sci. Paris 336*, vol. 8, pp. 697–702, 2003.

[114] M. Rumpf and B. Wirth, "An elasticity-based covariance analysis of shapes," *International Journal of Computer Vision*, vol. 92, no. 3, pp. 281–295, 2011.

[115] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York / Berlin: Springer, 1999.

[116] H. Hoppe, "Progressive meshes," *Computer Graphics Proceedings*, 1996.

[117] M. Garland and P. Heckbert, "Surface simplification using quadric error metrics," *ACM SIGGRAPH*, pp. 209–216, 1997.

[118] G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool, "A 3-d audio-visual corpus of affective communication," *Multimedia, IEEE Transactions on*, vol. 12, no. 6, pp. 591–598, 2010.

[119] D. Rueckert, A. F. Frangi, and J. A. Schnabel, "Automatic construction of 3D statistical deformation models using nonrigid registration," in *Medical Image Computing and Computer–Assisted Intervention, MICCAI*, ser. LNCS, W. Niessen and M. Viergever, Eds., vol. 2208, 2001, pp. 77–84.

[120] K. K. Bhatia, J. V. Hajnal, B. K. Puri, A. D. Edwards, and D. Rueckert, "Consistent groupwise non–rigid registration for atlas construction," in *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, vol. 1, 2004, pp. 908–911.

[121] S. Joshi, B. Davis, M. Jomier, and G. Gerig, "Unbiased diffeomorphic atlas construction for computational anatomy," *NeuroImage*, vol. 23, pp. 151–160, 2004, supplement 1.

[122] C. Studholme, "Simultaneous population based image alignment for template free spatial normalisation of brain anatomy," in *Second International Workshop, WBIR, Biomedical Image Registration*, ser. LNCS, J. C. Gee, J. B. A. Maintz, and M. W. Vannier, Eds., vol. 2717, 2003, pp. 81–90.

[123] S. Marsland, C. J. Twining, and C. J. Taylor, "Groupwise non–rigid registration using polyharmonic clamped–plate splines," in *Medical Image Computing and Computer–Assisted Intervention, MICCAI*, ser. LNCS, R. E. Ellis and T. M. Peters, Eds., vol. 2879, 2003, pp. 771–779.

[124] K. K. Bhatia, J. V. Hajnal, A. Hammers, and D. Rueckert, "Similarity metrics for groupwise non–rigid registration," in *Medical Image Computing and Computer–Assisted Intervention, MICCAI 2007*, ser. LNCS, N. Ayache, S. Ourselin, and A. Maeder, Eds., vol. 4792, 2007, pp. 544–552.

[125] D. G. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984.

[126] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," in *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 587–594.

[127] M. Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié," *Ann. Inst. H. Poincaré*, vol. 10, pp. 215–310, 1948.

[128] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Comm. Pure Appl. Math.*, vol. 30, no. 5, pp. 509–541, 1977.

[129] M. Rumpf and B. Wirth, "Variational time discretization of geodesic calculus," *IMA J. Numer. Anal.*, vol. 35, no. 3, pp. 1011–1046, 2015.

[130] P. Korshunov and T. Ebrahimi, "Using face morphing to protect privacy," in *Proc. AVSS*, 2013, pp. 208–213.

[131] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: High-resolution capture for modeling and animation," in *Data-Driven 3D Facial Animation*. Springer, 2008, pp. 248–276.

[132] G. Fyffe, A. Jones, O. Alexander, R. Ichikari, and P. Debevec, "Driving high-resolution facial scans with video performance capture," *ACM Trans. Graphic. (Proceedings of SIGGRAPH)*, vol. 34, no. 1, 2014.

[133] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. ICCV workshop on Automatic Facial Landmark Detection in-the-Wild Challenge*, 2013, pp. 397–403.

[134] M. Mauro, K. Khan, and R. Leonardi, "Multi-class semantic segmentation of faces," in *Proc. ICIP*, 2015.

[135] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 351–358.

[136] B. Lévy, "Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry," in *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*. IEEE, 2006, pp. 13–13.

[137] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental mathematics*, vol. 2, no. 1, pp. 15–36, 1993.

[138] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and mathematics III*. Springer, 2003, pp. 35–57.

[139] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, pp. 1455–1459, 2014. [Online]. Available: http://www.manopt.org

[140] J. J. Koenderink and A. J. van Doorn, "Surface shape and curvature scales," *Image and Vision Computing*, vol. 10, pp. 557–565, 1992.

[141] P. Nair and A. Cavallaro, "Matching 3d faces with partial data," in *Proc. BMVC*, 2008.

[142] N. Tsumura, N. Ojima, K. Sato, M. Shiraishi, H. Shimizu, H. Nabeshima, and Y. Miyake, "Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin," *ACM Trans. Graphic.*, vol. 22, no. 3, pp. 770–779, 2003.

[143] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3d facial expression database for facial behavior research," in *Proc. Face and Gesture*. IEEE, 2006, pp. 211–216.

[144] M. Ovsjanikov, M. Ben-Chen, F. Chazal, and L. Guibas, "Analysis and visualization of maps between shapes," *Computer Graphics Forum*, vol. 32, no. 6, pp. 135–145, 2013.

[145] A. O. Bălan and M. J. Black, "The naked truth: Estimating body shape under clothing," in *European Conference on Computer Vision*. Springer, 2008, pp. 15–29.

[146] G. Pons-Moll, S. Pujades, S. Hu, and M. Black, "ClothCap: Seamless 4D clothing capture and retargeting," *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, vol. 36, no. 4, 2017. [Online]. Available: http://dx.doi.org/10.1145/3072959.3073711

[147] M. Loper, "Chumpy is a python-based framework designed to handle the auto-differentiation problem," htttps://pypi.python.org/pypi/chumpy, 2015.

[148] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[149] S. Zuffi and M. J. Black, "The stitched puppet: A graphical model of 3d human shape and pose," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 3537–3546.