UNIVERSITAT POLITÈCNICA DE CATALUNYA

COMPUTER SCIENCE DEPARTMENT

Extending Procrustes Analysis: Building Multi-view 2-D Models from 3-D Human Shape Samples

A thesis submitted for the degree of
Doctor of Philosophy

Xavier Pérez Sala

Barcelona, March 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA

A thesis submitted for the degree of
Doctor of Philosophy

Extending Procrustes Analysis: Building Multi-view 2-D Models from 3-D Human Shape Samples

Xavier Pérez Sala

Advisors:
Cecilio Angulo Bahón
UPC ·BarcelonaTech
Sergio Escalera Guerrero
Universitat de Barcelona

Supervisor:
Antoni Yuste Marco
Fundació Privada Sant Antoni Abat

This work has been partially supported by SUR, Departament d'Economia i Coneixement, and Comissionat per a Universitats i Recerca del Departament d'Innovació, Universitats i Empresa de la Generalitat de Catalunya, with the programs BE and TEM-DGR at Fundació Privada Sant Antoni Abat.

Als meus pares, A&P. Cap agraïment és suficent...

Abstract

This dissertation formalizes the construction of multi-view 2-D shape models from 3-D data, by means of several extensions of the well-known Procrustes Analysis (PA) algorithm. The proposed extensions allow modeling rigid and non-rigid transformations in an efficient manner, and they are successfully tested on faces and human bodies datasets.

In human perception applications one can set physical restrictions, such as defining faces and human skeletons as sets of anatomical landmarks or articulated bodies. However, looking at people carries difficult tasks. The high variation of facial expressions and human postures from different viewpoints makes problems like face tracking or human pose estimation extremely challenging. The common approach to handle large viewpoint variations is training the models with several labeled images from different viewpoints. However, this approach has several important drawbacks: (1) it is not clear the extent to which the dataset must be enhanced with images from different viewpoints in order to build unbiased 2-D models; (2) extending the training set without this evaluation would unnecessarily increase memory and computation requirements to train the models; and (3) obtaining new labeled images from different viewpoints can be a difficult task because of the expensive labeling cost; finally, (4) a non-uniform coverage of the different viewpoints of a person leads to biased 2-D models. In this dissertation we propose successive extensions of PA to address these issues.

First of all, we propose Projected Procrustes Analysis (PPA) as a formalization for building multi-view 2-D rigid models by rotating 3-D datasets. PPA rotates and projects every 3-D training shape and builds a multi-view 2-D model from this enhanced training set. We also introduce common parametrizations of rotations, as well as mechanisms to uniformly sample the rotation space to build unbiased 2-D models. We show that uniformly distributed rotations generate unbiased models, while non-uniform rotations lead to models representing some viewpoints better than others.

Although PPA has been successful in building multi-view 2-D models, it requieres an enhanced dataset that increases the computational require-

ments in space and time. To address these PA and PPA drawbacks and to build unbiased 2-D models in an efficient manner, we propose Continuous Procrustes Analysis (CPA). CPA extends PA and PPA within a functional analysis framework and constructs multi-view 2-D rigid models in an efficient way through integrating all possible rotations in a given domain. We show that CPA models are inherently unbiased because of their integral formulation. However, CPA is not able to capture non-rigid deformations from the dataset.

Next, in order to efficiently compute multi-view 2-D deformable models from 3-D data, we propose Subspace Procrustes Analysis (SPA). By adding a subspace in the PA formulation, SPA is able to model non-rigid deformations, as well as rigid 3-D transformations of the training set. We developed a discrete (DSPA) and continuous (CSPA) formulation to provide a better understanding of the problem, where DSPA samples and CSPA integrates the 3-D rotation space.

Finally, we illustrate the benefits of our multi-view 2-D deformable models in the task of human pose estimation. We first reformulate the problem as a feature selection by subspace matching, and we propose an efficient approach for this task. Our proposed method is much more efficient than the state-of-the-art feature selection by subspace matching approaches, and it is able to handle larger number of outliers. Next, we show that our multi-view 2-D deformable models, combined with the subspace matching method, outperform state-of-the-art methods of human pose estimation. Our approach is more accurate in the joint positions and limb lengths because we use unbiased 2-D models trained on 3-D Motion Capture datasets. Our models are not biased to any particular point of view and they can successfully reconstruct different non-rigid deformations and viewpoints. Moreover, they are efficient in learning, as well as in test time.

Acknowledgments

First of all I would like to thank my advisors Cecilio Angulo and Sergio Escalera for the guidance and assistance during these years of research. Without their support and research experience I would never have been able to finish this work on time. I would also like to thank Cecilio for his flexibility with the research path that is summarized in this dissertation. I thank both of them for not only being my advisors but also my mentors who introduced me to the world of research.

I am grateful to Laura Igual and Fernando De La Torre for how involved they have been in most of the steps of this dissertation. Without them, this thesis would look really different.

I also would like to thank the team of Fundació Privada Sant Antoni Abat, and specially to Antoni Yuste and Esther Valldosera, who made possible these three years of work.

During these years I have been lucky to work in different research teams. I met great people who helped me or contributed at different points of my research life with nice ideas, technical support and good conversations in front of a blackboard. Since any ordering will be unfair, let's start from the beginning. I am grateful to Albert Samà, Dani Rodríguez, Oscar Franco and the other members of CETpD. I am also thankful to Toni Hernández, Miguel Ángel Bautista, Albert Clapés, Víctor Ponze, Miguel Reyes and the rest of the HuPBA group. I would like to show my gratitude to Francisco Vicente, Rachel Burcin, Marc Estruch, Ricardo Cabral, Feng Zhou, Zehua Huang, Pablo Navarro and the other members of Human Sensing Laboratory.

Finally, but not less important, I would like to thank my family and friends for all these years of unconditional support, specially to my parents Anna and José, Just, Núria, and Lluís. Without them, all this effort would not have made sense. Of course, I am also grateful to Laura for reading these esoteric lines and for being my motivation to finish them on time.

Actually I am also thankful to you. This dissertation is the result of a long effort. Whoever you are reading this work, I am grateful to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Perceiving human beings and their activities has been a fascinating and challenging endeavor ever since Eadweard Muybridge took the first pictures of human movements in 1884 [67]. Nowadays the difficulties are different, but looking at people is still a hot topic in artificial intelligence [64, 25]. Not only is it a major challenge in computer vision [60, 75], but it also has important implications in our daily lives as well as future possibilities. Digital cameras, for example, can now find the faces of the subjects and help us take the perfect shot. Computer vision advances can also bring sensing tools to assistive robots helping patients in rehabilitation tasks. Nowadays, wearable sensors provide assistance to elderly by detecting falls at home, and newer sensors can help doctors adjust medication dosage in patients suffering from Parkinson's disease. Although recent, these are examples of technological innovations that are becoming real necessities in the daily lives of our communities. Even more, with each day passing, we are demanding the technology to be faster, more accurate, and to make our lives easier. We are asking for technology that can make personalized recommendations based on our location, physical state or emotional condition– all with a unifying concept: We are asking machines for a better human perception.

Human perception in computer vision is mostly focused on bodies [82, 63, 34] and faces [111, 61, 12], with a huge variety of subproblems such as human pose estimation [6, 91, 107, 78, 42], hand pose recovery [24, 8, 69], activity segmentation [100, 5, 87, 11] and recognition [83, 100, 50, 41, 40], face tracking [103, 104], head pose estimation [66, 113, 26], and face landmark localization [113, 112, 86], just to mention a few. Many of them use shape models to represent configurations of face landmarks [61, 86, 113, 112] or body joints [6, 91, 107, 78, 42] in the image. This dissertation is focused on building multi-view 2-D models to this end, having been successfully tested on faces and bodies datasets.

Looking at people allows us to set physical restrictions, i.e. defining faces and human skeletons as sets of anatomical landmarks or articulated bodies. On the other hand, the high variation of facial expressions and human postures, as well as the extremely different appearances that faces and body poses can present from different viewpoints, make visual human perception very challenging. The common approach to handle large viewpoint variations is to train the models with several labeled images from different viewpoints [37, 99, 113, 107, 78, 79, 42]. However, this approach has some drawbacks: (1) it is not clear the extent to which the dataset must be enhanced with images from different viewpoints in order to build unbiased 2-D models; (2) extending the training set without this evaluation would unnecessarily increase memory and computation requirements to train models; (3) obtaining new labeled images from different viewpoints can be a difficult task because of the expensive labeling cost; and finally, (4) a non-uniform coverage of the different viewpoints of a person leads to biased 2-D models (i.e., some poses are better represented than others).

In this dissertation we propose to solve these drawbacks by changing the paradigm, learning 2-D multi-view models from 3-D datasets instead of training them on 2-D images. Recently, several 3-D datasets have been publicly and commercially released, providing 3-D objects of architecture, engineering, automation [4], animals and humans [2], among others. Moreover, the popularization of 3-D cameras and printers, as well as collaborative projects to build datasets of 3-D models [2, 3] confirm that 3-D data is becoming a more accessible resource with each day passing. It is likely that in the near future, public datasets will go beyond rigid objects and provide 3-D motion datasets, as research datasets already do for people performing different actions [16].

We propose using available 3-D datasets to build unbiased 2-D shape models, avoiding the previous drawbacks: (1) We formulate the problem of 2-D model building from 3-D data; (2) we avoid the need of 2-D labeled images; and (3) we provide novel formulations that guarantee the uniform coverage of the rotation space, as well as experimental validation for different datasets of faces and human bodies. Moreover, (4) we propose extremely efficient approaches; and (5) we illustrate the benefits of our unbiased 2-D models in the task of human pose estimation.

The question that arises at this point is why we are building multi-view 2-D models from 3-D data. In other words, why are we projecting 3-D data in training time, instead of learning 3-D models and projecting them onto the image plane during the test? The answer is that 2-D models have the same representation power than 3-D models, with 2-D models being faster in real-time fitting [61]. When thinking in new algorithms, it is important

to take into account that a full 90% of all the data in the world has been generated over the last two years. Going a little bit further, every minute, YouTube users upload 100 hours of video, and Instagram users share $48,600$ new photos. In this context, we need efficient algorithms to handle large amount of data in training as well as in test phases.

In this dissertation we provide novel algorithms for the efficient construction of 2-D multi-view models from 3-D datasets, by means of several extensions of the well known Procrustes Analysis [23, 36, 35] algorithm. These extensions are focused on the construction of unbiased 2-D models, able to generalize among different viewpoints of objects and their rigid and non-rigid transformations (e.g. facial expressions and human postures). In experimental sections we show that the unbiased 2-D models trained with our proposed methods outperform state-of-the-art Procrustes Analysis approaches, in human bodies and faces datasets. First of all, we formalize the problem of building 2-D rigid models from 3-D data, and we detail how to sample the rotation space to build unbiased 2-D models. We call Projected Procrustes Analysis (PPA) to this approach. Next, we focus on efficiency and propose an extension of PPA based on Functional Data Analysis (FDA), Continuous Procrustes Analysis (CPA), which enable us to build multi-view rigid models by integrating all possible viewpoints. Then, we take a step further and build statistical 2-D models with Subspace Procrustes Analysis (SPA). SPA models are able to generalize among rigid and non-rigid transformations of objects, across different viewpoints. We provide two SPA formulations, discrete (DSPA) and continuous (CSPA), by extending PPA and CPA, respectively. Finally, we illustrate the performance of our unbiased 2-D models in the problem of human pose estimation, by means of a novel feature selection method by subspace matching. These successive extensions of PA also determine the structure of the thesis. One chapter is dedicated to each one of these extensions and application. Note that for a better contextualization, state of the art concerning different contributions in this thesis is introduced in each chapter.

1.1 Scope of the Thesis

The main contributions of this Ph.D. thesis are the formalization of building unbiased 2-D models from 3-D data, and the successive extensions of Procrustes Analysis in order to efficiently model rigid and non-rigid transformations of faces and human bodies. We can list the contributions of this dissertation as:

- Study and formulation of multi-view 2-D rigid models from 3-D data

with Projected Procrustes Analysis, as well as the requirements to build unbiased models by uniformly sampling the rotation space.

- Continuous extension of PPA by means of Functional Data Analysis, CPA, which efficiently learns 2-D rigid models able to generalize to different viewpoints.

- Addition of a subspace in the PA formulation, SPA, able to build 2-D deformable models encoding rigid and non-rigid transformations of 3-D datasets, in an efficient manner.

- Application of our unbiased 2-D models to human pose estimation by means of an efficient algorithm of feature selection by subspace matching.

1.2 Thesis Overview

This thesis contains 6 chapters, the remaining of which are organized as follows:

Chapter 2. Procrustes Analysis (PA) is defined and the first PA extension, Projected Procrustes Analysis (PPA), is proposed as a formalization to build 2-D rigid models from rotating and projecting 3-D datasets. Different rotation parametrizations are presented, as well as mechanisms to uniformly sample the rotation space and build unbiased 2-D models.

Chapter 3. Functional Data Analysis (FDA) is introduced as well as the mathematical background to integrate into the rotation space in a uniform manner. Continuous Procrustes Analysis (CPA) is presented as an efficient extension of PPA to build unbiased 2-D rigid models from 3-D data.

Chapter 4. Statistical models are introduced as an approach to model non-rigid deformations with a subspace. Subspace Procrustes Analysis (SPA) is presented as a way to efficiently build 2-D models able to generalize to rigid and non-rigid transformations of objects, across different viewpoints.

Chapter 5. Human pose estimation task is defined and reformulated as a feature selection problem by subspace matching, and a novel method is proposed to this end. Finally, the performance of multi-view 2-D models is illustrated in the task of human pose estimation.

Chapter 6. Concludes the dissertation with a summary of the main contributions and a discussion about the future directions the research might have.

1.3 Resulted Publications

This document compiles and supplements the following papers previously published or submitted in international conferences and journals:

Journal Publications

- L. Igual, X. Perez-Sala, S. Escalera, C. Angulo, F. Dela Torre, *Continuous alternative to generalized procrustes analysis*, in: Pattern Recognition, vol. 47, issue 2, pp. 659–671, 2014.

- X. Perez-Sala, S. Escalera, C. Angulo, *A Survey on Model Based Approaches for Human Pose Recovery*, in: Sensors, vol. 14, issue 3, pp. 4189-4210, 2014.

- A. Hernandez-Vela, M. Bautista, X. Perez-Sala, V. Ponce, S. Escalera, X. Baró, O. Pujol, C. Angulo, *Probability-based Dynamic Time Warping and Bag-of-Visual-and-Depth-Words for Human Gesture Recognition in RGB-D*, in: Pattern Recognition Letters, vol. 50, issue December, pp. 112-121, 2013.

- X. Perez-Sala, F. Dela Torre, L. Igual, S. Escalera, C. Angulo, *Subspace Procrustes Analysis*, in: preparation to submit to International Journal of Computer Vision.

International Conferences and Workshops

- X. Perez-Sala, F. Dela Torre, L. Igual, S. Escalera, C. Angulo, *Subspace Procrustes Analysis*, in: European Conference on Computer Vision (ECCV'2014) Workshop on ChaLearn Looking at People, Zurich, Switzerland.

- X. Perez-Sala, S. Escalera, C. Angulo, *Survey on 2D and 3D Human Pose Recovery*, in: 15th International Conference of the Catalan Association of Artificial Intelligence (CCIA 2012), Alacant, Spain, pp. 101-110. IOS Press.

- A. Hernandez-Vela, M. Bautista, X. Perez-Sala, V. Ponce, X. Baró, O. Pujol, C. Angulo, S. Escalera, *Bovdw: Bag-of-visual-and-depth-words for gesture recognition*, in: 21st Conference of the International Association for Pattern Recognition (ICPR 2012), Tsukuba, Japan, pp. 449-452.

- M. Bautista, A. Hernandez-Vela, X. Perez-Sala, V. Ponce, X. Baró, O. Pujol, C. Angulo, S. Escalera, *Probability-based Dynamic Time Warping for Gesture Recognition*, in: International Workshop on Depth Image Analysis (WDIA 2012), Tsukuba, Japan.

- X. Perez-Sala, C. Angulo, S. Escalera, *Biologically inspired path execution using surf flow in robot navigation*, in: Lecture Notes in Computer Science, vol. 6692, pp. 581-588 (IWANN'2011), Torremolinos, Spain. Springer.

- X. Perez-Sala, C. Angulo Bahón, S. Escalera, *Biologically inspired turn control for autonomous mobile robots*, in: 14th International Conference of the Catalan Association of Artificial Intelligence (CCIA 2011), Lleida, Spain, pp. 189-198. IOS Press.

Book Chapters

- X. Perez-Sala, L. Igual, S. Escalera, C. Angulo, *Uniform Sampling of Rotations for Discrete and Continuous Learning of 2D Shape Models*, in: Robotic Vision: Technologies for Machine Learning and Vision Applications. Chapter 2, pp. 23-42. IGI Publications, 2012. (doi 10.4018/978-1-4666-2672-0.ch002)

1.4 Notation

In this section we briefly introduce the notation we will use along this dissertation, with an special emphasis to the vec-transpose operator.

Bold capital letters denote a matrix $\mathbf{A}$, bold lower-case letters a column vector $\mathbf{a}$. $\mathbf{a}_i$ represents the i^{th} column of the matrix $\mathbf{A}$. a_{ij} denotes the scalar in the i^{th} row and j^{th} column of the matrix $\mathbf{A}$. All non-bold letters represent scalars. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix. $\|\mathbf{a}\|_2 = \sqrt[2]{\sum_i |a_i|^2}$ and $\|\mathbf{A}\|_F = \sqrt{\sum_{ij} a_{ij}^2}$ denote the 2-norm for a vector and the Frobenius norm of a matrix, respectively. $\mathbf{A} \otimes \mathbf{B}$ is the Kronecker product of matrices and $\mathbf{A}^{(p)}$ is the vec-transpose operator.

Vec-transpose Operator. Vec-transpose $\mathbf{A}^{(p)}$ is a linear operator that generalizes vectorization and transposition operators [58, 62]. It reshapes matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ by vectorizing each i^{th} block of p rows, and rearranging it as the i^{th} column of the reshaped matrix, such that $\mathbf{A}^{(p)} \in \mathbb{R}^{pn \times \frac{m}{p}}$:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \\ a_{61} & a_{62} & a_{63} \end{bmatrix}^{(2)} = \begin{bmatrix} a_{11} & a_{31} & a_{51} \\ a_{21} & a_{41} & a_{61} \\ a_{12} & a_{32} & a_{52} \\ a_{22} & a_{42} & a_{62} \\ a_{13} & a_{33} & a_{53} \\ a_{23} & a_{43} & a_{63} \end{bmatrix},$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \\ a_{61} & a_{62} & a_{63} \end{bmatrix}^{(3)} = \begin{bmatrix} a_{11} & a_{41} \\ a_{21} & a_{51} \\ a_{31} & a_{61} \\ a_{12} & a_{42} \\ a_{22} & a_{52} \\ a_{32} & a_{62} \\ a_{13} & a_{43} \\ a_{23} & a_{53} \\ a_{33} & a_{63} \end{bmatrix}.$$

Note that $\mathbf{A}^{(m)} = \text{vec}(\mathbf{A})$, $\mathbf{A}^{(1)} = \mathbf{A}^T$, and $(\mathbf{A}^{(p)})^{(p)} = \mathbf{A}$. Then, the well known expression:

$$(\mathbf{B}^T \otimes \mathbf{A})\,\text{vec}(\mathbf{C}) = \text{vec}(\mathbf{ACB}),$$

generalizes [62] to:

$$((\mathbf{BA})^{(p)}\mathbf{C})^{(p)} = (\mathbf{C}^T \otimes \mathbf{I}_p)\mathbf{BA} = (\mathbf{B}^{(p)}\mathbf{C})^{(p)}\mathbf{A}.$$

A useful rule for pulling a matrix out of nested Kronecker products, which leads to:

$$(\mathbf{C}^T \otimes \mathbf{I}_2)\mathbf{B} = (\mathbf{B}^{(2)}\mathbf{C})^{(2)}.$$

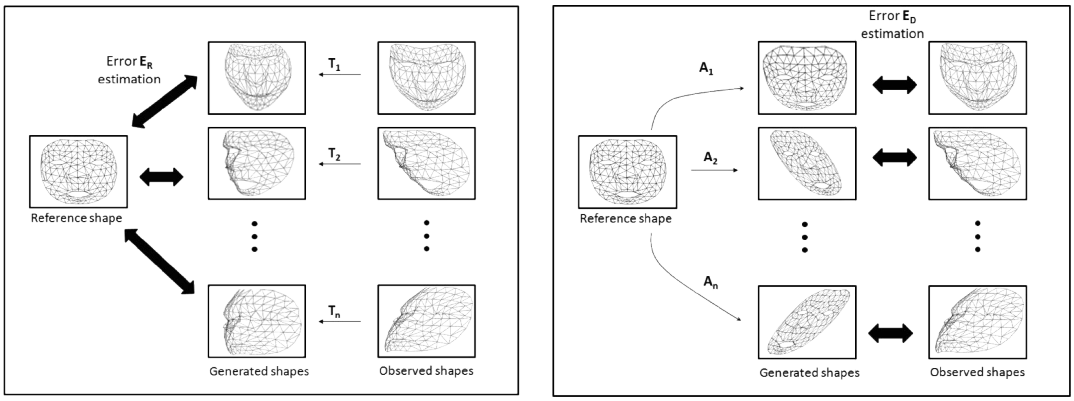
Chapter 2

Procrustes Analysis Revisited

In this chapter, we first define Procrustes Analysis (PA), as well as its state of the art. Then, we introduce our first PA extension, Projected Procrustes Analysis (PPA), as a formalization to build 2-D rigid models by rotating and projecting 3-D datasets. Next, different parametrizations of rotations are presented, as well as mechanisms to uniformly sample the rotation space and build unbiased 2-D models. Finally, we show the benefits of our approach in datasets of faces and bodies.

2.1 Procrustes Analysis

Procrustes Analysis (PA) [23, 36, 35] is an statistical method used to study the distribution of a set of shapes. Given two shapes and their landmark correspondences, PA "superimposes" both shapes by optimally translating, rotating and scaling one shape towards the other. When more than two shapes are registered, the problem is typically known as Generalized Procrustes Analysis (GPA). GPA simultaneously finds the best registration between each shape and the mean or reference shape, meanwhile the reference shape is being computed. Although we are aligning multiple shapes in this thesis, for simplicity on naming our multiple extensions we will shortly refer to this problem as PA in the remaining of this dissertation. PA has been typically used in computer vision as a first step to build 2-D models of shape or appearance of objects. These 2-D models have been applied to solve problems such as object recognition [98, 49], facial feature detection and tracking [96, 17], and image segmentation [72, 65]. In particular, Point Distribution Models (PDMs) and Active Shape Model (ASMs) [18] are among the most popular techniques to learn 2-D object models. PDMs and ASMs build the shape models from a 2-D training set of image landmarks. In PDMs and



(a) (b)

Figure 2.1: Illustration of PA alignment, following (*a*) *reference-space* model, and (*b*) *data-space* model. Note that $\mathbf{A}_i = \mathbf{T}_i^{-1}$.

ASMs, first PA is used to remove rigid transformations and, then Principal Component Analysis (PCA) is applied to construct a subspace that models the variation of the normalized shapes [18].

Given a set $\mathcal{D} = \{\mathbf{D}_i \in \mathbb{R}^{d\times\ell}\}$ of n centered shapes composed by ℓ landmarks, PA [23, 36, 35, 44, 10] computes the d-dimensional reference shape $\mathbf{M} \in \mathbb{R}^{d\times\ell}$ and the set $\mathcal{T} = \{\mathbf{T}_i \in \mathbb{R}^{d\times d}\}$ of n transformations (e.g., affine, Euclidean) that minimize the *reference-space* cost [44, 35, 10] (see Fig. 2.1 (a)):

$$E_R(\mathbf{M}, \mathcal{T}) = \sum_{i=1}^{n} ||\mathbf{T}_i\mathbf{D}_i - \mathbf{M}||_F^2. \tag{2.1}$$

Note that in the case of two-dimensional shapes ($d = 2$), $\mathbf{D}_i = \begin{bmatrix} x_1 & x_2 & \dots & x_\ell \\ y_1 & y_2 & \dots & y_\ell \end{bmatrix}$. Alternatively, PA can be optimized using the *data-space* model [10] (see Fig. 2.1 (b)):

$$E_D(\mathbf{M}, \mathcal{A}) = \sum_{i=1}^{n} ||\mathbf{D}_i - \mathbf{A}_i\mathbf{M}||_F^2, \tag{2.2}$$

where $\mathbf{A}_i = \mathbf{T}_i^{-1} \in \mathbb{R}^{d\times d}$ is the inverse transformation of $\mathbf{T}_i$, and the set $\mathcal{A} = \{\mathbf{A}_i \in \mathbb{R}^{d\times d}\}$ defines n rigid transformations for the reference shape $\mathbf{M}$.

The error function in Eq. (2.1) for the reference-space model minimizes the difference between the reference shape and the registered shape data. In the data-space model, the error function in Eq. (2.2) compares the observed shape points with the transformed reference shape, i.e., shape points predicted by the model and based on the notion of average shape [110]. This

difference between the two models leads to different properties. Since the reference-space cost (E_R, Eq. (2.1)) is a sum of squares and it is convex in the optimization parameters, it can be optimized globally with Alternated Least Squares (ALS) methods. On the other hand, the data-space cost (E_D, Eq. (2.2)) is a bilinear problem and non-convex. If there is no missing data, the data-space model can be solved using the Singular Value Decomposition (SVD) [10]. However, for large datasets it is more efficient (in both space and time) to minimize Eq. (2.2) with ALS methods. A major advantage of the data-space model is that it is *gauge invariant* (i.e., the cost does not depend on the coordinate frame in which the reference shape and the transformations are expressed). Benefits of both models are combined in [10]. Recently, Pizarro et al. [80] have proposed a convex approach for PA based on the reference-space model. In their case, the cost function is expressed with a quaternion parametrization which allows conversion to a Sum of Squares Program (SOSP). Finally, the equivalent semi-definite program of a SOSP relaxation is solved using convex optimization.

However, previous work on PA suffers from several limitations when building 2-D models from 2-D shapes or images: (1) the 2-D training samples do not necessarily cover a uniform sampling of all 3-D rigid transformations of an object and this can result in a biased model (i.e., some poses are better represented than others); (2) a non-uniform sampling of the rotation space can lead to an unnecessary increase memory and computation requirements to train the models; (3) the models learned using only 2-D landmarks cannot model missing landmarks with large pose changes. Moreover, PA methods can lead to local minima problems if there are missing components in the training data; finally, (4) PA is computationally expensive, it scales linearly with the number of samples and landmarks and quadratically with the dimension of the data. On the other hand, having access to 3-D models of objects, we can overcome PA issues and build 2-D models by rotating and projecting 3-D datasets.

2.2 Projected Procrustes Analysis

Due to advances in 3-D capture systems, nowadays it is common to have access to 3-D shape models for a variety of objects. Given a set $\mathcal{D} = \{\mathbf{D}_i \in \mathbb{R}^{3\times\ell}\}$ of n centered 3-D shapes, we can compute a set $\mathcal{P} = \{\mathbf{P}_j \in \mathbb{R}^{2\times 3}\}$ of r projections, one for each of the shapes (after removing translation) and

minimize the Projected Procrustes Analysis, PPA:

$$E_{\text{PPA}}(\mathbf{M}, \mathcal{A}) = \sum_{i=1}^{n} \sum_{j=1}^{r} \|\mathbf{P}_j \mathbf{D}_i - \mathbf{A}_{ij}\mathbf{M}\|_F^2 , \tag{2.3}$$

where $\mathcal{A} = \{\mathbf{A}_{ij} \in \mathbb{R}^{2\times 2}\}$ defines the set of n transformations for the reference shape $\mathbf{M} \in \mathbb{R}^{2\times \ell}$. $\mathbf{P}_j = \mathbf{P}\mathbf{R}(\boldsymbol{\omega}_j)$ is an orthographic projection onto the X-Y plane:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega}_j)$ in a given domain $\boldsymbol{\Omega}$, defined by the rotation angles $\boldsymbol{\omega}_j = \{\phi_j, \theta_j, \psi_j\} \in \mathbb{R}^3$. Note that, while data and reference shapes are d-dimensional in Eq. (2.1) and Eq. (2.2), data $\mathbf{D}_i$ and reference $\mathbf{M}$ shapes in Eq. (2.3) are fixed to be 3-D and 2-D, respectively. Hence, $\mathbf{A}_{ij}$ is a 2-D transformation mapping $\mathbf{M}$ to the j^{th} 2-D projection of the i^{th} 3-D shape. Alternate Least Squares (ALS) is a common method to optimize Eq. (2.2) and (2.3). When $\mathbf{A}_{ij}$ is an affine transformation, ALS alternates between minimizing over $\mathbf{M}$ and $\mathbf{A}_{ij}$ by using the following expressions:

$$\mathbf{A}_{ij} = \mathbf{P}_j \mathbf{D}_i \mathbf{M}^T \left(\mathbf{M}\mathbf{M}^T\right)^{-1} \qquad \forall i, j, \tag{2.4}$$

$$\mathbf{M} = \left(\sum_{i=1}^{n}\sum_{j=1}^{r} \mathbf{A}_{ij}^T \mathbf{A}_{ij}\right)^{-1} \left(\sum_{i=1}^{n}\left(\sum_{j=1}^{r} \mathbf{A}_{ij}^T \mathbf{P}_j\right)\mathbf{D}_i\right). \tag{2.5}$$

Note that PPA and its extensions deal with missing data naturally. Since they use the whole 3-D shape of objects, the enhanced 2-D dataset resulting of projecting the data from different viewpoints can be constructed without occluded landmarks. It is important to notice that building 2-D models from 3-D samples is a problem that has been relatively unexplored in computer vision [31].

PPA formalizes how to build multi-view 2-D models by projecting 3-D data samples after applying 3-D rotations to them. However, PPA does not guarantees unbiased 2-D models by itself, since it depends on how rotations $\mathbf{R}(\boldsymbol{\omega}_j)$ are chosen. Uniformly distributed rotations will generate unbiased models, while non-uniform rotations will lead to models representing some viewpoints better than others. Different parametrizations are detailed in the following section, as well as the methodologies to generate uniform rotations with them.

2.3 Uniform Distribution of Rotations

The Special Orthogonal group in 3-D, $SO(3)$, forms a group whose action is the composition of rotations. Each rotation is a linear transformation that preserves the length and spatial orientation of vectors.

How to generate uniform distributions over $SO(3)$ is a widely studied topic because of its known benefits in heterogeneous applications: computer graphics [7, 51], computer vision [77], reconstruction of biological complexes [81], and path planning [53, 109, 108], among others. Methods generating uniform rotations in $SO(3)$ are divided into random and deterministic approaches. Random approaches determine sets of rotations generated without a pattern design, and deterministic methods result in rotation sequences defined by a certain generative function. On the one hand, rotation distributions from deterministic uniform methods lead to resolution completeness. However, since the number of sample rotations must be known in advance, uniformity measures are not optimal for all samples of the sequence. On the other hand, if resolution completeness is not required, random uniform methods are the simplest way to obtain uniform distributions independently of the size of the samples set.

2.3.1 Parametrization of Rotations

There are several parameterizations for 3-D rotations around the origin, with Euler angles and quaternions being the most common ones.

Euler angles. Euler angles encode orientations in the three dimensional Euclidean space $\mathbb{R}^3$ through the composition of three rotations (ϕ, θ, ψ), each one around a single axis of a basis. The final rotation is obtained multiplying three rotation matrices, $\mathbf{R}(\boldsymbol{\omega}) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$:

$$\mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix},$$
$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix},$$
$$\mathbf{R}_z(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where Euler angles are uniformly distributed ($\mathcal{U}$) in the domains $\phi, \psi = \mathcal{U}(-\pi, \pi]$ and $\theta = \mathcal{U}[-\frac{\pi}{2}, \frac{\pi}{2}]$. As we show in the first column of Fig. 2.2 (a), random uniform sampling of Euler angles do not produce a uniform distribution of rotations. In addition they suffer from singularities like the Gimbal lock problem.

Gimbal Lock takes place on rotations in a 3-D space when two of the three axes are parallel. One degree of freedom is lost and, therefore, only rotations in 2-D space can be performed. An easy example to understand this issue appears when using the Z-Y-Z convention, i.e., first, a rotation on the Z-axis by ϕ angle, followed by a turn on the rotated Y-axis of θ angle and, finally, a rotation by ψ angle on the new Z-axis. If $\theta = 0$, it produces a rotation by $\delta_1 = \phi + \psi$ angle, only on Z-axis. In this case, the system loses a degree of freedom and it is "locked" rotating in a degenerate 2-D space. Of course, the same situation occurs when $\theta = \pi$, with a final rotation of $\delta_2 = \phi - \psi$ angle around Z-axis. It is a clear example of singularity on Euler angles, where different rotations in the Euler representation are mapped onto a single rotation in $SO(3)$. In the previous example, the final rotation described by $\theta = 0$ and $\delta_1 = \phi + \psi$ could be achieved by any different combination of ϕ and ψ.

Quaternions. In order to solve Euler angles issues, quaternions are generally used as a valid alternative. Quaternions, $\mathbf{q} = [a, bi, cj, dk]$, were conceived by Hamilton [38] by extending complex numbers. Each unit quaternion, $||\mathbf{q}|| = 1$, represents a 4-D point in the unit 3-sphere, S^3, being a rotation in $SO(3)$. For any unit quaternion, $\mathbf{q} = [\cos(\alpha/2), \sin(\alpha/2)\hat{\mathbf{u}}]$, and for any vector $\mathbf{v} \in \mathbb{R}^3$, the action of the triple product, $\mathbf{v}' = \mathbf{q}\mathbf{v}\mathbf{q}^* = \mathbf{R}_q\mathbf{v}$,

$$\mathbf{R}_q = \begin{pmatrix} 1-2(c^2+d^2) & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & 1-2(b^2+d^2) & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & 1-2(b^2+c^2) \end{pmatrix},$$

may be geometrically interpreted as a rotation of the vector $\mathbf{v}$ through an angle α, being $\hat{\mathbf{u}}$ the axis of rotation. However, not all quaternion parameterizations perform uniform rotations.

The chosen method to show this fact is the random sampling of the four quaternion components following uniform distributions $\mathbf{q} = [a, b, c, d]$, where $a, b, c, d \in \mathcal{U}(-1, +1)$, and normalizing the resulting random quaternion as $\mathbf{q} = \mathbf{q}/||\mathbf{q}||$. As we show in the first column of Fig. 2.2 (b), this parameterization leads to non-uniform rotations.

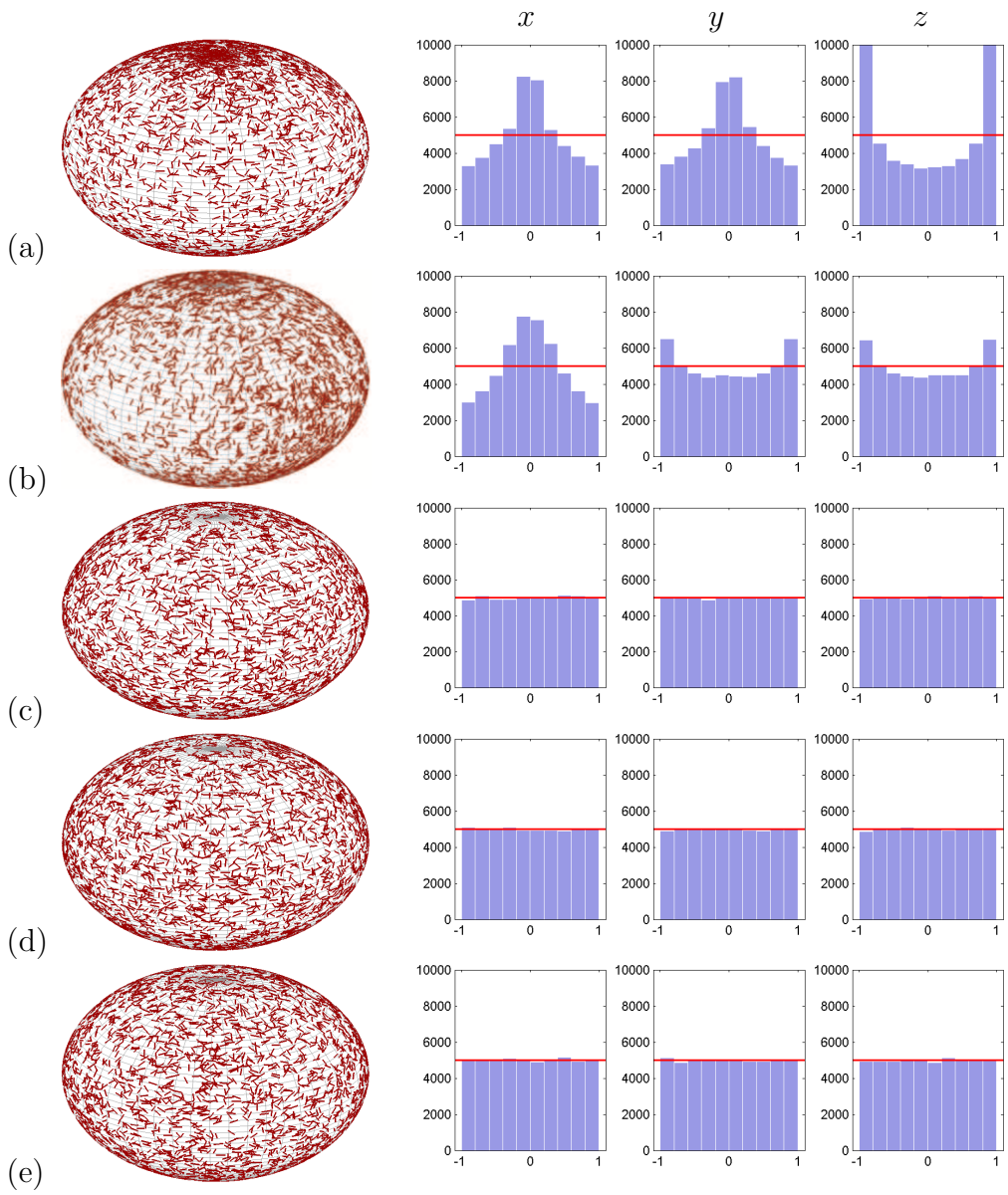


Figure 2.2: *(left)* Qualitative results and *(right)* distributions of x, y, z shape components onto the unit 2-sphere, rotating 3-D shapes around the origin using (a) *Euler angles*; (b) *Random sampling of the 4 quaternion components*; (c) *Euler trick* [53]; (d) *Gaussian distribution of quaternion components* [29]; and (e) *Subgroup algorithm* [89].

2.3.2 Random Uniform Distributions on $SO(3)$

Random uniform methods are those approaches that produce rotations equally distributed on $SO(3)$. In this section we introduce three random uniform

methods: *Euler angles trick*, *Gaussian distribution of quaternion components*, and *Subgroup algorithm*. To illustrate their distributions we followed the criteria proposed by Shoemake [89]: the three coordinates x, y, z of a vector uniformly distributed on a sphere are also uniformly distributed between their limits. In the first column of Fig. 2.2 we reported qualitative results of non-uniform (rows (a), (b)) and uniform (rows (c), (d), (e)) distributions. 3-D shapes composed by two centered vectors are rotated on the unit 2-sphere, and the line connecting both vectors is showed onto the sphere surface. The remaining columns of Fig. 2.2 show quantitative results of uniformity. From second to forth row, distributions of the three coordinates x, y, z of the shape are presented. Observe that coordinates of non-uniform methods present non-uniform distributions. However, they result in uniform distributions when rotations are performed with widely known uniform rotation approaches.

Euler angles trick. It is widely known that uniformly sampled Euler angles produce non-uniform rotations. However, it is possible to compensate such non-uniformity depending on the Euler angles convention. Uniformly randomized orientations using X-Y-Z convention, for example, can be achieved with $\phi, \psi = \mathcal{U}(-\pi, \pi]$, $z = \mathcal{U}(-1, +1)$, and $\theta = \sin^{-1}(z)$; or when the Z-Y-Z convention is used, uniformly distributed orientations can be achieved with $\phi, \psi = \mathcal{U}(-\pi, \pi]$, $z = \mathcal{U}(-1, +1)$, and $\theta = cos^{-1}(z)$. Nevertheless, these specific distributions rely on the convention used and there exist, at least, 24 conventions for Euler angles. On the other hand, a simple algorithm is presented by Kuffner [53] which generates uniform random distributions of rotations for any convention of Euler angles (Algorithm 1), as we show in the first column of Fig. 2.2 (c).

Gaussian distribution of quaternion components. In order to generate uniformly distributed unit random quaternions, spherical symmetry of the multidimensional Gaussian density function is exploited in [29]. For each pair of quaternion components, a zero-mean Gaussian distribution with a common variance is generated from a pair of random values. This operation is performed by generating 4 random variables, $X_i \in \mathcal{U}(0, 1), i = 1, \ldots, 4$, and drifting them to a pair of Gaussian distributions using the Box-Muller method [14]:

$$r_1 = -2\log(X_1), \alpha_1 = 2\pi X_2, r_2 = -2\log(X_3), \alpha_2 = 2\pi X_4.$$

Hence, $\mathbf{q} = [a, b, c, d]$ is defined as:

Algorithm 1: *Euler trick* algorithm [53], which generates random uniform distributions of rotations for any convention of Euler angles.

Input: 4 random numbers: $X_1, X_2, X_3, X_4 \in \mathcal{U}(0,1)$
Output: 3 uniform random Euler angles: ϕ, ψ, θ
$\phi = 2\pi X_1 - \pi$;
$\theta = \arccos(1 - 2X_2) + \pi/2$;
if $X_3 < 1/2$ **then**
  if $\theta < \pi$ **then**
    $\theta = \theta + \pi$;
  else
    $\theta = \theta - \pi$;
  end
end
$\psi = 2\pi X_3 - \pi$;
return ϕ, ψ, θ

$$
\begin{aligned}
a &= \sqrt{r_1}\cos(\alpha_1),\\
b &= \sqrt{r_1}\sin(\alpha_1),\\
c &= \sqrt{r_2}\cos(\alpha_2),\\
d &= \sqrt{r_2}\sin(\alpha_2),
\end{aligned}
$$

such as the normalized quaternion $\mathbf{q} = \mathbf{q}/||\mathbf{q}||$ produces uniform rotations, as we show in the first column of Fig. 2.2 (d).

Subgroup algorithm. The most widely known method to generate uniform random rotations using quaternions is the one presented by Shoemake [89], where random unit quaternions are generated (see the first column of Fig. 2.2 (e)) from three random variables through the subgroup algorithm.

Since all the 3-D rotations in this space compose a group, a subgroup $q = [w, 0, 0, s]$ is formed by planar rotations α around the Z-axis, being $w = \cos(\alpha/2)$ and $s = \sin(\alpha/2)$. Consequently, cosets[1] of this subgroup, $q = [a, b, c, 0]$, are rotations pointing Z-axis in different directions. Following the subgroup algorithm [22], a uniformly distributed element of the complete group can be achieved by the multiplication of a uniformly distributed

[1]Recall that a *coset* is either the left or right coset of some subgroup $\mathcal{H}$ in a group $\mathcal{G}$. If $g \in \mathcal{G}$, then the *left coset* of $\mathcal{H}$ in $\mathcal{G}$ with respect to g consists of all the products obtained by multiplying g by each of the elements of the subgroup $\mathcal{H}$, $g\mathcal{H} = \{gh : h \in \mathcal{H}\}$. Similarly, the *right coset* of $\mathcal{H}$ in $\mathcal{G}$ with respect to g consists of $\mathcal{H}g = \{hg : h \in \mathcal{H}\}$.

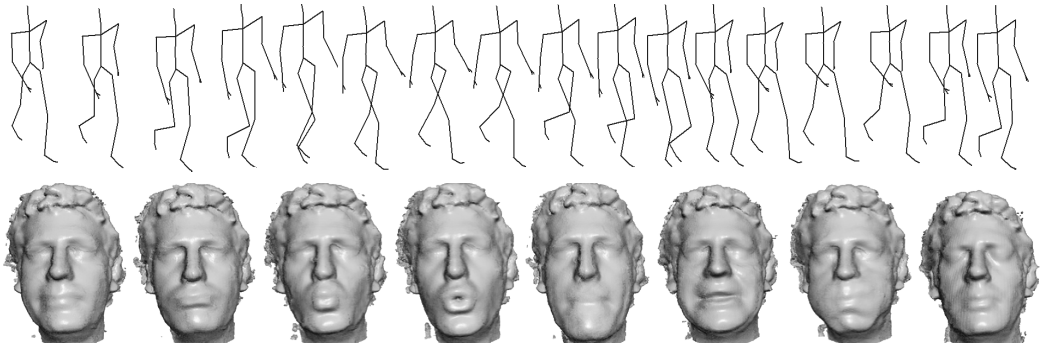


Figure 2.3: (*top*) Samples of skeletons from a person walking sequence extracted from the CMU MoCap dataset [1]. (*bottom*) Samples of faces with different expressions from the FaceWarehouse dataset [16].

element from the subgroup with a uniformly distributed coset:

$$\mathcal{U}([w,0,0,s])\mathcal{U}([a,b,c,0]) = \mathcal{U}([wa, wb+sc, -sb+wc, sa]).$$

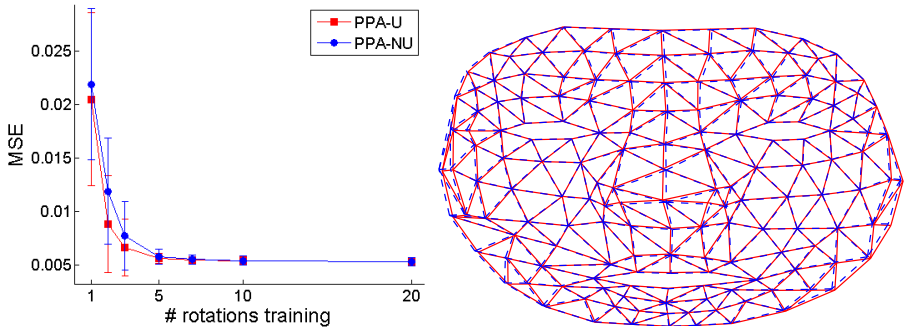
Given three independent random variables $X_i \in \mathcal{U}(0,1), i = 1,\ldots,3$, we can compute $\mathbf{q} = [\cos(\alpha_2)r_2, \sin(\alpha_1)r_1, \cos(\alpha_1)r_1, \sin(\alpha_2)r_2]$ random unit quaternions where,

$$\alpha_1 = 2\pi X_2, \alpha_2 = 2\pi X_3, r_1 = \sqrt{(}1 - X_1), r_2 = \sqrt{(}X_1).$$

2.4 Experiments

This section illustrates the benefits of PPA to build multi-view 2-D shape models of faces and human bodies, in terms of their generalization to different viewpoints within a given domain. We also show the benefits of building 2-D models by means of a uniform distribution of rotations instead of a non-uniform coverage of the rotation space. First, we compare the performance of PPA, trained with uniformly and non-uniformly rotated training sets, to build a 2-D shape model of faces from the FaceWarehouse [16] dataset. Next, we learn Motion Capture (MoCap) human bodies using the Carnegie Mellon University (CMU) MoCap dataset [1], again comparing performance between uniform and non uniform rotations. See Fig. 2.3 for examples of face and body samples from both datasets.

We rotated the 3-D face and body samples in yaw, pitch and roll angles, within the ranges of $\phi, \psi \in [-\pi, \pi]$ and $\theta \in [-\pi/2, \pi/2]$. Two PPA models were trained rotating and projecting the same 3-D data for each experiment: PPA-U rotated the 3-D shapes in a uniform way by means of



(a) (b)

Figure 2.4: Comparison of *PPA-U* and *PPA-NU* (*a*) reconstruction errors, and (*b*) mean shapes, on FaceWarehouse dataset.

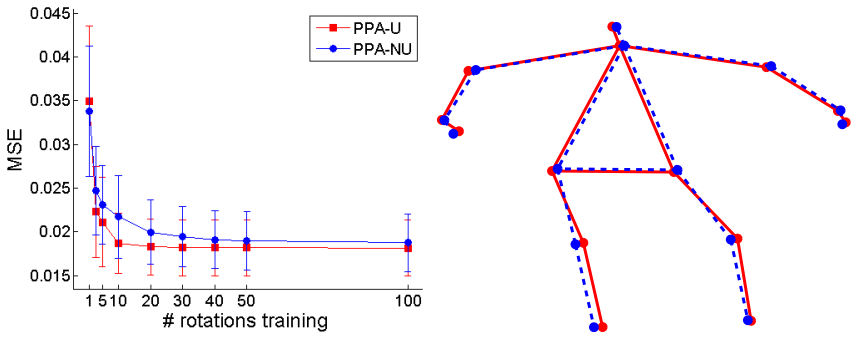
the subgroup algorithm, and PPA-NU rotated the 3-D samples following the common parametrization of Euler angles, that does not cover uniformly the rotation space. We report results for 300 uniform rotations for testing, while increasing the number of rotations in training.

2.4.1 Learning 2-D Face Models

The aim of this experiment is to build a generic 2-D face model that can reconstruct a large range of 3-D rotations. For training and testing, we used the FaceWarehouse dataset that is composed of 150 subjects, each one with 20 different facial expressions (see Fig. 2.3). For all the subjects, dense point meshes are available, as well as RGB data generated from RGB-D scans. The original model has 11510 points, and we sub-sampled the mesh to 162 landmarks. We report the Mean Squared Error (MSE) relative to the intra-eye size for 100 realizations, and we plot the MSE and the half of the standard deviation.

For training we randomly selected 20 subjects, with three expressions per subject. For testing we randomly selected 10 different subjects with the same three expressions as training. We report results varying the number of training rotations between $1 \sim 20$.

Fig. 2.4 compares PPA-U and PPA-NU. Fig. 2.4 (a) shows the mean reconstruction error. From the figure, one can observe that test errors for both PPA-U and PPA-NU decrease with the number of rotations in the training set, and they converge when the addition of more rotated faces do not provide supplementary information. PPA-U model trained with uniformly distributed rotations, generalizes slightly better to different viewpoints with



(a) (b)

Figure 2.5: Comparison of *PPA-U* and *PPA-NU* (*a*) reconstruction errors, and (*b*) mean shapes, on CMU MoCap dataset.

small number of rotations. Note that 5 rotations means to enhance the training set by 5 times. Fig. 2.4 (b) shows the mean face computed with both methods from a training set of 20 rotations.

2.4.2 Learning 2-D Human Body Models

The aim of this experiment is to build a generic 2-D skeleton model from 3-D Mottion Capture (MoCap), able to reconstruct large range of 3-D rigid transformations (i.e. viewpoints). For training and testing, we used the Carnegie Mellon University MoCap dataset that is composed of 2605 sequences performed by 109 subjects. The sequences cover a wide variety of daily human activities and sports. Skeletons with 31 joints are provided (see Fig. 2.3), as well as RGB video recordings for several sequences. We trained our models using the set of 14 landmarks as is common across several databases for human pose estimation. We report the MSE relative to the torso size for 100 realizations, and we plot the MSE and the half of the standard deviation.

For training we randomly selected 3 sequences with 30 frames per sequence from the set of 11 running sequences of the user number 9. For testing we randomly selected 2 sequences with 30 frames from the same set. We report results varying the number of training rotations between $1 \sim 100$.

Fig. 2.5 compares PPA-U and PPA-NU. Fig. 2.5 (a) shows the mean reconstruction error. From the figure, one can observe that errors in the test for both PPA-U and PPA-NU decrease with the number of rotations in the training. As expected, PPA-U model, trained with uniformly distributed rotations, generalizes better to different viewpoints, and PPA-NU converges to PPA-U error with a sufficient number of additional rotations. Note that the error achieved by PPA-U with 30 rotations is not achieved by PPA-

NU even with 100 rotations in the training set. Therefore, in this case, methods trained with a non-uniform sampling of the rotation space will need, at least, 70 times more space and computational time than methods trained with uniform rotations. Fig. 2.5 (b) shows the mean skeleton computed with both methods with a training set of 100 rotations. Observe that PPA-NU did not converge yet to the PPA-U after enhancing the dataset with 100 rotations. Also note that Fig. 2.5 (a) before convergence illustrates the reconstructed error for biased datasets to some specific viewpoints. PPA provides mechanisms to overcome PA limitations and build unbiased 2-D models.

2.5 Conclusions

Procrustes Analysis (PA) has been extensively used to align shapes and build rigid models despite suffering from significant issues. PA scales linearly with the number of samples and landmarks and quadratically with the dimension of the data, which can be computationally expensive when enhancing a dataset with different viewpoints of objects. In addition, 2-D training samples do not necessarily cover a uniform sampling of all 3-D rigid transformations of the training objects. This can result, not only in biased models, but also in an unnecessary increase of memory and computation requirements. Finally, the models learned using only 2-D landmarks cannot model missing landmarks with large pose changes, as well they can lead to local minima problems if there are missing components in the training data.

To address these issues, in this chapter we proposed Projected Procrustes Analysis. PPA formalizes the learning of multi-view 2-D rigid models by means of rotating and projecting 3-D data samples. When the rotation space is uniformly sampled, PPA builds unbiased 2-D models in terms of generalization to different viewpoints. In this chapter we also presented the most common parametrizations of rotations (i.e. Euler angles and quaternions) and different methodologies to produce random uniform rotations with them.

In the experimental section we showed that unbiased 2-D models are able to generalize better to different viewpoints with smaller number of rotations. We compared PPA with a training of uniform rotations (PPA-U) against the same method with a non-uniform sampling of the rotation space (PPA-NU), increasing the number of rotations in the training set. PPA-U needed less rotations for convergence than PPA-NU for both faces and skeletons datasets. Although large deformations of the CMU MoCap dataset make more evident the improvements of the uniform sampling in skeletons than

in faces experiments, we encourage the use of uniform sampling of rotation space in any dataset.

Therefore, we provide an intuitive PA extension to build unbiased 2-D rigid models able to generalize to different viewpoints. As a PA extension, PPA preserves all advantages of PA. However, due to PA limitations, PPA computation is still costly in memory and time. It requires to enhance the dataset with several rotations for each training shape. Note that we will overcome this issue in following chapters. PPA provides the basis of formulation and the understanding of the problem needed to develop the extensions presented in the remaining of this dissertation.

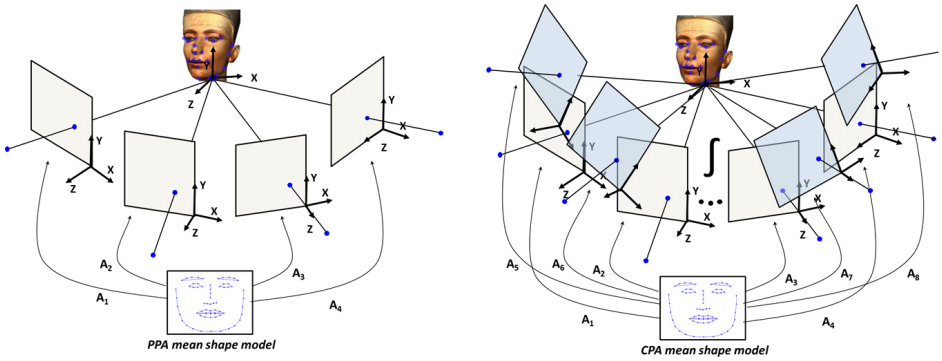
Chapter 3

Continuous Procrustes Analysis

Procrustes Analysis has been widely employed despite suffering from several limitations: (1) the 2-D training samples do not necessarily cover a uniform sampling of all 3-D rigid transformations of an object and this can result in a biased model (i.e., some poses are better represented than others); (2) it is computationally expensive to learn a shape model by sampling all possible 3-D rigid transformations of an object (see Fig. 3.1 (a)); (3) the models that are learned using only 2-D landmarks cannot model missing landmarks with large pose changes. Moreover, PA methods can lead to local minima problems if there are missing components in the training data; (4) finally, PA is computationally expensive, it scales linearly with the number of samples and landmarks and quadratically with the dimension of the data.

Projected Procrustes Analysis deals with most of these issues by building a multi-view 2-D model from 3-D samples. PPA enhances the training set with uniformly distributed viewpoints (i.e. rotations) of the 3-D training data. However, the number of training samples increases by r (number of rotations) times. Even small values of r will lead to a substantial increase of the number of shapes, and the complexity of PPA in space and time.

In order to deal with these drawbacks, in this chapter we propose Continuous Procrustes Analysis (CPA), by formulating PPA within a functional analysis framework (see Fig. 3.1 (b)). In this chapter we first introduce our approach in the context of Functional Data Analysis (FDA) and provide the basic mathematical background. Next, we detail the CPA formulation and optimization. Finally, we compare CPA against PPA and the state-of-the-art of PA in human samples datasets of faces and human bodies.



(a) (b)

Figure 3.1: Illustration of 2-D model building by means of projecting 3-D data from (a) discrete (PPA); and (b) continuous (CPA) approaches.

3.1 Functional Data Analysis

Our work is related to previous work on FDA [84]. FDA is a branch of statistics that analyzes data samples consisting of functions or surfaces, where each function is viewed as one sample element. The functions contained in the sample are typically considered to be independent and smooth. FDA methods are usually extensions of classical multivariate methods such as PCA [84], Linear Discriminant Analysis (LDA) [45] or ANalysis Of VAriance (ANOVA) [32].

In the context of human samples modeling, there are only a few works in computer vision that make use of FDA [70, 27, 71, 55]. Ormeneit et al. [71] proposed an automatic method for segmentation and modeling of cyclic motion sequences. They represented the body pose as a time series of joint angles, and applied a functional PCA based on a Singular Value Decomposition (SVD) operating in the Fourier domain. An advantage of this method is that it automatically deals with noise and missing data. They learned 3-D models of humans walking from Motion Capture data, that later used in a Bayesian tracking framework. Closer to our approach, Levin and Shashua [55] applied a continuous formulation of the PCA to model faces under different illuminations. Instead of modeling the raw sample data with the standard PCA, their method integrates over the convex hull of the data, and achieves unbiased estimates of the principal components of the images. Following these ideas, we parametrize any rotation of a 3-D object (e.g. faces, bodies) as a combination of three rotation matrices, and then we integrate on the rotation domain, in order to overcome the principal limitation of PPA: the need of explicitly rotate the 3-D data.

Fig. 3.1 (a) illustrates the PPA process of building 2-D models from a set of viewpoints of 3-D objects under different configurations. Continuous Procrustes Analysis (CPA) generalizes PPA by using an integral formulation (see Fig. 3.1 (b)) that avoids the need to generate 2-D projections from 3-D objects, and uniformly covers the space of 3-D transformations. In the continuous approach, rotation matrices are not parameters but functions of the rotation angles. Instead of averaging the 2-D projections of the 3-D objects, CPA integrates among the rotation angles, being extremely efficient in space and time. Before introducing the CPA formulation and optimization, we will review mathematical background on calculus and integrations on the rotation space.

3.2 Mathematical Background

This section describes the basic mathematical background for the CPA understanding. We review basic statements from the calculus of variations and integral calculus, as well as details regarding to the integration into the Special Orthogonal group in 3-D, $SO(3)$, and measures defined on it.

3.2.1 Calculus

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth scalar function. If $\mathbf{x}^* \in \mathbb{R}^n$ is a solution of the problem:

$$f(\mathbf{x}^*) = \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{3.1}$$

then the following equation is satisfied:

$$\nabla_{\mathbf{x}} f(\mathbf{x}^*) = 0, \tag{3.2}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator of the function $f(\mathbf{x})$ with respect to $\mathbf{x}$.

Now let $\mathbf{\Omega} \subset \mathbb{R}^n$ be an open and bounded subset, let $F : \mathbb{R}^d \to \mathbb{R}$ be a mapping, and we want to find a solution, $\mathbf{v}^* : \mathbf{\Omega} \to \mathbb{R}^d$, to the following functional problem:

$$\int_{\mathbf{\Omega}} F(\mathbf{v}^*(\mathbf{x}))\, d\mathbf{x} = \min_{\mathbf{v}} \left\{ \int_{\mathbf{\Omega}} F(\mathbf{v}(\mathbf{x}))\, d\mathbf{x} \right\}, \tag{3.3}$$

where the minimum is taken among all the functions $\mathbf{v} : \mathbf{\Omega} \to \mathbb{R}^d$ belonging to $L^p(\mathbf{\Omega}; \mathbb{R}^d) := \left\{ \mathbf{v} : \mathbf{\Omega} \to \mathbb{R}^d : \mathbf{v} \text{ is measurable and } \int_{\mathbf{\Omega}} |\mathbf{v}(x)|^p dx < \infty \right\}$. Then, it can be shown that the function $\mathbf{v}^* \in L^p(\mathbf{\Omega}; \mathbb{R}^d)$ satisfies:

$$\nabla_{\mathbf{v}} F(\mathbf{v}^*(\mathbf{x})) = 0, \qquad \forall \mathbf{x} \in \mathbf{\Omega} \setminus \mathbf{\Gamma}, \tag{3.4}$$

with $\int_{\mathbf{\Gamma}} 1\, d\mathbf{x} = 0$. i.e., $\mathbf{\Gamma}$ is a null set. The interested reader is referred to [30] and [19] for a more detailed review of the calculus of variations.

In order to manipulate multiple integrals, Fubini's Theorem determines conditions under which it is possible to compute a multiple integral using iterated integrals [92]. Let $\mathbf{\Omega}_p \subset \mathbb{R}^p$ and $\mathbf{\Omega}_q \subset \mathbb{R}^q$ be complete measure spaces. Let $f(x, y) \in L(\mathbf{\Omega}_p \times \mathbf{\Omega}_q; \mathbb{R}^d)$, i.e.:

$$\int_{\mathbf{\Omega}_p \times \mathbf{\Omega}_q} |f(x, y)| d(x, y) < \infty, \tag{3.5}$$

with $f(x, y)$ measurable, then:

$$\int_{\mathbf{\Omega}_p \times \mathbf{\Omega}_q} f(x, y) d(x, y) = \int_{\mathbf{\Omega}_p} \left(\int_{\mathbf{\Omega}_q} f(x, y) dy \right) dx = \int_{\mathbf{\Omega}_q} \left(\int_{\mathbf{\Omega}_p} f(x, y) dx \right) dy.$$

The integral with respect to a product of two measures can be calculated as iterated integrals with respect to those two measures.

3.2.2 Integration Over the $SO(3)$ Group

The Special Orthogonal group in 3-D, $SO(3)$, forms a group whose action is the composition of rotations. Each rotation is a linear transformation that preserves the length and spatial orientation of vectors.

Although different parametrizations of rotations exist, not all of them lead to uniform distribution of rotations (see Section 2.3). In te continuous domain, uniformity depends on finding a proper measure for $SO(3)$, since integration of functions on a particular space involves the definition of a specific measure on that space [92]. Hence, Euler angles parametrization of 3-D rotations can be adopted in the CPA formulation when defining a proper Haar measure, while it is unclear how to do it for quaternions.

The Haar measure is defined such that it assigns an "invariant volume" to subsets of locally compact topological groups and subsequently defines an integral for functions on those groups [68]. We may associate to any Haar measure μ on a group a bounded linear functional $F \in L(\mathbb{R}^p; \mathbb{R})$:

$$F(f) = \int_G f(\omega) d\mu(\omega).$$

As an example, the Haar measure on the group of rotations $SO(3)$ [68] (Section 7 of Chapter 1) leads to:

$$\int_{SO(3)} f(\omega) d\mu(\omega) = \int_0^{\pi} d\psi \int_0^{2\pi} d\theta \int_0^{2\pi} d\phi \frac{1}{8\pi^2} \sin(\theta) f(\omega(\phi, \theta, \psi)). \tag{3.6}$$

Using the Haar measure, we obtain an invariant integral for functions on the rotation group. Thus, the problem of discrete non-uniform distribution using Euler angles discussed in Chapter 2 is avoided with CPA in the definition of the integral.

3.3 Continuous Procrustes Analysis

A major limitation of PPA is the difficulty to generate uniform distributions in the Special Orthogonal group $SO(3)$ [53]. Due to the topology of $SO(3)$, in the discrete domain different angles should be sampled following different distributions, which becomes difficult when the rotation matrices must be confined in a specific region $\boldsymbol{\Omega}$ of $SO(3)$, restricted by rotation angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. Moreover, the computational complexity of PPA increases linearly with the number of rotations (r) and 3-D objects (n), and the number of samples needed will increase with r because several view-point projections are necessary to cover the rotation domain. In this section, we formulate Continuous Procrustes Analysis (CPA). CPA extends PPA by adopting a continuous formulation that incorporates the information of all rigid 3-D transformations.

We formulate the problem of CPA as an energy functional minimization, involving 3-D landmarks of objects and continuous 3-D rotations. We compute the reference shape following the data-space model (Fig. 2.1 (b)) because it is gauge invariant and its derivation is simpler than using the reference-space model. Our main assumption is that the best reference shape is the one that can approximate all possible 3-D shape configurations of a given set of shapes. We interpret this in the following way: we consider a set of 3-D shapes, we perform a predefined set of rotations, and we project them onto the 2-D space. Then, we estimate the reference shape by aligning it with each shape configuration using an estimated affine transformation.

Given a set $\mathcal{D} = \{\mathbf{D}_i \in \mathbb{R}^{3\times\ell}\}$ of n centered 3-D shapes, we integrate along all possible rotations $\mathbf{R}(\boldsymbol{\omega})$ in a given domain $\boldsymbol{\Omega} = \{\boldsymbol{\omega} = (\phi, \theta, \psi) \in \mathbb{R}^3\}$, projected to the X-Y plane, $\mathbf{P}(\boldsymbol{\omega}) = \mathbf{P}\mathbf{R}(\boldsymbol{\omega})$, by an orthographic projection $\mathbf{P}$. CPA finds the mean shape $\mathbf{M} \in \mathbb{R}^{2\times\ell}$ that best reconstructs the different projections $\mathbf{P}(\boldsymbol{\omega})$ of the 3-D data sample $\mathbf{D}_i$, up to an affinity transformation $\mathbf{A}(\boldsymbol{\omega})_i \in \mathbb{R}^{2\times 2}$. CPA minimizes the functional:

$$E_{\text{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \|\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}(\boldsymbol{\omega})_i\mathbf{M}\|_F^2 \, d\boldsymbol{\omega}, \tag{3.7}$$

where $\boldsymbol{\Omega}$ is the set of 3-D rotation domains, $\boldsymbol{\omega}$ are the Euler angles, and the Haar measure $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$ ensures uniformity in $SO(3)$ [68].

Note that for Euler angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$, the Haar measure can be computed for every domain $\boldsymbol{\Omega}$. For instance, for a complete sphere this measure corresponds to $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\beta)d\alpha d\beta d\gamma$. Therefore, CPA finds the optimal 2-D reference shape of a 3-D dataset, rotated and projected in a given domain $\boldsymbol{\Omega}$, by integrating over all possible rotations in that domain. Notice that building models by integrating on the rotation space has been a relatively unexplored problem in computer vision [43].

The main difference between Eq. (2.3) and Eq. (3.7) is that the entries in $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$ and $\mathbf{A}(\boldsymbol{\omega})_i \in \mathbb{R}^{2\times 2}$ are not scalars anymore, but functions of the integration angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. In both cases 2-D shape projections depend directly on the 3-D structure of the object $\mathbf{D}_i$ and the 3-D transformation parameters, but Eq. (3.7) is a continuous formulation, and discrete sums are extended by integrals.

After some linear algebra and functional analysis (see Appendix A for derivation and optimization details), it is possible to find an equivalent expression to the discrete approach (Eq. (2.3)), where $\mathbf{A}(\boldsymbol{\omega})_i$ and $\mathbf{M}$ have the following expressions:

$$\mathbf{A}(\boldsymbol{\omega})_i = \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1} \qquad \forall i, \tag{3.8}$$

$$\mathbf{M} = \left(\sum_{i=1}^{n}\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{A}(\boldsymbol{\omega})_i d\boldsymbol{\omega}\right)^{-1}\left(\sum_{i=1}^{n}\left(\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)\mathbf{D}_i\right). \tag{3.9}$$

It is important to notice that the 2-D projections are not explicitly computed in the continuous formulation. The solution of $\mathbf{M}$ is found using fixed-point iteration in Eq. (3.7):

$$\mathbf{M} = (\mathbf{Z}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1})^{-1}\mathbf{Z}, \tag{3.10}$$

where $\mathbf{X} = \int_{\boldsymbol{\Omega}}\mathbf{P}(\boldsymbol{\omega})^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances and[1] $\mathbf{Z} = (\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M}\left(\sum_{i=1}^{n}(\mathbf{D}_i^T \otimes \mathbf{D}_i^T)\operatorname{vec}(\mathbf{X})\right)^{(\ell)}$. Note that the definite integral $\mathbf{X}$ is not data dependent, and it can be computed off-line. Therefore, CPA builds multi-view 2-D rigid models by means of integrating among all possible rotations in a given domain, in an efficient manner. CPA models are unbiased (i.e. different viewpoints are equally represented) because we use the Haar measure in the definition of the integral.

[1] See Section 1.4 for an explanation of the vec-transpose operator.

3.4 Experiments

This section illustrates the benefits of PPA and CPA, and compares them with state-of-the-art PA methods to build 2-D shape models of faces and human bodies. First, we compare the performance of PA and and our extensions to build a 2-D shape model of faces from FaceWarehouse [16] dataset (Experiment 1). Next, we compare our discrete and continuous approaches in a large scale experiment (Experiment 2). Afterwards, we learn Motion Capture (MoCap) skeletons using the Carnegie Mellon University (CMU) MoCap dataset [1], and we compare them with the state-of-the-art (Experiment 3) and in a large scale experiment (Experiment 4).

3.4.1 Learning 2-D Face Models

The aim of Experiments 1 and 2 is to build a generic 2-D face model that can reconstruct non-rigid facial deformations under a large range of 3-D rotations. For training and testing, we used the FaceWarehouse dataset (detailed in Section 2.4.1). We sub-sampled the original mesh to 49 and 162 landmarks, depending on the experiment. We rotated the 3-D faces in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$. The rotations were uniformly selected and we report results for 300 rotations for testing, while varying the number of rotations in training. We report the Mean Squared Error (MSE) relative to the intra-eye size for 100 realizations, and we plot the MSE and the half of the standard deviation.

Experiment 1: Comparison with State-of-the-Art PA Methods on Faces

This section compares PPA and CPA methods with the state-of-the-art Stratified Generalized Procrustes Analysis (SGPA)[2] [10]. For training we randomly selected 20 subjects, three expressions per subject and 49 landmarks (this is due to the memory limitations of SGPA). For testing we randomly selected 10 different subjects with the same three expressions as training. We report results varying the number of training rotations between $1 \sim 100$.

There exist several versions of SGPA. We selected the "Affine-factorization" with the data-space model to make a fair comparison with our method. Recall that under our assumption of non-missing data "Affine-All" and "Affine-factorization" achieve the global optimum, being "Affine-factorization" faster.

[2]The code was downloaded from author's website (http://isit.u-clermont1.fr/~ab).

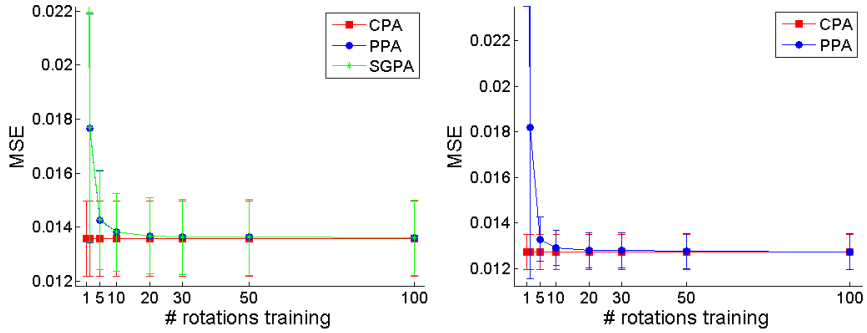


Figure 3.2: Comparisons of (*a*) *CPA*, *PPA*, and *SGPA* (Experiment 1) ; and (*a*) *CPA* and *PPA* (Experiment 2) as a function of the number of training viewpoint projections, on the FaceWarehouse dataset.

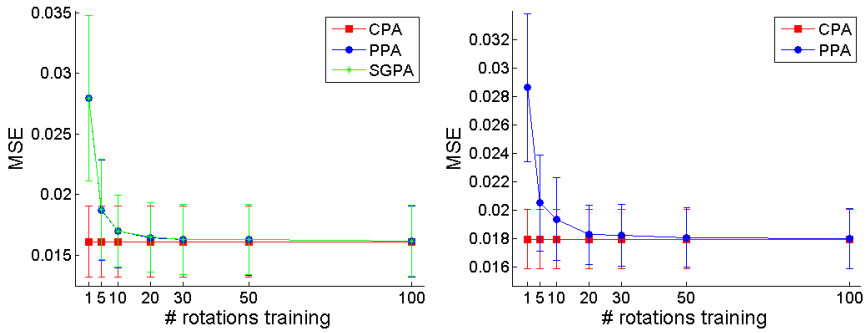
Fig. 3.2 (a) compares the mean reconstruction error in test for PPA, CPA, and SGPA. From the figure, one can observe that error in test for PPA and SGPA decreases with the number of rotations in training, and it converges to CPA, which provides a bound on the lower error. However, observe that CPA achieves the same performance, but it is much more efficient.

Note that we used 60 3-D faces (20 subjects and 3 expressions) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$, and PPA and SGPA needed about 20 rotations to achieve similar results to CPA. In this case, discrete methods need 20 times more space than the continuous one. Execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 1.11 sec. (PPA), 0.05 sec. (CPA) and 1.90 sec. sec. (SGPA).

Experiment 2: Comparison between CPA and PPA

This experiment compares PPA and CPA in a large-scale problem as a function of the number of rotations between $1 \sim 100$. For training we randomly selected 120 subjects, five expressions per subject and 162 landmarks. For testing we randomly selected 30 different subjects with the same five expressions as training.

Fig. 3.2 (b) shows the mean reconstruction error in test comparing PPA and CPA. As expected, PPA converges to CPA as the number of training rotations increases. Observe that CPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training faces (120 subjects and 5 expressions) and rotation domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 20 2-D viewpoint projections to achieve similar results to CPA. Thus, the discrete model PPA needs 20 times



(a) (b)

Figure 3.3: Comparisons of (*a*) *CPA*, *PPA*, and *SGPA* (Experiment 3); and (*b*) *CPA* and *PPA* (Experiment 3) on the CMU MoCap dataset, as a function of the number of training viewpoint projections.

more storage space than CPA. Execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 12.10 sec. (PPA) and 2.50 sec. (CPA).

3.4.2 Learning 2-D Human Body Models

The aim of Experiments 3 and 4 is to build a generic 2-D skeleton model from 3-D Motion Capture (MoCap). For training and testing, we used the Carnegie Mellon University MoCap dataset (detailed in Section 2.4.2). We trained our models using the set of 14 landmarks as is common across several databases for human pose estimation, and we rotated the shapes in the same way as the Experiments 1 and 2. We report the MSE relative to the torso size for 100 realizations, and we plot the MSE and the half of the standard deviation.

Experiment 3: Comparison with State-of-the-Art PA Methods

This section compares PPA and CPA methods with the state-of-the-art SGPA [10]. For training we randomly selected 3 sequences with 30 frames per sequence from the set of 11 running sequences of the user number 9. For testing we randomly selected 2 sequences with 30 frames from the same set. We report results varying the number of considered viewpoints in training between $1 \sim 100$ rotations.

Fig. 3.3 (a) compares the mean reconstruction error for PPA, CPA, and SGPA. From the figure, one can observe that error in test for PPA and SGPA decreases with the number of rotations in training, and it converges to CPA,

which provides a bound on the lower error. Observe, that we used 90 3-D bodies (3 sequences with 30 frames) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$, and PPA and SGPA needed about 30 rotations to achieve similar results to CPA. Therefore, in this case, discrete methods need 30 times more space than the continuous one. Execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 1.42 sec. (PPA), 0.02 sec. (CPA) and 3.30 sec. (SGPA).

Experiment 4: Comparison between PPA and CPA

This experiment compares PPA and CPA in a large-scale problem as a function of the number of rotations. For training we randomly selected 20 sequences with 30 frames per sequence. For testing we randomly selected 5 sequences with 30 frames. We report results varying the number of viewpoints in training between $1 \sim 100$ rotations.

Fig. 3.3 (b) shows the mean reconstruction error comparing PPA and CPA. As expected, PPA converges to CPA as the number of training rotations increases. However, observe that CPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training bodies (20 sequences with 30 frames) and rotation domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 30 2-D viewpoint projections to achieve similar results to CPA. Thus, the discrete model PPA needs 30 times more storage space than CPA. The execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 15.76 sec. (PPA) and 0.04 sec. (CPA).

3.5 Conclusions

In Chapter 2 we proposed PPA to deal with PA limitations to build 2-D models, such as: (1) the 2-D training samples do not necessarily cover a uniform sampling of all 3-D rigid transformations of an object, which can result in a biased model (i.e., some poses are better represented than others); (2) it is computationally expensive to learn a shape model by sampling all possible 3-D rigid transformations of an object; (3) the models learned using only 2-D landmarks cannot model missing landmarks with large pose changes. Moreover, PA methods can lead to local minima problems if there are missing components in the training data; finally, (4) PA is computationally expensive, it scales linearly with the number of samples and landmarks and quadratically with the dimension of the data. PPA solved most of PA drawbacks by building multi-view 2-D models from 3-D samples, after en-

hancing the training set with uniformly distributed viewpoints (i.e. rotations) of the 3-D training data. Although successful, this process incremented the computational requirements of PPA in space and time.

To address PA and PPA drawbacks in an efficient manner, in this chapter we proposed CPA. CPA extends PA and PPA within a functional analysis framework and builds multi-view 2-D rigid models in an efficient way, by means of integrating among all possible rotations in a given domain. CPA models are unbiased because we use the Haar measure in the definition of the integral.

In experimental section we compared CPA models against the state-of-the-art PA methods and our previous discrete multi-view approach (PPA). We compared all methods in faces and bodies datasets, raising the number of rotations in the training set. As the number of projections increased, discrete methods converged to CPA, which provided a lower bound for the error in all experiments. Moreover, CPA was much more efficient in space and time.

Therefore, CPA extends PA and PPA by integrating among all rotations in a given domain. It provides and efficient approach to build unbiased 2-D rigid models, able to generalize to different viewpoints, but being much more efficient in space and time. CPA generates unbiased models because it uniformly covers the space of projections. Experiments comparing 2-D CPA models of faces and bodies show improvements w.r.t. state-of-the-art PA methods. However, note that CPA is not able to capture non-rigid deformations in the dataset. We will overcome this limitation in the next chapter by extending the formulation to model non-rigid variations.

Chapter 4

Subspace Procrustes Analysis

In previous chapters we extended Procrustes Analysis (PA) to build multi-view 2-D rigid models from 3-D data samples. We modeled 3-D datasets across viewpoints with a 2-D mean shape and an affine transformation in a discrete (see Projected PA in Chapter 2) and in a continuous way (see Continuous PA in Chapter 3). In this chapter, we add a subspace that is able to model non-rigid deformations of the samples, as well as rigid 3-D transformations that the affine transformation cannot model by itself. We call this novel method Subspace Procrustes Analysis (SPA). We propose a discrete and continuous formulation in order to provide a better understanding of the problem, and experimentally show that they converge to the same solution when the number of sampled rotations (r) increases. As we will describe later, adding a continuous subspace to the CPA formulation is not a trivial task. For instance, modeling a subspace following the standard methodology based on CPA would still require to generate r rotations for each 3-D sample. Hence, the CPA efficiency is limited to rigid models while the method presented in this chapter is not.

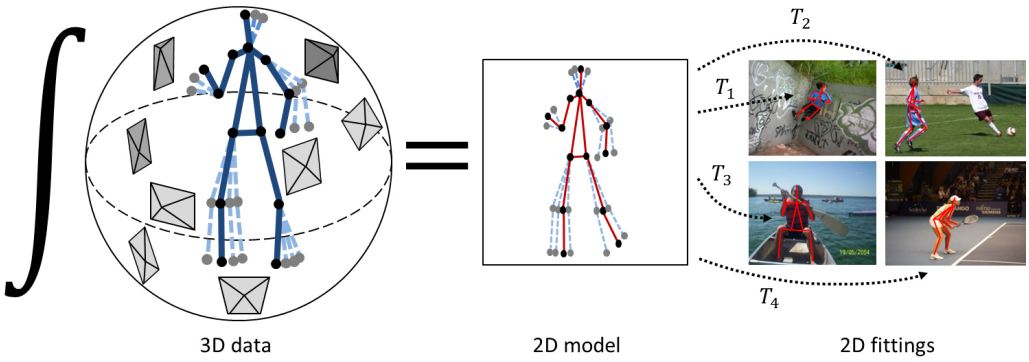
We first introduce statistical models as an approach to learn non-rigid deformations by means of a subspace, and review the most relevant state-of-the-art. Then we propose Discrete Subspace Procrustes Analysis (DSPA) to learn unbiased 2-D models from 3-D deformable objects. Next, we extend the discrete approach using Functional Data Analysis (FDA) into the efficient Continuous Subspace Procrustes Analysis (CSPA). Finally we evaluate both models, comparing them to the state-of-the-art of PA.

4.1 Statistical Models

In computer vision, Procrustes Analysis (PA) has been extensively used to align shapes (e.g., [80, 17]) and appearance (e.g., [95, 54]) as a pre-processing step to build 2-D models of shape variation. Usually, shape models are learned from a discrete set of 2-D landmarks through a two-step process [35]. Firstly, the rigid transformations are removed by aligning the training set w.r.t. the mean using PA; next, the remaining deformations are modeled using Principal Component Analysis (PCA) [74, 21]. In this process, PCA learns the non-rigid deformations of the data by keeping the subspace with the largest variance in the training set. Then, the subspace can be used to analyze if new shapes are plausible examples, or to generate new samples similar to the training ones. These models are called statistical models [17] and have been applied to solve problems such as object recognition [98, 49], facial feature detection and tracking [96, 17], and image segmentation [72, 65]. In particular, Point Distribution Models (PDMs) and Active Shape Models (ASMs) [17, 18] are among the most popular techniques to learn 2-D objects models. PDMs and ASMs build the shape models from a 2-D training set of image landmarks.

Statistical models have also been applied to learn appearance models invariant to geometric transformations. When applied to shapes, the geometric transformation computed by PA (e.g., $\mathbf{T}_i$ or $\mathbf{A}_i$) can be directly applied to the image coordinates. However, to align appearance features the geometric transformations have to be composed with the image coordinates, and the process is a bit more complicated. This is the main difference when applying PA to align appearance and shape. Frey and Jojic [33] proposed a method for learning a factor analysis model that is invariant to geometric transformations. The computational cost of this method grows polynomially with the number of possible spatial transformations and it can be computationally intensive when working with high-dimensional motion models. To improve upon that, De la Torre and Black [95] proposed parameterized component analysis: a method that learns a subspace of appearance invariant to affine transformations and extend it to non-linear appearance models [96]. Miller et al. proposed the congealing method [54], which uses an entropy measure to align images with respect to the distribution of the data. Cox et al. [9] extended [54] through a least-squares optimization. Kookinos and Yuille [52] proposed a probabilistic framework and extended previous approaches to deal with articulated objects using a Markov Random Field (MRF) on top of AAMs.

To address standard PA issues but extending the solution to the construction of multi-view statistical models, this chapter proposes a discrete



(a) (b) (c)

Figure 4.1: Illustration of Continuous Subspace Procrustes Analysis (CSPA), which builds an (b) unbiased 2-D model of human joints' variation by (a) integrating over all possible viewpoints of a 3-D motion capture data. This 2-D body shape model is used to (c) reconstruct 2-D shapes from different viewpoints. Our CSPA model generalizes across poses and camera views because it is learned from 3-D data.

and a continuous formulation of Subspace Procrustes Analysis (SPA). SPA is able to efficiently compute the non-rigid subspace of possible 2-D projections given several 3-D samples of a deformable object. Note that our proposed work is the *inverse* problem of Non-Rigid Structure From Motion (NRSFM) [101, 97, 15]. The goal of NRSFM is to recover 3-D shape models from 2-D tracked landmarks, while SPA builds unbiased 2-D models from 3-D data. As we show in the experimental section, the learned 2-D model has the same representational power as a 3-D model but leads to faster fitting algorithms [61]. SPA uniformly samples the space of possible 3-D rigid transformations, and it is extremely efficient in space and time. The main idea of SPA is to combine functional data analysis (see Section 3.1) with subspace estimation techniques.

Fig. 4.1 illustrates the approach proposed in this chapter. In Fig. 4.1 (a), we represent many samples of 3-D Motion Capture (MoCap) data of humans performing several activities. SPA aligns all 3-D samples projections, computes a 2-D subspace (Fig. 4.1 (b)) that can represent all possible projections of the 3-D MoCap samples under different camera views. Hence, SPA provides a simple, efficient and effective method to learn a 2-D subspace that accounts for non-rigid and 3-D geometric deformation of 3-D objects. These 2-D subspace models can be used for detection (i.e., constrain body parts, see Fig. 4.1 (c)), because the subspace models all 3-D rigid projections and

non-rigid deformations. As we will show in the experimental validation, the models learned by SPA are able to generalize better than existing PA approaches across view-points (because they are built using 3-D models) and preserve expressive non-rigid deformations. Moreover, computing SPA is extremely efficient in space and time.

4.2 Discrete Subspace Procrustes Analysis

Given a set $\mathcal{D} = \{\mathbf{D}_i \in \mathbb{R}^{3\times\ell}\}$ of n centered 3-D shapes, with $\mathbf{d}_i = \text{vec}(\mathbf{D}_i) \in \mathbb{R}^{3\ell\times 1}$, we can compute a set $\mathcal{P} = \{\mathbf{P}_j \in \mathbb{R}^{2\times 3}\}$ of r projections, one for each of the shapes. Then, Discrete Subspace PA (DSPA) extends PA by considering a subspace $\mathbf{B} \in \mathbb{R}^{2\ell\times k}$ and a set of weights $\mathbf{c}_{ij} \in \mathbb{R}^{k\times 1}$ which model the non-rigid deformations that the mean $\mathbf{M} \in \mathbb{R}^{2\times\ell}$ and the transformation $\mathbf{A}_{ij} \in \mathbb{R}^{2\times 2}$ are not able to reconstruct. DSPA finds $\mathbf{M}$, $\mathbf{B}$ and the sets $\mathcal{A} = \{\mathbf{A}_{ij}\}$ and $\mathcal{C} = \{\mathbf{c}_{ij}\}$ by minimizing the following function[1]:

$$E_{\text{DSPA}}(\mathbf{M}, \mathcal{A}, \mathbf{B}, \mathcal{C}) = \sum_{i=1}^{n}\sum_{j=1}^{r} \left\|\mathbf{P}_j\mathbf{D}_i - \mathbf{A}_{ij}\mathbf{M} - (\mathbf{c}_{ij}^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)}\right\|_F^2 = \quad (4.1)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{r} \left\|(\mathbf{I}_\ell \otimes \mathbf{P}_j)\mathbf{d}_i - (\mathbf{I}_\ell \otimes \mathbf{A}_{ij})\boldsymbol{\mu} - \mathbf{B}\mathbf{c}_{ij}\right\|_2^2, \quad (4.2)$$

where $\mathbf{P}_j = \mathbf{PR}(\boldsymbol{\omega}_j)$ is an orthographic projection onto the X-Y plane of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega}_j)$ in a given domain $\boldsymbol{\Omega}$, defined by the rotation angles $\boldsymbol{\omega}_j = \{\phi_j, \theta_j, \psi_j\} \in \mathbb{R}^3$, $\boldsymbol{\mu} = \text{vec}(\mathbf{M}) \in \mathbb{R}^{2\ell\times 1}$ is the vectorized version of the reference shape, $\mathbf{c}_{ij}$ is the k-dimensional weights vector of the subspace for each 2-D shape projection, and $\mathbf{B}^{(2)} \in \mathbb{R}^{2k\times\ell}$ is the reshaped subspace. Observe that the difference with Eq. (2.3) is that we have added a subspace. This subspace will compensate for the non-rigid components of the 3-D object and the rigid component (3-D rotation and projection to the image plane) that the affine transformation cannot model (see Fig. 4.2 (a), where the first three bases of the subspace capture rigid and non-rigid deformations). Recall that a 3-D rigid object under orthographic projection can be recovered with a three-dimensional subspace [94] (if the mean is removed), but PA cannot recover it because it is only rank two. Also, observe that the coefficients vector $\mathbf{c}_{ij}$ depends on two indexes, i for the object and j for the geometric projection. Dependency of $\mathbf{c}_{ij}$ on the geometric projection is a key point. If j index is not considered, the subspace would not be able to capture the variations in pose and its usefulness for our purposes would be unclear.

[1]See Section 1.4 for an explanation of the vec-transpose operator.

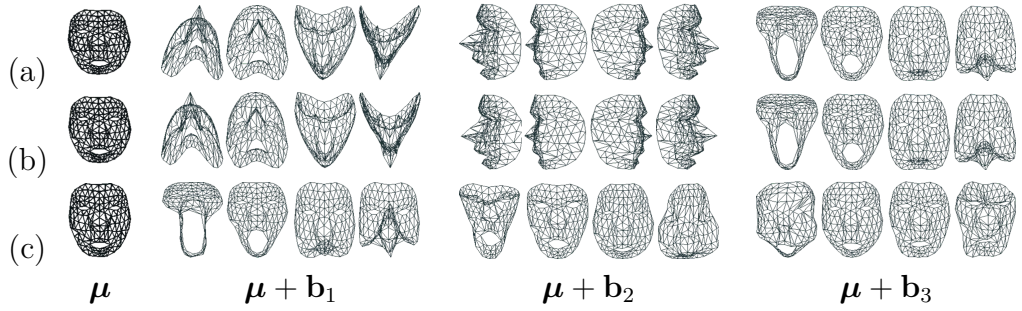


Figure 4.2: Illustration of the reference shape ($\boldsymbol{\mu}$) and the first three bases ($\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$) of the 2-D subspace models from (a) *DSPA*, and (b) *CSPA*; as well as a conventional (c) 3-D model (PA + PCA). We sampled each basis 4 times between the standard limits [18] to show their deformation behavior. All models were trained on FaceWarehouse [16] dataset, with 10 3-D faces from expressions number 0 and 1 (neutral and open mouth, respectively). Pitch an yaw integration limits were set to $\phi, \theta \in [-\pi/2, \pi/2]$ for (b), and 100 projections were generated for each 3-D shape within the same interval to train (a). Note that $\boldsymbol{\mu}$ and $\mathbf{b}_i$ in (c) are 3-D. They are projected frontally for a better comparison.

Although Eq. (4.1) and the NRSFM problem follow similar formulation [15], the assumptions are different and variables have opposite meanings. For instance, the NRSFM assumptions about rigid transformations do not apply here, since $\mathbf{A}_{ij}$ are affine transformations in our case.

Given an initialization of $\mathbf{B} = 0$, the function error $E_{\text{DSPA}}(\mathbf{M}, \mathcal{A}, \mathbf{B}, \mathcal{C})$ associated to the DSPA algorithm is minimized by finding the transformations $\mathbf{A}^*_{ij}$ and reference shape $\mathbf{M}^*$ that minimize Eq. (2.3), using the same Alternate Least Squares (ALS) framework as in PA. Then, we substitute $\mathbf{A}^*_{ij}$ and $\mathbf{M}^*$ in Eq. (4.2), resulting in the expression:

$$E_{\text{DSPA}}(\mathbf{B}, \mathcal{C}) = \sum_{i=1}^{n} \sum_{j=1}^{r} \left\| \widetilde{\mathbf{D}}_{ij} - (\mathbf{c}_{ij}^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 = \tag{4.3}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{r} \left\| \widetilde{\mathbf{d}}_{ij} - \mathbf{B}\mathbf{c}_{ij} \right\|_2^2 = \left\| \widetilde{\mathbf{D}} - \mathbf{B}\mathbf{C} \right\|_F^2, \tag{4.4}$$

where $\widetilde{\mathbf{D}}_{ij} = \mathbf{P}_j\mathbf{D}_i - \mathbf{A}^*_{ij}\mathbf{M}^* \in \mathbb{R}^{2\times\ell}$, $\widetilde{\mathbf{d}}_{ij} = \text{vec}(\widetilde{\mathbf{D}}_{ij}) \in \mathbb{R}^{2\ell\times 1}$, $\widetilde{\mathbf{D}} = [\widetilde{\mathbf{d}}_1 \dots \widetilde{\mathbf{d}}_{nr}] \in \mathbb{R}^{2\ell\times nr}$, and $\mathbf{C} \in \mathbb{R}^{k\times nr}$. Finally, we can find the global optima of Eq. (4.4) by Singular Value Decomposition (SVD): $\mathbf{B} = \mathbf{U}$ and $\mathbf{C} = \mathbf{S}\mathbf{V}^T$, where $\widetilde{\mathbf{D}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$.

4.3 Continuous Subspace Procrustes Analysis

As it was discussed in the Chapter 3, the discrete formulation is not efficient in space nor time, and might suffer from not uniform sampling of the rotation space. CSPA generalizes DSPA by re-writting it in a continuous formulation. CSPA minimizes the following functional:

$$E_{\text{CSPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i, \mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) =$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}(\boldsymbol{\omega})_i\mathbf{M} - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} = \tag{4.5}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \|(\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\mathbf{d}_i - (\mathbf{I}_\ell \otimes \mathbf{A}(\boldsymbol{\omega})_i)\boldsymbol{\mu} - \mathbf{B}\mathbf{c}(\boldsymbol{\omega})_i\|_2^2 \, d\boldsymbol{\omega}, \tag{4.6}$$

where $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$. The main difference between Eq. (4.6) and Eq. (4.2) is that the entries in $\mathbf{c}(\boldsymbol{\omega})_i \in \mathbb{R}^{k\times 1}$, $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$ and $\mathbf{A}(\boldsymbol{\omega})_i \in \mathbb{R}^{2\times 2}$ are not scalars anymore, but functions of integration angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$.

Given an initialization of $\mathbf{B} = 0$, and similarly to the DSPA model, CSPA is minimized by finding the optimal reference shape $\mathbf{M}^*$ that minimizes Eq. (3.7). We used the same fixed-point framework as CPA. Given the value of $\mathbf{M}^*$ and the expression of $\mathbf{A}(\boldsymbol{\omega})_i^*$ from Eq. (3.8), we substitute them in Eq. (4.6) resulting in:

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\bar{\mathbf{D}}_i - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} = \tag{4.7}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i - \mathbf{B}\mathbf{c}(\boldsymbol{\omega})_i \right\|_2^2 d\boldsymbol{\omega}, \tag{4.8}$$

where $\bar{\mathbf{D}}_i = \mathbf{D}_i(\mathbf{I}_\ell - (\mathbf{M}^{*T}(\mathbf{M}^*\mathbf{M}^{*T})^{-1}\mathbf{M}^*))$ and $\bar{\mathbf{d}}_i = \text{vec}(\bar{\mathbf{D}}_i)$. We can find the global optima of Eq. (4.8) by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$.

After some algebra (see Appendix B) we show that the covariance matrix $\boldsymbol{\Sigma} = ((\mathbf{I}_\ell \otimes \mathbf{Y})\,\text{vec}(\sum_{i=1}^{n}\sum_{j=1}^{r}\bar{\mathbf{d}}_{ij}\bar{\mathbf{d}}_{ij}^T))^{(2\ell)}$, where the definite integral $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))d\boldsymbol{\omega} \in \mathbb{R}^{2\ell\times 2\ell}$ can be computed off-line, leading to an efficient optimization in space and time. Though the number of elements in matrix $\mathbf{Y}$ increases quadratically with the number of landmarks ℓ, note that the integration time is constant since $\mathbf{Y}$ has a sparse structure with only 36 different non-zero values (recall that $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$).

Although $\mathbf{A}(\boldsymbol{\omega})_i$ and $\mathbf{c}(\boldsymbol{\omega})_i$ are not explicitly computed during training, this is not a limitation compared to DSPA. During testing time, training

values of $\mathbf{c}(\boldsymbol{\omega})_i$ are not needed. Only the deformation limits in each principal direction of $\mathbf{B}$ are required. These limits depend on eigenvalues [18], which are computed with CSPA. The three principal bases between these limits are illustrated in Fig. 4.2. We show how the first 2 bases of CSPA (Fig. 4.2 (b)) and DSPA (Fig. 4.2 (a)) learn viewpoint changes, as well as the common expression for all the subjects in the training set (mouth opening) is learned as the third basis. Note that the 3-D model (Fig. 4.2 (c)) learns the common facial expression in the first basis (because 3-D shapes are not rotated to train the 3-D model), and its following bases model inter-person differences. These distinctive person characteristics are also learned by SPA models in their following bases.

4.4 Experiments

This section illustrates the benefits of DSPA and CSPA, and compares them with state-of-the-art PA methods to build 2-D shape models of faces and human body joints' variation. First, we compare the performance of PA+PCA and SPA to build a 2-D shape model of faces from the FaceWarehouse [16] dataset (Experiment 1). Next, we compare our discrete and continuous approaches in a large scale experiment (Experiment 2). Afterwards, we learn Motion Capture (MoCap) joint's variation of bodies using the Carnegie Mellon University (CMU) MoCap dataset [1], and we compare them with the state-of-the-art (Experiment 3) and in a large scale experiment (Experiment 4). Finally, we show the benefits of our continuous 2-D model (CSPA) over 3-D models (Experiment 5) in the same datasets.

4.4.1 Learning 2-D Face Models

The aim of Experiments 1 and 2 is to build a generic 2-D face model that can reconstruct non-rigid facial deformation under a large range of 3-D rotations. For training and testing, we used the FaceWarehouse dataset (detailed in Section 2.4.1). We sub-sampled the original mesh to 49 and 162 landmarks, depending on the experiment. We rotated the 3-D faces in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$. Rotations were uniformly selected and we report results for 300 rotations for testing, while increasing the number of viewpoints in training. We report the Mean Squared Error (MSE) relative to the intra-eye size for 100 realizations, and we plot the MSE and the half of the standard deviation.

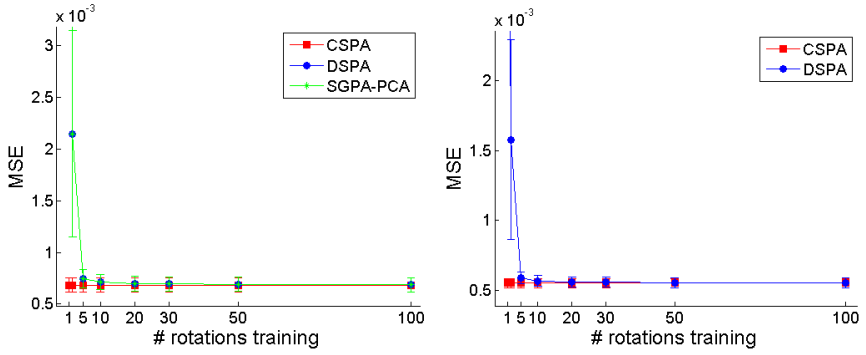


Figure 4.3: Comparisons of (*a*) *CSPA*, *DSPA*, and *SGPA-PCA* (Experiment 1) ; and (*a*) *CSPA* and *DSPA* (Experiment 2) as a function of the number of training viewpoint projections, using a subspace of 25 bases for all models.

Experiment 1: Comparison with State-of-the-Art PA Methods on Faces

This section compares DSPA and CSPA methods with the state-of-the-art Stratified Generalized Procrustes Analysis (SGPA)[2] [10]. For training we randomly selected 20 subjects, three expressions per subject and 49 landmarks (this is due to the memory limitations of SGPA). For testing we randomly selected 10 different subjects with the same three expressions as training. We report results varying the number of training rotations between $1 \sim 100$.

Similarly to the experimental section in Chapter 3, among the different SGPA versions, we selected the "Affine-factorization" with the data-space model to make a fair comparison with our method. Recall that under our assumption of non-missing data "Affine-All" and "Affine-factorization" achieve the global optimum, being "Affine-factorization" faster.

Fig. 4.3 (a) compares the mean reconstruction error for DSPA, CSPA, and SGPA followed by PCA (we will refer to this method SGPA-PCA). From the figure, one can observe that error in the test for DSPA and SGPA-PCA decreases with the number of rotations in training, and it converges to CSPA, which provides a bound on the lower error. Observe, that we used 60 3-D faces (20 subjects and 3 expressions) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$, and DSPA and SGPA-PCA needed about 20 rotations to achieve similar results to CSPA. In this case, discrete methods need 20 times more space than the continuous one. Execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 1.25 sec.

[2]The code was downloaded from author's website (http://isit.u-clermont1.fr/~ab).

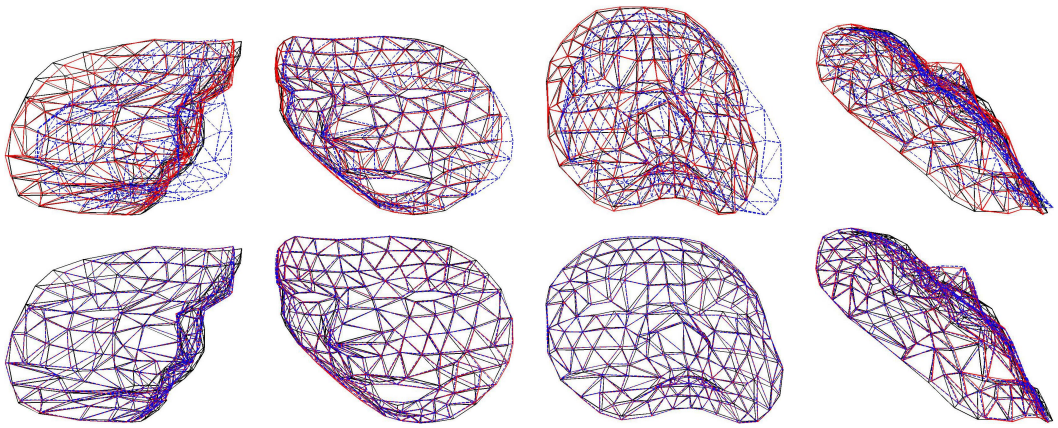


Figure 4.4: Experiment 2 results with 1 (*top*) and 20 (*bottom*) rotations. *CSPA* (*solid red lines*) and *DSPA* (*dashed blue lines*) face reconstructions over ground truth (*solid black lines*).

(DSPA), 0.38 sec. (CSPA) and 2.36 sec. (SGPA-PCA).

Experiment 2: Comparison between CSPA and DSPA

This experiment compares DSPA and CSPA in a large-scale problem as a function of the number of rotations between $1 \sim 100$. For training we randomly selected 120 subjects, five expressions per subject and 162 landmarks. For testing we randomly selected 30 different subjects with the same five expressions as training.

Fig. 4.3 (b) shows the mean reconstruction error in test comparing DSPA and CSPA. As expected, DSPA converges to CSPA as the number of training rotations increases. However, observe that CSPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training faces (120 subjects and 5 expressions) and rotation domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 20 2-D viewpoint projections to achieve similar results to CSPA. Thus, discrete model DSPA needs 20 times more storage space than CSPA. Execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 13.05 sec. (DSPA) and 3.17 sec. (CSPA).

Qualitative results from CSPA and DSPA models trained with different number of rotations are shown in Fig. 4.4. Note that training DSPA model with 1 rotation (*top*) results in not properly reconstructed faces. However, training it with 20 rotations (*bottom*) leads to reconstructions almost as accurate as made by CSPA.

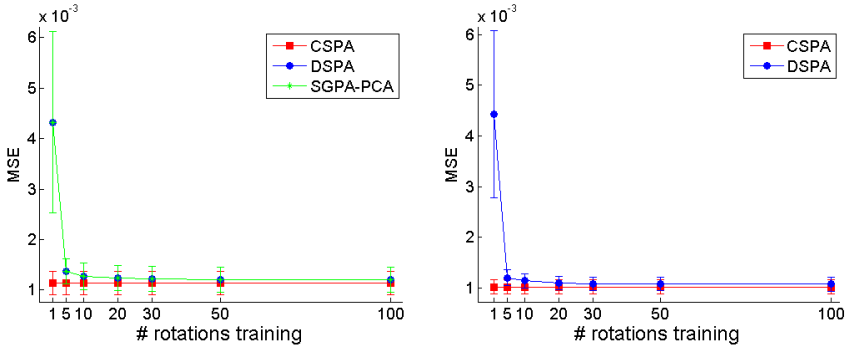


Figure 4.5: Comparisons of (*a*) *CSPA*, *DSPA*, and *SGPA-PCA* using a subspace of 9 bases (Experiment 3); and (*b*) *CSPA* and *DSPA* using a subspace of 12 bases (Experiment 3) as a function of the number of training viewpoint projections.

4.4.2 Learning 2-D Human Joints' Variation Models

The aim of Experiments 3 and 4 is to build a generic 2-D skeleton model from 3-D Motion Capture (MoCap). For training and testing, we used the Carnegie Mellon University MoCap dataset (detailed in Section 2.4.2). Skeletons with 31 joints are provided, as well as RGB video recordings for several sequences. We trained our models using the set of 14 landmarks as is common across several databases for human pose estimation, and we rotated the shapes in the same way as the experiments 1 and 2. We report the MSE relative to the torso size for 100 realizations, and we plot the MSE and the half of the standard deviation.

Experiment 3: Comparison with State-of-the-Art PA Methods

This section compares DSPA and CSPA methods with the state-of-the-art Stratified SGPA [10]. For training we randomly selected 3 sequences with 30 frames per sequence from the set of 11 running sequences of the user number 9. For testing we randomly selected 2 sequences with 30 frames from the same set. We report results varying the number of considered viewpoints in training between $1 \sim 100$ rotations.

Fig. 4.5 (a) compares the mean reconstruction error for DSPA, CSPA, and SGPA followed by PCA (we will refer to this method SGPA-PCA). From the figure, one can observe that the mean error in test for DSPA and SGPA-PCA decreases with the number of rotations in training, and it converges to CSPA. CSPA provides a bound on the lower error. Observe, that we used 90 3-D bodies (3 sequences with 30 frames) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$,

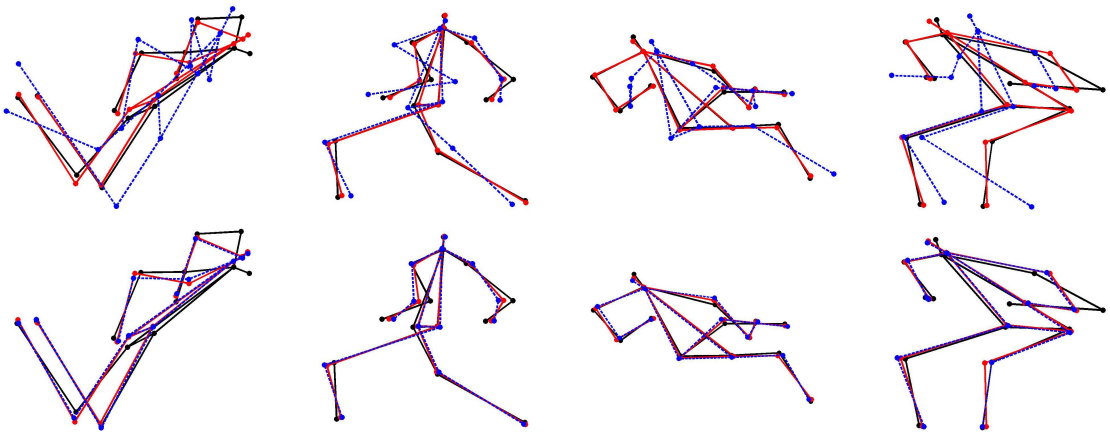


Figure 4.6: Experiment 2 results with 1 (*top*), and 30 (*bottom*) rotations. Examples show skeleton reconstructions from continuous (*CSPA* in *solid red lines*) and discrete (*SPA* in *dashed blue lines*) models over ground truth (*solid black lines*).

and DSPA and SGPA-PCA needed about 30 rotations to achieve similar results to CSPA. Therefore, in this case, discrete methods need 30 times more space than the continuous one. Execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 1.44 sec. (DSPA), 0.03 sec. (CSPA) and 3.54 sec. (SGPA-PCA).

Experiment 4: Comparison between CSPA and DSPA

This experiment compares DSPA and CSPA in a large-scale problem as a function of the number of rotations. For training we randomly selected 20 sequences with 30 frames per sequence. For testing we randomly selected 5 sequences with 30 frames. We report results varying the number of viewpoints in training between $1 \sim 100$ rotations.

Fig. 4.5 (b) shows the mean reconstruction error in test comparing DSPA and CSPA. As expected, DSPA converges to CSPA as the number of training rotations increases. However, observe that CSPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training bodies (20 sequences with 30 frames) and domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 30 2-D viewpoint projections to achieve similar results to CSPA. Thus, the discrete model DSPA needs 30 times more storage space than CSPA. Execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 15.89 sec. (DSPA) and 0.05 sec. (CSPA).

Qualitative results from CSPA and DSPA models trained with different number of rotations are shown in Fig. 4.6. Note that training DSPA model

with 1 rotation (*top*) results in poor reconstruction. However, training it with 30 rotations (*bottom*) leads to reconstructions almost as accurate as made by CSPA.

4.4.3 Experiment 5: 2-D vs 3-D Models

In previous experiments we have shown that 2-D models learned from 3-D data overcome typical 2-D models learned directly from 2-D data. This is because the use of 3-D data allows us to build unbiased models, able to generalize among different viewpoints. The question that strikes at this point is: why to project the 3-D data in training time, building 2-D models? Why do not learn 3-D models and project them to the image plane during test time? From the comparison between 2-D and 3-D face models performed in [61], one concludes that both models have the same representation power, with 2-D models being faster in real-time fitting.

This section compares unbiased 2-D to 3-D models of faces and skeletons, in order to check that the same conclusions apply to faces and body MoCap data. In this comparison the 2-D model will be the CSPA model from Eq. (4.5). On the other hand, we will train the standard 3-D model optimizing Eq. (2.2) with the number of dimensions $d = 3$ and $\mathbf{A} \in \mathbb{R}^{3\times3}$ being a rotation matrix, followed by a PCA. For the 2-D fitting of the 3-D model, we will use the standard algorithm from [39, 105], where the deformation parameters $\mathbf{c}_{3\mathrm{D}} \in \mathbb{R}^{k_{3\mathrm{D}}\times1}$ of the 3-D model $\mathbf{M}_{3\mathrm{D}} + (\mathbf{B}_{3\mathrm{D}}\mathbf{c}_{3\mathrm{D}})^{(3)}$, as well as the rotation and scaling of the projection matrix $\mathbf{P} \in \mathbb{R}^{2\times3}$, are estimated until convergence in a 2-step iterative algorithm[3]. For a fair comparison between models, the intrinsic camera matrix in $\mathbf{P}$ is fixed to be a scaled orthographic projection.

We compared 2-D and 3-D methods on FaceWarehouse and CMU MoCap datasets for faces and body joints' modeling, respectively. For both datasets, we performed the comparison in different angle domains ($\phi, \theta \in [-\pi/4, \pi/4]$ and $\phi, \theta \in [-\pi/2, \pi/2]$) for training and test, and we report results varying the number of subspace bases for both 2-D and 3-D models. For training the models on the FaceWarehouse dataset we randomly selected 120 subjects, 20 expressions per subject and 162 landmarks. For testing we randomly selected 30 different subjects with the same 20 expressions (all expressions of the dataset). For training the models on the CMU MoCap dataset we randomly selected 80 sequences with 30 frames per sequence and 14 landmarks. For testing we randomly selected 20 different sequences with 30 frames. Recall

[3]The code was downloaded from author's website and adapted to our own code (http://www.research.rutgers.edu/~feiyang/web2/face_morphing.htm).

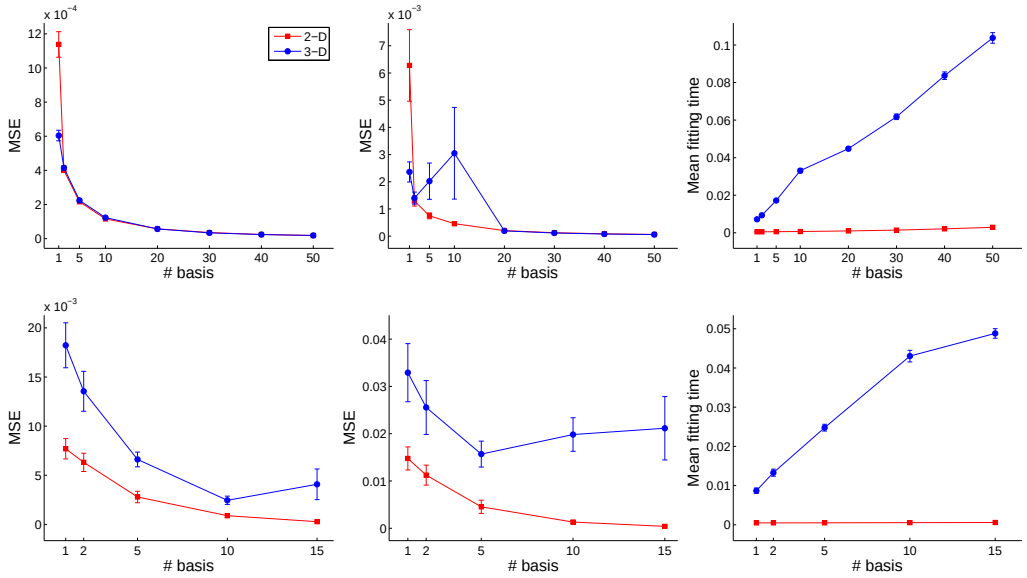


Figure 4.7: Experiment 5 results on (*top*) FaceWarehouse and (*bottom*) CMU MoCap datasets within $[-\pi/4, \pi/4]$ and $[-\pi/2, \pi/2]$ angle domains. Comparisons between 2-D and 3-D models as a function of the number of subspace bases, in terms of (*a*) mean reconstruction error and (*b*) mean fitting time (extremely similar mean times for both experiments).

that all models in this experiment are trained with 3-D data. For testing, we rotated and projected 30 times each test shape.

Fig. 4.7 (a) shows the mean reconstruction error for 100 realizations, as well as the half of the standard deviation, incrementing the number of bases of the subspace models. We show the MSE for both experiments performed within $[-\pi/4, \pi/4]$ and $[-\pi/2, \pi/2]$ angle domains. Fig. 4.7 (b) reports the mean fitting time. Since experiments in both angle domains have similar test times, we only provide the time for one of them ($[-\pi/2, \pi/2]$) to avoid redundancy.

Fig. 4.7 (*top*) reports the comparison on FaceWarehouse dataset. For narrow angle domains ($[-\pi/4, \pi/4]$), both 3-D and 2-D face models have similar performance, but 2-D models being faster (see Fig. 4.7 (b)). However, 2-D models are more stable (lower deviation width) than 3-D models in the experiment with a wider test domain ($[-\pi/2, \pi/2]$). The fitting algorithm between the 3-D model and the 2-D test shape fails to estimate the projection matrix under extreme viewpoints, leading to a poor convergence. Note that

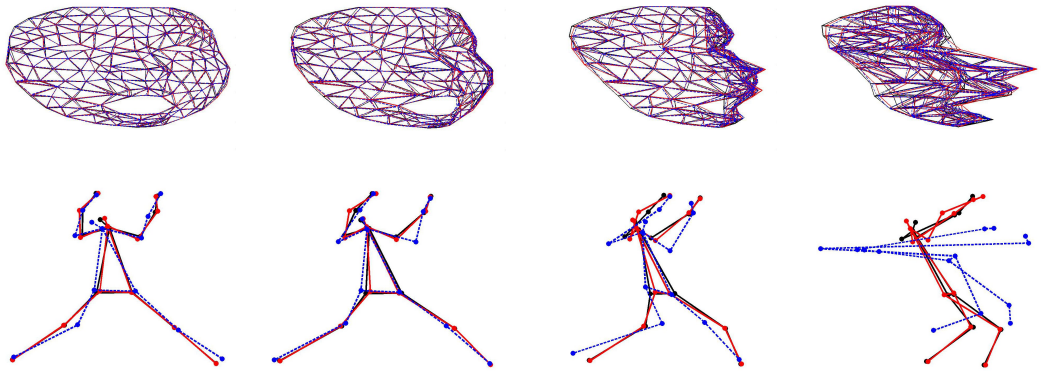


Figure 4.8: Qualitative results from Experiment 5, rotating the test shapes in yaw on FaceWarehouse (*top*) and CMU MoCap (*bottom*) datasets. 2-D model (*solid red lines*) and 3-D model (*dashed blue lines*) reconstructions over ground truth (*solid black lines*). For both models, the number of bases was $k = 14$ on CMU MoCap dataset, and $k = 25$ on FaceWarehouse dataset.

the 3-D subspace will absorb the poorly estimated projection matrices, with a sufficient number of bases.

The same effect occurs with models of body joints' variation in Fig. 4.7 (*bottom*), however, 2-D models outperform 3-D for any number of bases on CMU MoCap dataset. Although the performance deteriorates on both datasets under large rotations, this is more evident on CMU MoCap dataset due to the high variability non-rigid deformations of the human body (see Fig. 4.8).

Note that in those situations where 2-D models obtain similar reconstruction error than 3-D models, increasing the number of bases of the 2-D model would lead to more accurate reconstructions than 3-D models, still preserving faster fittings in test.

4.5 Discussion: How to Build a 2-D Model from a 3-D Model

We argued that unbiased 2-D and 3-D models have the same reconstruction power, being 2-D models faster, as well as we detailed how to build multi-view 2-D models from 3-D data. However, we might be interested in building an unbiased 2-D model even though we do not have access to the 3-D training data (e.g. 3-D NRSFM model built from 2-D data). In this section we discuss how to build a 2-D model directly from a 3-D model, integrating over

all possible viewpoints but also along the deformation parameters. Clearly, building a model from a previous learned model will lead to a loose of information, but benefits in some applications (e.g. real-time fitting, enlarged pose variation models) can outweigh the information loss.

A method to *downgrade* a 3-D model to its homologous in 2-D was presented in [61]. They generated a 2-D dataset by a systematic sampling of the deformation and rotation parameters of the 3-D model. Then, they built a 2-D model from this enhanced 2-D dataset in a conventional manner. However, a uniform sampling of the rotation angles does not lead to a uniform sampling of the rotation space $SO(3)$. In addition, it is not clear how much sub-sampling is needed in the deformation parameters in order to generate a synthetic dataset with similar variance to the original training data. Just to give some example values, imagine that our model has only $k = 10$ bases, and we need $r = 20$ rotations to cover the domain of viewpoints that we are modeling. If we sample 4 times each axis of variance, we will need over $2 \cdot 10^7$ 2-D samples to train the 2-D model. Note that handle this dataset would be a large scale problem, even though we did not take extreme values for the example.

As a proof of concept, we discuss here how to build a 2-D model directly from a 3-D model, ensuring a uniform coverage of the rotation space, without the need of generating a huge synthetic 2-D dataset. Given a 3-D model composed by a mean $\mathbf{M}_{3\text{D}} \in \mathbb{R}^{3\times\ell}$, the $k_{3\text{D}}$ bases $\mathbf{B}_{3\text{D}} \in \mathbb{R}^{3\ell\times k_{3\text{D}}}$, and their corresponding eigenvalues $\boldsymbol{\lambda}_{3\text{D}} \in \mathbb{R}^{k_{3\text{D}}\times 1}$, we build a 2-D model ($\mathbf{M} \in \mathbb{R}^{2\times\ell}$, $\mathbf{B} \in \mathbb{R}^{2\ell\times k}$) by integrating along the axis of variance $\mathbf{B}_{3\text{D}}$ within a domain $\boldsymbol{\Upsilon}$, depending on the eigenvalues $\boldsymbol{\lambda}_{3\text{D}}$, as we will discuss afterwards. Moreover, similarly to CSPA, we rotate and project the 3-D model to the image plane using $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$. Note that we ensure uniformity in $SO(3)$ by means of the Haar measure $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$ defined in the integral [68].

Given the 3-D model and the rotation domain $\boldsymbol{\Omega}$, we find its homologous 2-D model by minimizing the following error[4]:

$$E_{\text{2D-3D}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i, \mathbf{B}, \mathbf{c}(\boldsymbol{\omega})) = \int_{\boldsymbol{\Upsilon}}\int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega}) \left[\mathbf{M}_{3\text{D}} + (\mathbf{c}_{3\text{D}}(\boldsymbol{\upsilon})^T \otimes \mathbf{I}_3)\mathbf{B}_{3\text{D}}^{(3)}\right] - \mathbf{A}(\boldsymbol{\omega},\boldsymbol{\upsilon})\mathbf{M} - (\mathbf{c}(\boldsymbol{\omega},\boldsymbol{\upsilon})^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} d\boldsymbol{\upsilon} \tag{4.9}$$

where $\mathbf{P}(\boldsymbol{\omega})$ is an orthographic projection of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega})$ in the given domain $\boldsymbol{\Omega}$, defined by the rotation angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. The main difference between Eq. (4.5) and Eq. (4.9) is that instead of learning a 2-D model from 3-D shapes, our input now is a 3-D model. Hence, entries in the affinity

[4]See Section 1.4 for an explanation of the vec-transpose operator.

transformation $\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\upsilon}) \in \mathbb{R}^{2\times 2}$ and the subspace weights $\mathbf{c}(\boldsymbol{\omega}, \boldsymbol{\upsilon}) \in \mathbb{R}^{k\times 1}$, $\mathbf{c}_{3\mathrm{D}}(\boldsymbol{\upsilon}) \in \mathbb{R}^{k_{3\mathrm{D}}\times 1}$ are not only functions of the integration angles $\boldsymbol{\omega}$, but also functions of the deformation parameters $\boldsymbol{\upsilon} = \{\upsilon_1, \ldots, \upsilon_{k_{3\mathrm{D}}}\}$.

In addition, 2-D modeling from a 3-D model would be efficient, since the diagonal matrix $\mathbf{W} = \int_{\boldsymbol{\Upsilon}} \mathbf{c}_{3\mathrm{D}}(\boldsymbol{\upsilon})^T \mathbf{c}_{3\mathrm{D}}(\boldsymbol{\upsilon}) d\boldsymbol{\upsilon}$ encoding the deformations does not require the explicit computation of the definite integral. This statement comes from solving:

$$E_{\mathrm{CPACA}}(\mathbf{B}, \mathbf{c}_i) = \int_{\boldsymbol{\Upsilon}} \|[\boldsymbol{\mu}_1 + \mathbf{B}_1\mathbf{c}_1(\boldsymbol{\upsilon})] - \mathbf{B}_2\mathbf{c}_2(\boldsymbol{\upsilon})\|_2^2 \, d\boldsymbol{\upsilon}, \tag{4.10}$$

where, assuming zero mean $\boldsymbol{\mu}_1 = \mathrm{vec}(\mathbf{M}_1)$, we find that $\boldsymbol{\Sigma}_2 = \mathbf{B}_1\mathbf{W}\mathbf{B}_1^T$. Since $\boldsymbol{\Sigma}_1 = \mathbf{B}_1\boldsymbol{\Lambda}_1\mathbf{B}_1^T$, and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, we find that the optimal value for the matrix encoding the deformations is the diagonal matrix containing the eigenvalues of the 3-D model $\mathbf{W} = \mathrm{diag}(\boldsymbol{\lambda}_{3\mathrm{D}})$.

Similarly to CPA model (see Section 2) we find $\mathbf{M}$ by minimizing Eq. (4.9) using fixed point minimization (i.e. Eq. 3.10), where:

$$\mathbf{Z} = (\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M}\left(\mathbf{M}_{3\mathrm{D}}^T\mathbf{X}\mathbf{M}_{3\mathrm{D}} + \mathbf{B}_{3\mathrm{D}}^T((\mathbf{N}\otimes\mathbf{I}_3)\,\mathrm{vec}(\mathbf{X}))^{(3\mathbf{k}_{3D})}\mathbf{B}_{3\mathrm{D}}\right). \tag{4.11}$$

Matrix $\mathbf{N} = (\mathbf{c}_{3\mathrm{D}}(\boldsymbol{\upsilon}) \otimes \mathbf{I}_3 \otimes \mathbf{c}_{3\mathrm{D}}(\boldsymbol{\upsilon}))$ is a sparse matrix, with the nonzero elements being the eigenvalues in $\mathbf{W}$, and $\mathbf{X} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T\mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances.

Similarly to CSPA model (see Section 4), substituting the optimal $\mathbf{M}^*$ and the expression $\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\upsilon})$ in Eq. (4.9), allows us to find the optimal $\mathbf{B}$ by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$, and the covariance matrix $\boldsymbol{\Sigma} = ((\mathbf{I}_\ell \otimes \mathbf{Y})\,\mathrm{vec}\,[\mathbf{L}])^{(2\ell)}$, being $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega})) d\boldsymbol{\omega} \in \mathbb{R}^{2\ell\times 2\ell}$ and $\mathbf{L} = \boldsymbol{\mu}_{3D}\boldsymbol{\mu}_{3D}^T + \mathbf{B}_{3\mathrm{D}}\mathbf{W}\mathbf{B}_{3\mathrm{D}}^T \in \mathbb{R}^{2\ell\times 2\ell}$.

As we illustrate in Fig. 4.2 and Fig. 4.9, our 2-D model Fig. 4.9 (a) built directly from a 3-D model Fig. 4.9 (b) have the same behavior that those models learned from the original 3-D data, Fig. 4.2 (a-b), rotated and projected to 2-D.

4.6 Conclusions

In previous chapters we formalized the construction of multi-view 2-D rigid models from 3-D data, in a discrete (see Projected PA in Chapter 2) and a continuous (see Continuous PA in Chapter 3) way. PPA and CPA build unbiased rigid models by extending the standard PA, CPA being much more efficient in space and time. Although CPA overcomes PA limitations, CPA

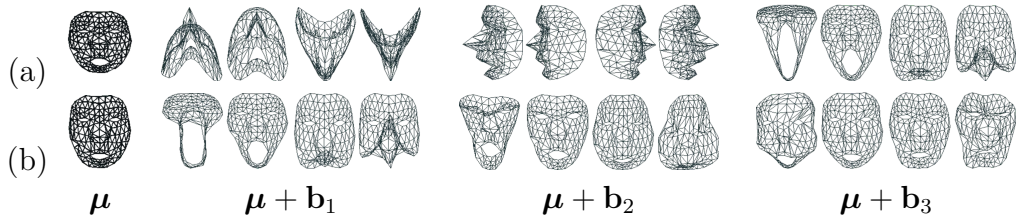


Figure 4.9: Illustration of the reference shape ($\boldsymbol{\mu}$) and the first three bases ($\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$) of the 2-D subspace model (a) directly build from 3-D model (b). We sampled each basis 4 times between the standard limits [18] to show their deformation behavior. All models were trained on FaceWarehouse [16] dataset, with 10 3-D faces from expressions number 0 and 1 (neutral and open mouth, respectively). Pitch an yaw integration limits were set to $\phi, \theta \in [-\pi/2, \pi/2]$ to train (a). Note that $\boldsymbol{\mu}$ and $\mathbf{b}_i$ in (b) are 3-D. They are projected frontally for a better comparison.

efficiency is limited to rigid models. For instance, modeling a subspace following the standard methodology based on CPA would still require to enhance the training set with several 2-D projections of each 3-D sample, to model non-rigid deformations of the training set.

To address standard PA issues but extending the solution to the construction of multi-view statistical models, in this chapter we proposed Subspace Procrustes Analysis (SPA). SPA is able to efficiently compute a 2-D subspace of rigid and non-rigid deformations of 3-D objects. We proposed two models, one discrete (DSPA) that samples the 3-D rotation space, and one continuous (CSPA) that integrates over $SO(3)$.

In the experimental section we compared SPA models against the state-of-the-art PA methods on faces and bodies datasets, while raising the number of rotations in the training set. As the number of projections increased, DPSA converged to CSPA. CSPA has two advantages over traditional PA, PPA: (1) it generates unbiased deformable models because it uniformly covers the space of projections, and (2) it is much more efficient in space and time. Experiments comparing 2-D SPA models of faces and bodies show improvements w.r.t. state-of-the-art PA methods.

Moreover, we also compared the performance of our multi-view 2-D models trained on 3-D datasets against standard 3-D models, projected in fitting time to the 2-D test. We showed that our 2-D models are as expressive as 3-D models, but SPA models being faster in test time. Finally, we also discussed the possibility of building 2-D models directly from 3-D models, by integrating along the deformation domain. We found this application interesting for those problems where 3-D data are not available, and we will study in-depth

this possibility in future research.

Therefore, SPA extends PA by building multi-view deformable models from 3-D datasets. DSPA and CSPA provide efficient approaches to build unbiased 2-D models, able to generalize to different viewpoints, but CSPA building models more efficiently in space and time. In addition, our unbiased 2-D models are also efficient in test fitting, as we showed on faces and bodies datasets. Finally, in the next chapter we will illustrate the performance of our multi-view 2-D models in the task of human pose estimation.

Chapter 5

Human Pose Estimation

Human pose estimation from still images refers to recover the configuration of body parts [6, 91, 107, 78], usually, after finding those image cues that represent body parts or joints. The high variation of human postures from different viewpoints makes this problem really challenging. The common approach to handle large viewpoint variations is to train the models with several labeled images from different viewpoints [28, 6, 73, 107, 78, 79, 42]. However this approach has some drawbacks: (1) it is not clear the extent to which the dataset must be enhanced with images from different viewpoints in order to build unbiased 2-D models; (2) extend the training set without this evaluation would unnecessary increase memory and computation requirements to train the models; (3) obtaining new labeled images from different viewpoints could be a difficult task because the expensive labeling cost; and finally, (4) a non uniform coverage of the different viewpoints of a person leads to biased 2-D models.

In this dissertation we have proposed to solve these drawbacks by changing the paradigm, learning 2-D multi-view models from 3-D datasets instead of learning them from 2-D images. In previous chapters we extended Procrustes Analysis (PA) to build multi-view 2-D models by rotating and projecting 3-D data samples. By means of Continuous Subspace PA (see Chapter 4) we modeled rigid and non-rigid deformations of 3-D motion capture sequences, in an efficient manner. Finally, in this chapter we illustrate the benefits of using multi-view 2-D models in the task of human pose estimation.

We first introduce the human pose estimation problem and review the most relevant state-of-the-art. Next, we reformulate the problem as a feature selection by subspace matching, and introduce our approach for this task. Finally we evaluate our multi-view 2-D deformable models (see Chapter 4) in combination with the feature selection method in the problem of human pose estimation, compared to the state-of-the-art on the Leeds Sports dataset [48].

5.1 Human Pose Estimation

In order to recover the human pose from images, state-of-the-art approaches [73, 107, 78, 79] use discriminative detectors (e.g. HOG [20] filters) to estimate the likelihood of image pixels to belong to each body part. Then, body configurations are usually modeled as pairwise constraints between body parts, with generative [28, 6, 78, 79] or discriminative [73, 107, 42] models, also trained from labeled images. These constraints are usually modeled as edges of a graph, whose nodes are the body joints or limbs. Configurations range from loopy graph models [85, 47, 93, 90] to trees [28, 6, 107]. In order to handle human body variations and viewpoints, part relations tend to be loosely modeled (e.g. Gaussian distribution), and efficiency of the tree structure leads to take into account only consecutive body parts (e.g. connection between left hand to left forearm). Although trees allow efficient and exact inference on graphical models, they suffer from "double-counting" phenomena (left limbs are confused with their right parts of the body). This problem is usually addressed by augmenting the graph with additional symmetry constraints, as appearance [85] or connections between symmetrical parts [47, 93]. Although efficient and optimal inference algorithms exist [93], loopy graphs are usually slow to optimize [47, 93] and the final solution is not exact. An alternative approach to handle "double-counting" phenomena is the combination of small HOG filters [107] with higher part detectors (e.g. poselets [13]) modeling groups of non adjacent body parts [78, 79, 42]. However, these methods still rely on loosely constrained kinematic models, allowing non-anthropomorphic detections.

Although successful, state-of-the-art 2-D models typically require a large amount of training data across views to achieve view-invariance. In preliminary results [76], we showed that unbiased 2-D models learned from 3-D data outperform those trained from 2-D data, also on human pose estimation datasets. In contrast, we propose a method that takes advantage of state-of-the-art body part detectors, but adding correlation among body parts by modeling 3-D body poses and viewpoints. Our approach is similar in spirit to [91], since they fit a 3-D statistical model to body part detections in the image. However, they rely on a first 2-D pose estimation from [106]. In [90] they overcame this limitation; nevertheless, they set additional strong constraints such as the use of a calibrated camera and a coarse initialization to speed up the process.

In order to reconstruct body configurations from different viewpoints, in this chapter we reformulate the human pose estimation problem as a subspace matching [86, 59] between image pixels and 2-D deformable models trained on 3-D MoCap data. As we show in the experimental section, our method

outperforms state-of-the-art approaches on Leeds Sports dataset [48] (LSP) because it is able to handle large viewpoint variations. In addition, our method is robust to large amounts of outliers, and we efficiently solved the subspace matching problem with linear programming.

5.2 Subspace Matching

This section describes the proposed subspace matching algorithm to estimate the human pose in images, given the unbiased 2-D model computed in Chapter 4. Human pose estimation refers to finding the body configuration in images, usually after estimating the likelihood of image pixels to belong to each body part [73, 107, 78, 79]. When body configurations are described by means of a subspace model, we can represent human pose estimation as a subspace matching problem [86, 59], between a 2-D deformable model of joints' variation and a pool of features or pixel candidates for each body joint, resulting of running state-of-the-art body part detectors.

The goal of feature selection by subspace matching is to determine the subset of ℓ landmarks from n_f candidate image features or landmarks that minimize the distance to a subspace model. It was first introduced by Roig et al. [86] to establish correspondences between a sparse set of d-dimensional image features $\mathbf{Q} \in \mathbb{R}^{d \times n_f}$ and a previously learned model of frontal faces. Given the candidate features and a model composed of a reference shape $\mathbf{M} \in \mathbb{R}^{d \times \ell}$ and k bases $\mathbf{B} \in \mathbb{R}^{d\ell \times k}$, the problem consisted on finding the optimal correspondence $\mathbf{S} \in \{0,1\}^{\ell \times n_f}$ and the subspace coefficients $\mathbf{c} \in \mathbb{R}^{k \times 1}$ which minimize the following error:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{c}} \left\| \mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - \boldsymbol{\mu} - \mathbf{B}\mathbf{c} \right\|_2^2, \\ &\text{s.t.} \quad \mathbf{S}\mathbf{1}_{n_f} = \mathbf{1}_\ell, \end{aligned} \tag{5.1}$$

where $\boldsymbol{\mu} = \mathrm{vec}(\mathbf{M}) \in \mathbb{R}^{d\ell \times 1}$ is the vectorization of the mean. The linear constraint enforces to select only one candidate for each landmark. To reduce the number of parameters, $\mathbf{c}$ is replaced by its optimal value $\mathbf{c} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - \boldsymbol{\mu})$ and the solution for $\mathbf{S} \in \{0,1\}^{\ell \times n_f}$ is found by means of Quadratic Programming (QP). Although novel, this formulation presents three main drawbacks: (1) QP is computationally expensive and the solution is found by combining the error of two QP problems, one for the shape (location of the pixels in the image, $d = 2$), and another one for the appearance (SIFT description [57] of the image at those locations, $d = 128$); (2) only frontal and centered objects (faces) are modeled; and (3) deformation parameters $\mathbf{c}$ are not restricted to be plausible values [18].

Feature selection has also been studied in the topic of graph matching. In [56], they introduced a matching method based on a locally affine-invariant geometric constraint and Linear Programming (LP) techniques. This work was extended in [112], making the method more robust to non-rigid facial poses contained in the training set, and additional constraints were considered to reduce the search space.

In this chapter, we build on [86], but solving the above mentioned drawbacks: (1) we reformulate the joint shape and appearance minimization as a single LP problem [56] instead of two QP problems, making feasible to handle the large number of candidate features of human pose estimation problems ($n_f \geq 2 \cdot 10^4$); (2) we add an affinity transformation $\mathbf{A} \in \mathbb{R}^{2\times 2}$ to model non-frontal objects, and a translation $\mathbf{t} \in \mathbb{R}^{2\times 1}$ to compensate for not being centered; and (3) we introduce constraints on the subspace parameters to guide the optimization to plausible values of deformation.

Moreover, we borrow landmark-candidate association formulation and constraints (see Fig. 5.1) from graph matching literature [112]. From now on, $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2\times n_f}$ denotes the set of 2-D candidate image pixels, where $\mathbf{Q}^t \in \mathbb{R}^{2\times n_t}$ is the subset of candidates of the t^{th} landmark and $n_f = \sum_{t=1}^{\ell} = n_t$. Each subset of candidates results from applying the state-of-the-art body part detector [107] for the corresponding joint. Hence, each of the n_f candidates is known to be associated with one of the ℓ landmarks and have an assignation cost, depending on the detector response. The landmark-candidate relation is encoded in the binary matrix $\mathbf{G} \in \{0,1\}^{\ell\times n_f}$, where $g_{ti} = 1$ if the i^{th} candidate belongs to the t^{th} landmark. In the same way, the assignation cost h_{ti} of choosing the i^{th} candidate as the t^{th} landmark is computed from a detector score by an efficient two-pass dynamic programming inference [73] and encoded in the matrix $\mathbf{H} \in \mathbb{R}^{\ell\times n_f}$ (see Fig. 5.1).

Given the candidate features, association constraints and cost ($\mathbf{Q}$, $\mathbf{G}$, $\mathbf{H}$), and the shape model (mean $\mathbf{M} \in \mathbb{R}^{2\times\ell}$, $\mathbf{B} \in \mathbb{R}^{2\ell\times k}$), the problem consists on finding the optimal correspondence $\mathbf{S}$, the affinity transformation $\mathbf{A}$, the translation $\mathbf{t}$ and the deformation weights $\mathbf{c}$ that minimize the following error:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{A},\mathbf{c},\mathbf{t}} \quad & \eta\,\mathrm{tr}(\mathbf{H}\mathbf{S}^T) + \left\|\mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - (\mathbf{I}_\ell \otimes \mathbf{A})\boldsymbol{\mu} - \mathbf{B}\mathbf{c} - (\mathbf{1}_\ell \otimes \mathbf{t})\right\|_1, \qquad (5.2) \\ \text{s.t.} \quad & \mathbf{S}\mathbf{1}_{n_f} = \mathbf{1}_\ell, \quad \text{with } \mathbf{S} \in \{0,1\}^{\ell\times n_f} \\ & s_{ti} = 0, \quad \text{when } g_{ti} = 0, \\ & -3\sqrt{\boldsymbol{\lambda}_j} \leq \mathbf{c}_j \leq 3\sqrt{\boldsymbol{\lambda}_j}, \quad j = 1 \ldots k, \end{aligned}$$

where the first term in the objective function measures the assignation cost, and the second one the self reconstruction error. η is a parameter to trade off between the two terms. In the experiments, we always set the value to $\eta = 100$ and we found the final result was not sensitive to small changes in

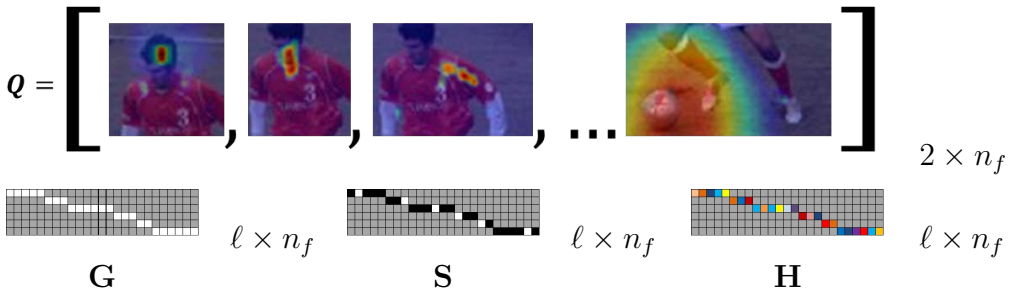


Figure 5.1: Illustration of the candidate features matrix $\mathbf{Q}$, as the concatenation of the detector responses for each body joint. More specifically, $\mathbf{Q}$ concatenates those pixel locations $\mathbf{Q}^t$ with high detection score after applying each t^{th} joint's filter. Association matrix $\mathbf{G}$ is illustrated by a sparse matrix, only having ones in those positions of each t^{th} row that correspond with $\mathbf{Q}^t$ candidates. Similarly, $\mathbf{H}$ provides an association cost for each possible selection. $\mathbf{S}$ shows an example of feature selection matrix, satisfying $\mathbf{G}$ restrictions and $\mathbf{H}$ cost.

this weight. Note that, instead of using l_2 norm, the reconstruction error is defined in l_1 norm because of its efficiency and robustness. Similarly to Eq. (5.1), the first constraint enforces $\mathbf{S}$ to select only one candidate for each landmark. However, the second constraint only allows $\mathbf{S}$ to select candidates for the t^{th} landmark from the corresponding set of candidates $\mathbf{Q}^t$ defined in $\mathbf{G}$. Finally, the third constraint imposes the subspace parameters to be plausible deformation values, where $\boldsymbol{\lambda} \in \mathbb{R}^{k\times 1}$ is a column vector containing the first k eigenvalues of the covariance matrix, of the training data.

Optimizing Eq. (5.2) is, however, NP-hard because of the integer constraints on $\mathbf{S}$. As in [56, 112], we approximate the problem with a continuous constraint, $\mathbf{S} \in [0,1]^{\ell\times n_f}$, and reformulate the objective function in order to avoid the non-smoothness of l_1 and apply LP:

$$\min_{\mathbf{S},\mathbf{A},\mathbf{c},\mathbf{t},\mathbf{u},\mathbf{v}} \quad \eta\,\mathrm{tr}(\mathbf{H}\mathbf{S}^T) + \mathbf{1}_{2\ell}^T(\mathbf{u}+\mathbf{v}), \tag{5.3}$$

$$\text{s.t.} \quad \mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - (\mathbf{I}_\ell \otimes \mathbf{A})\boldsymbol{\mu} - \mathbf{B}\mathbf{c} - (\mathbf{1}_\ell \otimes \mathbf{t}) = \mathbf{u}-\mathbf{v}, \mathbf{u} \geq \mathbf{0}_{2\ell}, \mathbf{v} \geq \mathbf{0}_{2\ell} \tag{5.4}$$

$$\mathbf{S} \in [0,1]^{\ell\times n_f}$$

$$s_{ti} = 0, \quad \text{when } g_{ti} = 0,$$

$$-3\sqrt{\boldsymbol{\lambda}_j} \leq \mathbf{c}_j \leq 3\sqrt{\boldsymbol{\lambda}_j}, \quad j = 1\ldots k,$$

where the two auxiliary variables $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{2\ell}$ replace l_1 norm with a smooth term, and the linear constraint defined in Eq. (5.4). Finally, we gradually discretize $\mathbf{S}$, after solving the LP, by taking successive refinements based on trust-region shrinking [46]. Note that several elements in $\mathbf{S}$ will be zero

during the optimization process (illustrated in gray in Fig. 5.1). we simplify the optimization task by removing those elements (i.e. $[t, i] \in \{[t, i]|g_{ti} = 0\}$), reducing the number of variables and the LP cost from $O(\ell n_f)$ to $O(n_f)$.

5.3 Experiments

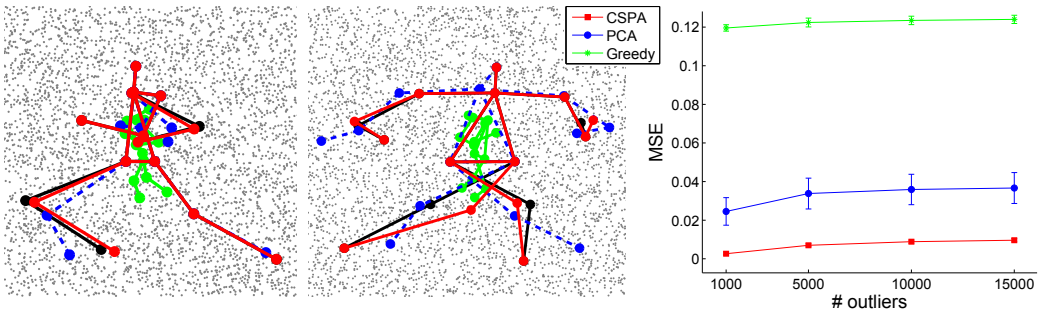
This section compares our unbiased 2-D models and the subspace matching method against state-of-the-art algorithms, in the problem of human pose estimation. We performed synthetic experiments on the CMU MoCap dataset (detailed in Section 2.4.2), and real experiments on the Leeds Sports (LSP) [48] dataset. For all experiments in this section we used the continuous version of our 2-D models, CSPA (see Section 4.3), trained with a set of 14 body joints, as is common across several databases for human pose estimation.

CMU MoCap dataset

The aim of this experiment is to show the performance of our feature selection method by subspace matching in the problem of human pose estimation, as a function of the number of outliers in the image. This synthetic experiment compares our method against two baselines on the CMU MoCap dataset: a greedy approach not restricting the feature selection by a shape model, and a method restricting the shape as [86]. Since this model is composed by a mean and a PCA of the data, we refer to this model as PCA. Recall that we introduced an affinity transformation to the feature selection formulation, which allows us to use a CSPA model in our approach. We refer to our method as CSPA. Also note that we are using our own implementation of [86] optimized in l_1 norm, since it was infeasible to perform this experiment with the original implementation in adequate computational time (we add 100 times more features candidates and double of the number of landmarks in our experiment).

For training we randomly selected 3 sequences, each one with 30 frames, from the set of 11 running sequences of the user number 9. For testing we randomly selected 2 sequences with 30 frames from the same set, and we rotated 30 times each 3-D shape in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$, as the training domain. For each projected 2-D skeleton we synthetically added $1 \sim 15000$ random outliers in the frame of the image, uniformly distributed per each joint. See Fig. 5.2 (a) for examples of random feature candidates.

We built the candidates matrix $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2 \times n_f}$ by concatenat-



(a) (b)

Figure 5.2: Results on CMU MoCap dataset. (*a*) *CSPA* model (*solid red lines*), *PCA* model (*dashed blue lines*), and *Greedy* (*green solid lines*) reconstructions over ground truth (*solid black lines*) and 5000 outliers (*grey dots*); and (*b*) MSE for each method as a function of the number of outliers.

ing the pixel locations $\mathbf{Q}^t \in \mathbb{R}^{2\times n_t}$ of the candidates features of each t^{th} landmark. The association cost of each candidate in matrix $\mathbf{H}$ is the Euclidean distance between the candidate feature and the ground truth landmark location plus a random noise. We report the MSE relative to the torso size, varying the number of candidates for three methods.

Fig. 5.2 (b) shows the mean reconstruction error and the standard deviation for the 100 realizations. As expected, methods restricting the search with a shape model have better performance than the greedy approach. Moreover, observe that our approach using the CSPA model outperforms the one using just a PCA model. This is due to the addition to the affinity transformation, as well as the limits on the deformation parameters in the feature selection formulation. Fig. 5.2 (a) shows two examples of the user number 9 of CMU MoCap dataset from two different viewpoints. Qualitative results also show that our method achieves a better fitting by means of a selection method robust to outliers. The execution times with 15000 outliers, on a 2.2GHz computer with 8Gb of RAM, were 0.72 sec. (PCA) and 0.68 sec. (CSPA) per image.

Leeds Sport Dataset

In this experiment, we tested the performance of our unbiased 2-D models, in combination to the proposed subspace matching method, to detect humans on Leeds Sports (LSP) dataset. LSP contains 2000 images of people performing different sports, some of them including extreme viewpoints. We performed the comparison in the test set of 1000 images. We trained our 2-D CSPA model in the CMU MoCap dataset [1] using 1000 frames. From

Table 5.1: Comparison of human pose estimation approaches on LSP dataset. Errors in pixels are provided for each body joint (left and right joints are averaged), as well as the mean estimated error for the 14 joints.

Method	**Head**	**Neck**	**Shoulder**	**Elbow**	**Wrist**	**Hip**	**Knee**	**Ankle**	**Mean**
YR [107]	21.75	18.97	20.54	31.27	49.03	22.78	27.24	38.42	29.95
Greedy	22.48	**18.41**	20.73	32.81	48.58	23.41	27.36	40.04	30.48
CSPA	**21.58**	18.48	**19.83**	**29.39**	**43.69**	**21.97**	**26.28**	**37.02**	**28.32**

the 2605 sequences of the motion capture data, we randomly selected 1000 and the frame in the middle of sequence is selected as representative frame. Using this training data, we built the 2-D CSPA model using the following ranges for the pitch, roll and yaw angles: $\phi, \theta, \psi \in [-3/4\pi, 3/4\pi]$. We built the candidates matrix $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2 \times n_f}$ by concatenating the pixel locations $\mathbf{Q}^t \in \mathbb{R}^{2 \times 1000}$ of the 1000 candidates pixels with higher response of each t^{th} joint, where the association cost of each candidate in matrix $\mathbf{H}$ is obtained from the detector score [107]. We normalized the response of all the pixel candidates from each landmark. We will refer to this model as CSPA. To evaluate the performance, we compared our approach with the state-of-the-art pose estimation method proposed by Yang and Ramanan[1] [107]. The error for each method is computed as the pixel distance between the estimated and ground-truth part locations.

Table 5.1 compares the error for each body joint of our method against that in [107], and a greedy approach. Our method improves the accuracy of all estimated joints compared to the baselines, and only the *Neck* estimation of the greedy approach is better. In order to find the global fitting of joints, CSPA method placed *Neck* landmarks according to the training shapes, even though it was selecting *Neck* landmarks with higher cost. Part of this error is due to different anatomical labeling between LSP dataset and the training set of our CSPA model, CMU MoCap dataset. Qualitative results in Fig. 5.3 show that our approach has similar results to the state-of-the-art, but being more accurate in the estimation of the limb lengths.

The execution time per image of our feature selection method, on a 2.2GHz computer with 8Gb of RAM, was 6.84 sec. The most computationally intensive part of the method is calculating the response for each image using [107], which is shared with all compared methods.

[1]The code was downloaded from author's website and adapted to our own code (http://www.ics.uci.edu/~dramanan/).

Figure 5.3: Qualitative results on LSP dataset. Left image from each pair of images shows the result from *YR* [107], and the right image shows our full approach using the *CSPA* model. Note how the *CSPA* leads to a more precise fitting of the body joints and more accurate limb lengths from different viewpoints.

5.4 Conclusions

In this dissertation we proposed to learn multi-view 2-D models from 3-D MoCap datasets, instead of learning them from 2-D images. In previous chapters we extended Procrustes Analysis (PA) to build multi-view 2-D models by rotating and projecting 3-D data samples. By means of CSPA (see Chapter 4) we modeled rigid and non-rigid deformations of 3-D MoCap sequences, in an efficient manner. Finally, in this chapter we illustrated the benefits of using multi-view 2-D models in the task of human pose estimation.

We first reformulated the human pose estimation problem as a feature selection by subspace matching, and introduced an efficient feature selection method to this end. Our proposed approach is much more efficient than the state-of-the-art of feature selection by subspace matching and is able to handle larger number of outliers. In experimental section, we showed the benefits of our method in a synthetic experiment on CMU MoCap dataset.

Finally, we evaluated our multi-view 2-D deformable models in the task of human pose estimation. CSPA models trained with motion capture data, combined with our subspace matching method, outperformed human pose estimation state-of-the-art approaches on the LSP dataset. Our method provides similar results to the state-of-the-art, but being more accurate in the joint positions and limb lengths. This is because our unbiased 2-D models

can successfully reconstruct different viewpoints, and the proposed feature matching method is able to handle large amounts of outliers.

Therefore, in this dissertation we provided the formalization and the tools for building multi-view 2-D shape models from 3-D data and we successfully illustrated their usability in the task of human pose estimation.

Chapter 6

Summary and Conclusions

The main contributions of this thesis are briefly summarized in this chapter, followed by an outline of future research direction.

6.1 Summary and Contributions

Human perception allows us to set physical restrictions, such as define faces and human skeletons as sets of anatomical landmarks or articulated bodies. However, the high variation of facial expressions and human postures from different viewpoints makes problems like human pose estimation or facial landmark localization extremely challenging. The common approach to handle large viewpoint variations is to train the models with several labeled images from different viewpoints [37, 99, 113, 107, 78, 79, 42]. However this approach has some drawbacks: (1) it is not clear the extent to which the dataset must be enhanced with images from different viewpoints in order to build unbiased 2-D models; (2) extending the training set without this evaluation would unnecessarily increase memory and computation requirements to train the models; (3) obtaining new labeled images from different viewpoints can be a difficult task because of the expensive labeling cost; and finally, (4) a non-uniform coverage of the different viewpoints of a person leads to biased 2-D models. In this dissertation we proposed successive extensions of the well-known Procrustes Analysis (PA) algorithm to address these issues.

First of all, we proposed Projected Procrustes Analysis (PPA) in **Chapter 2** as a formalization for building multi-view 2-D rigid models by rotating 3-D datasets. PPA rotates and projects every 3-D training shape and builds a multi-view 2-D model from this enhanced training set. However, PPA does not guarante unbiased 2-D models by itself, since it depends on how rotations are chosen. Uniformly distributed rotations will generate unbiased models,

while non-uniform rotations will lead to models not able to reconstruct properly particular viewpoints. Therefore, different rotation parametrizations are presented as well as mechanisms to uniformly sample the rotation space and build unbiased 2-D models. In experimental section we showed that unbiased 2-D models are able to generalize better to different viewpoints with smaller number of rotations in both faces and bodies datasets. Although large deformations of the CMU MoCap dataset make more evident the improvements of the uniform sampling in skeletons than in faces experiments, we encourage the use of uniform sampling of the rotation space in any dataset. In addition, PPA provides the basis of formulation and the understanding of the problem needed to develop the extensions presented in the following chapters.

Although successful in building multi-view 2-D models, the enhanced dataset required by PPA increased the computational requirements in space and time. To address this PA and PPA drawbacks and build unbiased 2-D models in an efficient manner, we proposed Continuous Procrustes Analysis (CPA) in **Chapter 3**. CPA extends PA and PPA within a functional analysis framework and builds multi-view 2-D rigid models in an efficient way, by means of integrating among all possible rotations in a given domain. CPA models are unbiased because we use the Haar measure in the definition of the integral. In experimental section we compared CPA models against the state-of-the-art PA methods and PPA. We compared all methods in faces and bodies datasets, increasing the number of rotations in the training set. As the number of projections increased, discrete methods converged to CPA, which provided a lower bound for the error in all experiments. Moreover, CPA was much more efficient in space and time.

After formalizing the construction of multi-view 2-D rigid models from 3-D data in a discrete (PPA) and a continuous (CPA) way, we showed the benefits in efficiency of the continuous approach. However, CPA efficiency was limited to rigid models, and building standard statistical models based on CPA would still require generating an enhanced dataset with rotations and projections of the 3-D samples in the training set. In **Chapter 4** we went an step further and proposed Subspace Procrustes Analysis (SPA) to efficiently compute multi-view 2-D deformable models. We added a subspace in the PA formulation that is able to model non-rigid deformations, as well as rigid 3-D transformations of the training set. We introduced a discrete (DSPA) and continuous (CSPA) formulation in order to provide a better understanding of the problem, where DSPA samples the 3-D rotation space, and CSPA integrates over $SO(3)$. Experiments comparing 2-D SPA models of faces' and joints' variations showed improvements w.r.t. state-of-the-art PA methods. Moreover, as the number of rotations in the training set increased DPSA converged to CSPA, but CSPA was much more efficient in space and

time.

Note that in this dissertation we proposed multi-view 2-D models trained on 3-D data sets, which is a relatively unexplored problem in computer vision. We rotated and projected 3-D training samples instead of learning a 3-D model and projecting it onto 2-D, in test time. We also reported experiments in faces and skeletons datasets, comparing multi-view 2-D models against standard 3-D models, projected onto a 2-D test set. We showed that our multi-view 2-D models were as expressive as 3-D models, but SPA models being faster in test time. Therefore, in this dissertation we proposed several extensions of PA to learn multi-view 2-D models, being efficient in test time.

Finally, in **Chapter 5** we illustrated the benefits of our multi-view 2-D deformable models in the task of human pose estimation. We first reformulated the problem as a feature selection by subspace matching and we proposed an efficient approach for this task. Our proposed method is much more efficient than the state-of-the-art feature selection by subspace matching approaches and it is able to handle larger number of outliers. In experimental section, our multi-view 2-D deformable models, combined with the subspace matching method, outperformed the state of the art of human pose estimation on the LSP dataset. Our method provided similar results to the current state of the art, but being more accurate in the joint positions and limb lengths. This is because our unbiased 2-D models can successfully reconstruct different viewpoints, and the proposed feature matching method is able to handle large amounts of outliers.

In this dissertation we proposed successive extensions of PA to build multi-view 2-D models for human perception, and solve the main challenges of modeling 3-D deformable objects (e.g. faces, bodies) from different viewpoints. In sum, our models are not biased to any particular viewpoint, and they are efficient in learning, as well as in test time.

6.2 Future Directions

This thesis creates some clear directions for future lines of research. In this dissertation we detailed our contributions to **build unbiased models** in terms of their generalization to different viewpoints, and we extended PA to build not only unbiased rigid models, but also deformable models. However, in the standard method PA aligns the data with respect to the mean and independently computes the PCA subspace. Although successful and widely extended, independently optimize the mean, geometric transformations and the subspace can result in loss of optimality [102]. Our future research will focus on solving the simultaneous alignment of the 2-D projections of 3-D

samples, while computing the 2-D subspace that can represent all possible projections of the 3-D samples under different camera views.

During this dissertation we argued that PA extensions presented in this thesis deal with missing data naturally. Since they use the whole 3-D shape of objects, the enhanced 2-D dataset resulting of projecting the data from different viewpoints can be constructed without occluded landmarks. Our future work will further the research in this direction by taking advantage of the 3-D structure, and we will model the behavior of occluded landmarks in test time. We will **model the likelihood of landmarks to be occluded**, depending of the rotation angles. The use of this information in the test phase would lead to a faster test time fitting, even with occluded face or body parts.

Our future research will include an in-depth study about the relation between rigid transformations in the training set and the subspace being spanned by the first bases of the model. We will also examine how small changes in the deformation parameters are mapped as small rigid and non-rigid deformations in the reconstructed shapes. This study will be extremely useful in extending our **feature selection by subspace matching method to video**, and how to restrict the optimization of the subspace parameters in the temporal domain.

During this research we found it useful to **build multi-view 2-D models directly from 3-D models**. Since unbiased 2-D and 3-D models have the same reconstruction power and the 2-D models are faster, this methodology would allow faster real-time applications for such domains where only 3-D models are available but not 3-D data (e.g. NRSFM model built from 2-D data). We outline the usefulness of this methodology in Section 4.5, and we provide the formulation and preliminary results as a proof of concept. Our future research will include an in-depth study and evaluation of this preliminary work, as well as a comparison against our multi-view 2-D models presented in this dissertation.

Appendix A

CPA Formulation

In this Appendix, we detail the derivation and optimization steps of the CPA method introduced in (Eq. (3.7)):

$$E_{\mathrm{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} F(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) d\boldsymbol{\omega} = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \|\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}(\boldsymbol{\omega})_i \mathbf{M}\|_F^2 \, d\boldsymbol{\omega}.$$

In order to minimize the CPA functional:

$$\min_{\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_1, \ldots, \mathbf{A}(\boldsymbol{\omega})_n} E_{\mathrm{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i), \tag{A.1}$$

we propose an algorithm based on the closed-form solution of two optimization subproblems. Unlike standard PA, in the present formulation, $\mathbf{A}(\boldsymbol{\omega})_i : \boldsymbol{\Omega} \to \mathbb{R}^{2\times 2}$ are functions and not parameters. Moreover, it is worth noticing that the dependence of E_{CPA} on the functions $\mathbf{A}(\boldsymbol{\omega})_i$ is non-linear. This makes the minimization of E_{CPA}, Eq. (A.1), a non-linear variational problem. Although the existence of a solution $(\mathbf{M}^*, \mathbf{A}(\boldsymbol{\omega})_1^*, \ldots, \mathbf{A}(\boldsymbol{\omega})_n^*)$ to the problem in Eq. (A.1) is guaranteed from a theoretical point of view, it is not straightforward to find its explicit expression (see Section 3.2.1). For this reason, we propose the following minimization algorithm to find a stationary point. First, we set an initial value $\mathbf{M} = \mathbf{M}^0$ and we optimize over the functions $\mathbf{A}(\boldsymbol{\omega})_1, \ldots, \mathbf{A}(\boldsymbol{\omega})_n$, obtaining a close solution for $[\mathbf{A}(\boldsymbol{\omega})_1^*, \ldots, \mathbf{A}(\boldsymbol{\omega})_n^*]$. In the next step, we minimize over $\mathbf{M}$ the functional $\mathbf{M} \to E_{\mathrm{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_1^*, \ldots, \mathbf{A}(\boldsymbol{\omega})_n^*)$, and we iterate until convergence. This two step algorithm is detailed below:

Step 1: Optimizing E_{CPA} over the functions $\mathbf{A}(\boldsymbol{\omega})_i$, i.e.:

$$\min_{\mathbf{A}(\boldsymbol{\omega})_i} E_{\mathrm{CGPA}}(\mathbf{M}^*, \mathbf{A}(\boldsymbol{\omega})_i), \tag{A.2}$$

can be solved using the following equation: $\nabla_{\mathbf{A}(\boldsymbol{\omega})_i} F(\mathbf{M}^*, \mathbf{A}(\boldsymbol{\omega})_i) = \mathbf{0}$, where $\nabla_{\mathbf{A}(\boldsymbol{\omega})_i}$ is the gradient operator with respect to the unknown parameters of the matrix $\mathbf{A}(\boldsymbol{\omega})_i$. First, let us rewrite $F(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i)$ with the following equivalent expression:

$$F(\mathbf{M}, \mathbf{A}_i) = \operatorname{tr}\left[(\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i)^T(\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i)\right] + \operatorname{tr}\left[(\mathbf{A}(\boldsymbol{\omega})_i\mathbf{M})^T(\mathbf{A}(\boldsymbol{\omega})_i\mathbf{M})\right] - 2\operatorname{tr}\left[(\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i)^T\mathbf{A}(\boldsymbol{\omega})_i\mathbf{M}\right].$$

Then:

$$\nabla_{\mathbf{A}(\boldsymbol{\omega})_i} F(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = 2\mathbf{A}(\boldsymbol{\omega})_i\mathbf{M}\mathbf{M}^T - 2\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{M}^T = \mathbf{0}.$$

Finally, the solution of these equations is:

$$\mathbf{A}(\boldsymbol{\omega})_i = \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{M}^{*T}(\mathbf{M}^*\mathbf{M}^{*T})^{-1} \qquad \forall i. \tag{A.3}$$

Step 2: To optimize E_{CPA} over $\mathbf{M}$, i.e.:

$$\min_{\mathbf{M}} E_{\text{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i^*), \tag{A.4}$$

the necessary conditions are: $\nabla_{\mathbf{M}} E_{\text{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i^*) = \mathbf{0}$. First, let us rewrite $E_{CPA}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i)$ with the following equivalent expression:

$$E_{CPA}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = \operatorname{tr}\left[\sum_{i=1}^{n}\mathbf{D}_i^T + \left(\int_{\boldsymbol{\Omega}}\mathbf{P}(\boldsymbol{\omega})^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)\mathbf{D}_i\right] + \operatorname{tr}\left[\mathbf{M}^T\left(\sum_{i=1}^{n}\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{A}(\boldsymbol{\omega})_i d\boldsymbol{\omega}\right)\mathbf{M}\right] - 2\operatorname{tr}\left[\sum_{i=1}^{n}\mathbf{D}_i^T\left(\int_{\boldsymbol{\Omega}}\mathbf{P}(\boldsymbol{\omega})^T\mathbf{A}(\boldsymbol{\omega})_i d\boldsymbol{\omega}\right)\mathbf{M}\right].$$

Then:

$$\nabla_{\mathbf{M}} E_{CPA}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = 2\left(\sum_{i=1}^{n}\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{A}(\boldsymbol{\omega})_i d\boldsymbol{\omega}\right)\mathbf{M} - 2\sum_{i=1}^{n}\left(\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)\mathbf{D}_i = \mathbf{0}.$$

Finally, the solution of these equations is (Eq. (3.10) in the main text):

$$\mathbf{M} = \mathbf{K}^{-1}\mathbf{Z} \tag{A.5}$$

where:

$$\mathbf{K} = \sum_{i=1}^{n}\int_{\boldsymbol{\Omega}}^{*}\mathbf{A}(\boldsymbol{\omega})_i^{*T}\mathbf{A}(\boldsymbol{\omega})_i^* d\boldsymbol{\omega} \tag{A.6}$$

$$\mathbf{Z} = \sum_{i=1}^{n}\left(\int_{\boldsymbol{\Omega}}\mathbf{A}(\boldsymbol{\omega})_i^{*T}\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)\mathbf{D}_i. \tag{A.7}$$

Note that we can replace the expression of $\mathbf{A}(\boldsymbol{\omega})_i^*$ (Eq. (A.3)) in $\mathbf{M}$ and, given an initial value $\mathbf{M} = \mathbf{M}^0$, we can iterate on Eq. (A.5) until convergence. As we show on the remaining of this appendix, fixed point optimization leads to an efficient formulation, since it allows to compute the definite integral off-line.

Fixed point minimization: Replacing the expression of the optimal $\mathbf{A}(\boldsymbol{\omega})_i^*$ from Eq. (A.3) in both Eq. (A.6) and Eq. (A.7), we find[1]:

$$\mathbf{Z} = \sum_{i=1}^{n} \left(\int_{\boldsymbol{\Omega}} \left(\mathbf{P}(\boldsymbol{\omega}) \mathbf{D}_i \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} \right)^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \right) \mathbf{D}_i =$$

$$(\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M} \left(\sum_{i=1}^{n} \mathbf{D}_i^T \underbrace{\left(\int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \right)}_{\mathbf{X}} \mathbf{D}_i \right) =$$

$$(\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M} \left(\sum_{i=1}^{n} \mathbf{D}_i^T \mathbf{X} \mathbf{D}_i \right) =$$

$$(\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M} \left(\sum_{i=1}^{n} (\mathbf{D}_i^T \otimes \mathbf{D}_i^T) \operatorname{vec}(\mathbf{X}) \right)^{(\ell)},$$

$$\mathbf{K} = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left(\mathbf{P}(\boldsymbol{\omega}) \mathbf{D}_i \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} \right)^T \left(\mathbf{P}(\boldsymbol{\omega}) \mathbf{D}_i \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} \right) d\boldsymbol{\omega} =$$

$$(\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M} \left(\sum_{i=1}^{n} \mathbf{D}_i^T \underbrace{\left(\int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \right)}_{\mathbf{X}} \mathbf{D}_i \right) \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} =$$

$$(\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M} \left(\sum_{i=1}^{n} \mathbf{D}_i^T \mathbf{X} \mathbf{D}_i \right) \mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1} = \mathbf{Z}\mathbf{M}^T (\mathbf{M}\mathbf{M}^T)^{-1},$$

where $\mathbf{X} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances. In order to compute the value of these integrals we only need to solve the definite integral for $\mathbf{X}$. Since $\mathbf{X}$ is not data dependent, we can compute the definite integral off-line for any given interval, leading to an extremely efficient optimization. For instance, considering, $\boldsymbol{\Omega} = \{(\phi, \theta, \psi) \in \mathbb{R}^3; |\phi| \leq$

[1]See Section 1.4 for an explanation of the vec-transpose operator.

$\pi/2, |\theta| \leq \pi/2, |\psi| \leq \pi/2\}$, we obtain:

$$\int_{\mathbf{\Omega}} \mathbf{X} d\boldsymbol{\omega} = \begin{pmatrix} \frac{\pi^2}{8} + \frac{\pi^3}{16} & 0 & 0 \\ 0 & \frac{\pi}{8} + \frac{3\pi^3}{32} & 0 \\ & 0 & \frac{-\pi}{8} + \frac{\pi^2}{8} + \frac{3\pi^3}{32} \end{pmatrix}.$$

Avoiding singularities: Finally, note that the special orthogonal group of rotations in 3-D space, $SO(3)$, is smooth except for a polar coordinate singularity along an angle of zero [88]. In order to avoid Euler singularities, we use Fubini's Theorem. We divide the domains containing zero into disconnected intervals, and we compute the joint integral as iterated integrals. For instance, in order to compute the integral in the domain defined above, i.e.:

$$\int_{\mathbf{\Omega}} \mathbf{X} d\boldsymbol{\omega} = \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \mathbf{X} d\boldsymbol{\omega},$$

we compute the integrals:

$$\mathbf{J}^1 = \int_{-\pi/2}^{0} \mathbf{X} d\phi + \int_{0}^{\pi/2} \mathbf{X} d\phi$$

$$\mathbf{J}^2 = \int_{-\pi/2}^{0} h \sin(\theta) \mathbf{J}^1 d\theta + \int_{0}^{\pi/2} h \sin(\theta) \mathbf{J}^1 d\theta,$$

where h is the Haar measure for the Euler angle interval. And finally:

$$\int_{\mathbf{\Omega}} \mathbf{X} d\boldsymbol{\omega} = \int_{-\pi/2}^{0} \mathbf{J}^2 d\psi + \int_{0}^{\pi/2} \mathbf{J}^2 d\psi$$

Appendix B

CSPA Formulation

In this Appendix, we detail the steps from Eq. (4.5) to Eq. (4.8), as well as the definition of the covariance matrix, introduced in Section 4.

Given the value of $\mathbf{M}^*$ and the optimal expression of $\mathbf{A}(\boldsymbol{\omega})_i^*$ from Eq. (3.8), we substitute them in Eq. (4.5) resulting in:

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{H} - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega}, \tag{B.1}$$

where $\mathbf{H} = \mathbf{M}^{*T}(\mathbf{M}^*\mathbf{M}^{*T})^{-1}\mathbf{M}^*$ and $\mathbf{D}_i \in \mathbb{R}^{3\times\ell}$. Then,

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i(\mathbf{I}_\ell - \mathbf{H}) - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} \tag{B.2}$$

leads us to Eq. (4.7) and Eq. (4.8), where $\bar{\mathbf{D}}_i = \mathbf{D}_i(\mathbf{I}_\ell - \mathbf{H})$ and $\bar{\mathbf{d}}_i = \text{vec}(\bar{\mathbf{D}}_i)$. From Eq. (4.8), solving $\nabla_{\mathbf{c}(\boldsymbol{\omega})_i} E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \mathbf{0}$ we find:

$$\mathbf{c}(\boldsymbol{\omega})_i^* = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i. \tag{B.3}$$

The substitution of $\mathbf{c}(\boldsymbol{\omega})_i^*$ in Eq. (4.8) results in:

$$E_{\text{CSPA}}(\mathbf{B}) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{B.4}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{B.5}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \text{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \left((\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i\right)^T \right] d\boldsymbol{\omega} = \tag{B.6}$$

$$\text{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)\boldsymbol{\Sigma} \right], \tag{B.7}$$

where:

$$\boldsymbol{\Sigma} = \int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega}. \tag{B.8}$$

We can find the global optima of Eq. (B.7) by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Sigma}$ is the covariance matrix and $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$. However, the definite integral in $\boldsymbol{\Sigma}$ is data dependent. To be able to compute the integral off-line, we need to rearrange the elements in $\boldsymbol{\Sigma}$. Using vectorization and vec-transpose operator[1]:

$$\boldsymbol{\Sigma} = (\mathrm{vec}\,[\boldsymbol{\Sigma}])^{(2\ell)} = \tag{B.9}$$

$$\left(\mathrm{vec} \left[\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega} \right] \right)^{(2\ell)} = \tag{B.10}$$

$$\left(\left(\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) d\boldsymbol{\omega} \right) \mathrm{vec} \left[\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right] \right)^{(2\ell)}, \tag{B.11}$$

which finally leads to:

$$\boldsymbol{\Sigma} = \left((\mathbf{I}_\ell \otimes \mathbf{Y}) \,\mathrm{vec} \left[\sum_{i=1}^{n} \bar{\mathbf{d}}_{ij} \bar{\mathbf{d}}_{ij}^T \right] \right)^{(2\ell)}, \tag{B.12}$$

where the definite integral $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega})) d\boldsymbol{\omega} \in \mathbb{R}^{4\ell \times 9\ell}$ can be computed off-line.

[1] See Section 1.4 for the vec-transpose operator.

Bibliography

[1] Carnegie mellon motion capture database. http://mocap.cs.cmu.edu

[2] Google 3d warehouse. https://3dwarehouse.sketchup.com/

[3] Grabcad. http://grabcad.com/

[4] Turbosquid. http://www.turbosquid.com/

[5] Ali, A., Aggarwal, J.: Segmentation and recognition of continuous human activity. In: IEEE Workshop on Detection and recognition of events in video. pp. 28–35 (2001)

[6] Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1014–1021 (2009)

[7] Arvo, J.: Graphics gems II, vol. 2. Morgan Kaufmann (1991)

[8] Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. II–432 (2003)

[9] Baker, S., Matthews, I., Schneider, J.: Automatic construction of active appearance models as an image coding problem. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). vol. 26, pp. 1380–1384 (2004)

[10] Bartoli, A., Pizarro, D., Loog, M.: Stratified generalized procrustes analysis. International Journal of Computer Vision (IJCV) 101(2), 227–253 (2013)

[11] Bautista, M.A., Hernández-Vela, A., Ponce, V., Perez-Sala, X., Baró, X., Pujol, O., Angulo, C., Escalera, S.: Probability-based dynamic time warping for gesture recognition on rgb-d data. In: International

Workshop on Depth Image Analysis (WDIA). pp. 126–135. Springer (2012)

[12] Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: ACM SIGGRAPH. pp. 187–194 (1999)

[13] Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: European Conference on Computer Vision (ECCV), pp. 168–181. Springer (2010)

[14] Box, G.E.P., Muller, M.E.: A note on the generation of random normal deviates. The Annals of Mathematical Statistics 29(2), 610–611 (June 1958)

[15] Brand, M.: Morphable 3d models from video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. II–456 (2001)

[16] Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: a 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics PP(99), 1–1 (2013)

[17] Cootes, T.F., Edwards, G.J., Taylor, C.J., et al.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 23(6), 681–685 (2001)

[18] Cootes, T.F., Taylor, C.J.: Statistical models of appearance for computer vision (2004)

[19] Dacorogna, B.: Direct methods in the calculus of variations. Springer-Verlag (1989)

[20] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1, pp. 886–893 (2005)

[21] De la Torre, F.: A least-squares framework for component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(6), 1041–1055 (2012)

[22] Diaconis, P., Shahshahani, M.: The subgroup algorithm for generating uniform random variables. Probability in the Engineering and Informational Sciences 1(01), 15–32 (1987)

[23] Dryden, I.L., Mardia, K.V.: Statistical shape analysis, vol. 4. John Wiley & Sons New York (1998)

[24] Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding (CVIU) 108(1), 52–73 (2007)

[25] Escalera, S., Baró, X., Gonzalez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: ECCV ChaLearn Workshop on Looking at People (2014)

[26] Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 617–624 (2011)

[27] Faraway, J.J.: Regression analysis for a functional response. Technometrics 39(3), 254–261 (1997)

[28] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision (IJCV) 61(1), 55–79 (2005)

[29] Fishman, G.S.: Monte Carlo: Concepts, algorithms, and applications. Springer Series in Operations Research, Springer-Verlag, New York, NY, USA (1996)

[30] Fonseca, I., Leoni, G.: Modern Methods in the Calculus of Variations: Lp̂ Spaces. Springer (2007)

[31] Franco, A., Maio, D., Maltoni, D.: 2d face recognition based on supervised subspace learning from 3d models. Pattern Recognition 41(12), 3822–3833 (2008)

[32] Freedman, D.A.: Statistical models: theory and practice. Cambridge University Press (2005)

[33] Frey, B.J., Jojic, N.: Transformation-invariant clustering using the em algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 25(1), 1–17 (2003)

[34] Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding (CVIU) 73(1), 82–98 (1999)

[35] Goodall, C.: Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society. Series B (Methodological) pp. 285–339 (1991)

[36] Gower, J.C., Dijksterhuis, G.B.: Procrustes problems, vol. 3. Oxford University Press Oxford (2004)

[37] Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing 28(5), 807–813 (2010)

[38] Hamilton, S.W.R.: Lectures on quaternions. Hodges and Smith, Dublin, Ireland (1853)

[39] Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)

[40] Hernández-Vela, A., Bautista, M.A., Perez-Sala, X., Ponce, V., Baró, X., Pujol, O., Angulo, C., Escalera, S.: Bovdw: Bag-of-visual-and-depth-words for gesture recognition. In: International Conference on Pattern Recognition (ICPR). pp. 449–452 (2012)

[41] Hernández-Vela, A., Bautista, M.Á., Perez-Sala, X., Ponce-López, V., Escalera, S., Baró, X., Pujol, O., Angulo, C.: Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d. Pattern Recognition Letters (2013)

[42] Hernàndez-Vela, A., Sclaroff, S., Escalera, S.: Contextual rescoring for human pose estimation. In: BMVC (2014)

[43] Igual, L., De la Torre, F.: Continuous procrustes analysis to learn 2d shape models from 3d objects. In: NORDIA workshops, in conjunction with CVPR (2010)

[44] Igual, L., Perez-Sala, X., Escalera, S., Angulo, C., De la Torre, F.: Continuous generalized procrustes analysis. Pattern Recognition 47(2), 659–671 (2014)

[45] James, G.M., Hastie, T.J.: Functional linear discriminant analysis for irregularly sampled curves. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63(3), 533–550 (2001)

[46] Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29(6), 959–975 (2007)

[47] Jiang, H., Martin, D.R.: Global pose estimation using non-tree models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)

[48] Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: Proceedings of the British Machine Vision Conference (2010)

[49] Jones, M.J., Poggio, T.: Multidimensional morphable models. In: IEEE International Conference on Computer Vision (ICCV). pp. 683–688 (1998)

[50] Junejo, I.N., Dexter, E., Laptev, I., Perez, P.: View-independent action recognition from temporal self-similarities. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 33(1), 172–185 (2011)

[51] Kirk, D.: Graphics Gems III (IBM Version): Ibm Version. Elsevier (1994)

[52] Kokkinos, I., Yuille, A.: Unsupervised learning of object deformation models. In: IEEE International Conference on Computer Vision (ICCV). pp. 1–8 (2007)

[53] Kuffner, J.J.: Effective sampling and distance metrics for 3d rigid body path planning. In: ICRA. vol. 4, pp. 3993–3998 (2004)

[54] Learned-Miller, E.G.: Data driven image models through continuous joint alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28(2), 236–250 (2006)

[55] Levin, A., Shashua, A.: Principal component analysis over continuous subspaces and intersection of half-spaces. In: European Conference on Computer Vision (ECCV). pp. 635–650. Springer (2002)

[56] Li, H., Huang, X., He, L.: Object matching using a locally affine invariant and linear programming techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35(2), 411–424 (2013)

[57] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) 60(2), 91–110 (2004)

[58] Marimont, D.H., Wandell, B.A.: Linear models of surface and illuminant spectra. JOSA A 9(11), 1905–1913 (1992)

[59] Marques, M., Stosić, M., Costeira, J.: Subspace matching: Unique solution to point matching with geometric constraints. In: IEEE International Conference on Computer Vision (ICCV). pp. 1288–1294 (2009)

[60] Marr, D.: Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. Inc., New York, NY pp. 2–46 (1982)

[61] Matthews, I., Xiao, J., Baker, S.: 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. International Journal of Computer Vision (IJCV) 75(1), 93–113 (2007)

[62] Minka, T.P.: Old and new matrix algebra useful for statistics. http://research.microsoft.com/ minka/papers/matrix/, 2000

[63] Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding (CVIU) 104(2), 90–126 (2006)

[64] Moeslund, T.B., Hilton, A., Krüger, V., Sigal, L.: Visual Analysis of Humans. Springer (2011)

[65] Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on pure and applied mathematics 42(5), 577–685 (1989)

[66] Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31(4), 607–626 (2009)

[67] Muybridge, E.: The human figure in motion. Courier Dover Publications (1955)

[68] Naimark, M.A.: Linear representatives of the Lorentz group (translated from Russian). New York, Macmillan (1964)

[69] Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC. vol. 1, p. 3 (2011)

[70] Olshen, R.A., Biden, E.N., Wyatt, M.P., Sutherland, D.H.: Gait analysis and the bootstrap. The annals of statistics pp. 1419–1440 (1989)

[71] Ormoneit, D., Black, M.J., Hastie, T., Kjellström, H.: Representing cyclic human motion using functional analysis. Image and Vision Computing 23(14), 1264–1276 (2005)

[72] Osher, S., Paragios, N.: Geometric level set methods in imaging, vision, and graphics. Springer (2003)

[73] Park, D., Ramanan, D.: N-best maximal decoders for part models. In: IEEE International Conference on Computer Vision (ICCV). pp. 2627–2634 (2011)

[74] Pearson, K.: On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11), 559–572 (1901)

[75] Pentland, A.P.: The visual inference of shape: computation from local features. Ph.D. thesis, Massachusetts Institute of Technology (1982)

[76] Perez-Sala, X., De la Torre, F., Igual, L., Escalera, S., Angulo, C.: Subspace procrustes analysis. In: ECCV Workshop on ChaLearn Looking at People (2014)

[77] Perez-Sala, X., Igual, L., Escalera, S., Angulo, C.: Robotic Vision: Technologies for Machine Learning and Vision Applications, chap. Uniform Sampling of Rotations for Discrete and Continuous Learning of 2D Shape Models, pp. 23–42. IGI Global, Hershey, PA, USA (2012)

[78] Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 588–595 (2013)

[79] Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong appearance and expressive spatial models for human pose estimation. In: IEEE International Conference on Computer Vision (ICCV). pp. 3487–3494 (2013)

[80] Pizarro, D., Bartoli, A.: Global optimization for optimal generalized procrustes analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2409–2415 (2011)

[81] Plaisier, J.R., Jiang, L., Abrahams, J.P.: Cyclops: New modular software suite for cryo-EM. Journal of Structural Biology 157(1), 19–27 (2007)

[82] Poppe, R.: Vision-based human motion analysis: An overview. Computer Vision and Image Understanding (CVIU) 108(1), 4–18 (2007)

[83] Poppe, R.: A survey on vision-based human action recognition. Image and vision computing 28(6), 976–990 (2010)

[84] Ramsay, J.O., Silverman, B.W.: Functional Data Analysis. Springer (1997)

[85] Ren, X., Berg, A.C., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: IEEE International Conference on Computer Vision (ICCV). vol. 1, pp. 824–831 (2005)

[86] Roig, G., Boix, X., De la Torre, F.: Optimal feature selection for subspace image matching. In: ICCV Workshops. pp. 200–205 (2009)

[87] Rui, Y., Anandan, P.: Segmenting visual actions based on spatio-temporal motion patterns. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1, pp. 111–118 (2000)

[88] Shoemake, K.: Animating rotation with quaternion curves. In: ACM SIGGRAPH. vol. 19, pp. 245–254 (1985)

[89] Shoemake, K.: Uniform random rotations. In: Kirk, D. (ed.) Graphics Gems III, pp. 124–132. Academic Press Professional, Inc., San Diego, CA, USA (1992), http://dl.acm.org/citation.cfm?id=130745.130769

[90] Simo-Serra, E., Quattoni, A., Torras, C., Moreno-Noguer, F.: A joint model for 2d and 3d pose estimation from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3634–3641 (2013)

[91] Simo-Serra, E., Ramisa, A., Alenya, G., Torras, C., Moreno-Noguer, F.: Single image 3d human pose estimation from noisy observations. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2673–2680 (2012)

[92] Spivak, M.: Calculus. Corrected. Cambridge University Press (2006)

[93] Tian, T.P., Sclaroff, S.: Fast globally optimal 2d human detection with loopy graph models. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 81–88 (2010)

[94] Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision (IJCV) 9(2), 137–154 (1992)

[95] De la Torre, F., Black, M.J.: Robust parameterized component analysis: theory and applications to 2d facial appearance models. Computer Vision and Image Understanding (CVIU) 91(1), 53–71 (2003)

[96] De la Torre, F., Nguyen, M.H.: Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8 (2008)

[97] Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 30(5), 878–892 (2008)

[98] Ullman, S., Basri, R.: Recognition by linear combinations of models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 13(10), 992–1006 (1991)

[99] Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H., Ma, Y.: Toward a practical face recognition system: Robust alignment and illumination by sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 34(2), 372–386 (2012)

[100] Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding (CVIU) 115(2), 224–241 (2011)

[101] Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. International Journal of Computer Vision (IJCV) 67(2), 233–246 (2006)

[102] Xiao, J., Georgescu, B., Zhou, X., Comaniciu, D., Kanade, T.: Simultaneous registration and modeling of deformable shapes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 2429–2436 (2006)

[103] Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 532–539 (2013)

[104] Xiong, X., la Torre, F.D.: Supervised descent method for solving nonlinear least squares problems in computer vision (2014)

[105] Yang, F., Shechtman, E., Wang, J., Bourdev, L., Metaxas, D.: Face morphing using 3d-aware appearance optimization. In: Graphics Interface. pp. 93–99. Canadian Information Processing Society (2012)

[106] Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1385–1392 (2011)

[107] Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 35(12), 2878–2890 (2013)

[108] Yershova, A., Jain, S., Lavalle, S.M., Mitchell, J.C.: Generating uniform incremental grids on SO(3) using the Hopf fibration. The Internation Journal of Robotics Research 29(7), 801–812 (June 2010)

[109] Yershova, A., LaValle, S.M.: Deterministic sampling methods for spheres and SO(3). In: ICRA. vol. 4, pp. 3974–3980. IEEE (2004)

[110] Yezzi, A.J., Soatto, S.: Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. International Journal of Computer Vision (IJCV) 53(2), 153–167 (2003)

[111] Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. Acm Computing Surveys (CSUR) 35(4), 399–458 (2003)

[112] Zhou, F., Brandt, J., Lin, Z.: Exemplar-based graph matching for robust facial landmark localization. In: IEEE International Conference on Computer Vision (ICCV). pp. 1025–1032 (2013)

[113] Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2879–2886 (2012)