Spectrum Analysis Methods for 3D Facial Expression Recognition and Head Pose Estimation

Dmytro Derkach

TESI DOCTORAL UPF / ANY 2018

DIRECTOR DE LA TESI
Federico M. Sukno Departament of Information and Communication Technologies

To my mom and dad

Acknowledgments

I would like to thank several people who have contributed to the accomplishment of this thesis.

First of all, I would like to thank my supervisor, Dr. Federico Sukno that he gave me an opportunity to do a PhD. He supported me across the full spectrum, from giving me a really interesting topic to continuous guidance and help in all aspects of my research. Thanks, for his tireless enthusiasm for my research and many hours that we spent discussing my work, helping me to solve the problems I met in my research. I would like to thank Dr. Constantine Butakoff, who also supported me during my PhD life in Barcelona.

Many thanks to all my colleagues in the CMTech: Adria, Oriol, Adriana, Celi, Decky without whom the time during my PhD study would be inconceivable. I would like to thank all whom I met during being here, who helped me and allowed me to learn new things every day. Finaly, I would like to thank all the administrative staff in the DTIC, especially Lydia and Luis.

Also I am very grateful to my family and friends for standing next to me and making me laugh during the difficult times.

Ця робота присвячується моїм батькам. Хочу подякувати їм за безцінну підтримку та турботу яку вони мені надавали протягом всього часу. Мамо і тату – все що я досягнув у своєму житті, це завдяки вам. Також хотів би подякувати свої дівчині Наталі, як весь час була поруч, підтримувала мене та завжди вірила в мене. Дуже вдячний всім друзям, які весь цей час були в Україні і не забували про мене, і тим яких я зустрів тут в Барселоні, а особливо Святославу, який був хорошим другом з перших днів перебування в Барселоні.

Abstract

Facial analysis has attracted considerable research efforts over the last decades, with a growing interest in improving the interaction and cooperation between people and computers. This makes it necessary that automatic systems are able to react to things such as the head movements of a user or his/her emotions. Further, this should be done accurately and in unconstrained environments, which highlights the need for algorithms that can take full advantage of 3D data. These systems could be useful in multiple domains such as human-computer interaction, tutoring, interviewing, health-care, marketing etc. In this thesis, we focus on two aspects of facial analysis: expression recognition and head pose estimation. In both cases, we specifically target the use of 3D data and present contributions that aim to identify meaningful representations of the facial geometry based on spectral decomposition methods:

1. We propose a spectral representation framework for facial expression recognition using exclusively 3D geometry, which allows a complete description of the underlying surface that can be further tuned to the desired level of detail. It is based on the decomposition of local surface patches in their spatial frequency components, much like a Fourier transform, which are related to intrinsic characteristics of the surface. We propose the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace. The proposed approach is tested in terms of expression and Action Unit recognition and results confirm that the proposed GLFs produce state-of-the-art recognition rates.

2. We propose an approach for head pose estimation that allows modeling the underlying manifold that results from general rotations in 3D. We start by building a fully-automatic system based on the combination of landmark detection and

dictionary-based features, which obtained the best results in the FG2017 Head Pose Estimation Challenge. Then, we use tensor representation and higher order singular value decomposition to separate the subspaces that correspond to each rotation factor and show that each of them has a clear structure that can be modeled with trigonometric functions. Such representation provides a deep understanding of data behavior, and can be used to further improve the estimation of the head pose angles.

Resum

Al llarg de les últimes dècades, l'anàlisi facial ha atret un interès creixent i considerable per part de la comunitat investigadora amb l'objectiu de millorar la interacció i la cooperació entre les persones i les màquines. Aquest interès ha propiciat la creació de sistemes automàtics capaços de reaccionar a diversos estímuls com ara els moviments del cap o les emocions d'una persona. Més enllà, les tasques automatitzades s'han de poder realitzar amb gran precisió dins d'entorns no controlats, fet que ressalta la necessitat d'algoritmes que aprofitin al màxim els avantatges que proporcionen les dades 3D. Aquests sistemes poden ser útils en molts àmbits com ara la interacció home-màquina, tutories, entrevistes, atenció sanitària, màrqueting, etc. En aquesta tesi, ens centrem en dos aspectes de l'anàlisi facial: el reconeixement d'expressions i l'estimació de l'orientació del cap. En ambdós casos, ens enfoquem en l'ús de dades 3D i presentem contribucions que tenen com a objectiu la identificació de representacions significatives de la geometria facial mitjançant mètodes basats en la descomposició espectral:

1. Proposem una tecnologia basada en la representació espectral per al reconeixement d'expressions facials utilitzant exclusivament la geometria 3D, la qual ens permet una descripció completa de la superfície subjacent que pot ser ajustada al nivell de detall desitjat. Dita tecnologia, es basa en la descomposició de fragments locals de la superfície en les seves components de freqüència espacial, d'una manera semblant a la transformada de Fourier, que estan relacionades amb característiques intrínseques de la superfície. Concretament, proposem la utilització de les Graph Laplacian Features (GLFs) que resulten de la projecció dels fragments locals de la superfície a una base comuna obtinguda a partir del Graph Laplacian eigenspace. El mètode proposat s'ha avaluat en termes de reconeixement d'expressions i Action

Units (activacions musculars facials), i els resultats obtinguts confirmen que els GLFs produeixen taxes de reconeixement comparables a l'estat de l'art.

2. Proposem un mètode per a l'estimació de l'orientació del cap que permet modelar el manifold subjacent que formen les rotacions generals en 3D. En primer lloc, construïm un sistema completament automàtic que combina la detecció de landmarks (punts facials rellevants) i característiques basades en diccionari, el qual ha obtingut els millors resultats al FG2017 Head Pose Estimation Challenge. Posteriorment, utilitzem una representació basada en tensors i la seva descomposició en els valors singulars d'ordre més alt per tal de separar els subespais de cada factor de rotació i mostrar que cada un d'ells té una estructura clara que pot ser modelada amb funcions trigonomètriques. Aquesta representació proporciona un coneixement detallat del comportament de les dades i pot ser utilitzada per millorar l'estimació de les orientacions dels angles del cap.

Summary

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Facial analysis

The face provides a large amount of information. Human-human interaction is accompanied not only by different gestures but also by gazing and different facial expressions. Just by seeing the faces of another person we can tell a lot about his or her feelings. On the other hand, most of the human-computer interaction is still performed using peripheral devices, like a keyboard, mouse, and/or a display. while automatic facial analysis is a promising tool to be used for more effective, versatile, and user-friendly human-computer interaction.

There are many aspects in which automatic face analysis can be used, and it is fundamental in many applications. For instance, a person can be identified from the face image and, further, facial expressions can be analyzed so that the computer can adapt to the user mood. Potential applications span a wide spectrum, ranging from security to entertainment, but the common thing among them is that they all try to equip computers with the ability to gain high-level understanding of the user by means of digital images of the face. There is currently intensive research on methods to extract high-level information from faces, and two important branches from it are our focus in this thesis – facial expression recognition (FER) and head

pose estimation.

1.1.1 Facial expression recognition

The human face contains important and rich visual information for expressing emotion. Facial analysis studies have been carried out over the past few decades for different purposes. The growing interest in improving the interaction and cooperation between people and computers makes it necessary that automatic systems are able to react to a user and his/her emotions, as it takes place in natural human intercourse. Facial expressions provide the cues of non-verbal communication by means of which we can interpret the mood, meaning and emotions at the same time. Due to that, FER could be applied to a wide range of situations. For example, automatic detection of expressions is essential when the user's attention is highly required, such as in surveillance and vehicle driving [Vural et al., 2007]. In marketing, facial analysis could be applied to analyze the reaction of consumers [McDuff et al., 2013b]. It, also, could be applied in tutoring system [Ammar et al., 2010]. For every distant learning environment, detecting a learner's emotional reaction could be a fundamental element, which would show us if the students are bored, interested or puzzled. In a clinical context, a doctor may monitor patients and be alerted when the patients are suffering, annoyed, depressed, or uncomfortable [Cohn et al., 2009, Lucey et al., 2011]. Many other applications such as virtual reality [Riva, 2006, Parsons et al., 2017], video-conferencing [Li et al., 2015c, Shih et al., 2017], user profiling [Arapakis et al., 2009] and customer satisfaction studies for broadcast and web services [McDuff et al., 2013a], require efficient FER in order to achieve the desired results [Girard et al., 2013]. Therefore, it is important to have accurate and robust expression classification to harness the information available in human expression.

Facial expressions recognition is a challenging problem as the face is capable of complex motions and the range of possible expressions

is wide [Sandbach et al., 2012b]. The psychologist Paul Ekman has made a lot of efforts to define facial expressions precisely. In his early work [Ekman and Friesen, 1971], there a cross-cultural study was performed on the existence of universal categories of emotional expressions. Based on that he suggested that there exist six basic human emotions: happiness, sadness, surprise, fear, anger and disgust (Fig. 1.1). Another relevant work of Ekman [Ekman et al., 1978] is that he developed the Facial Action Coding System (FACS) that has become the most well known system for describing facial expressions. It defines 46 facial action units (AUs) that are based on facial muscle movements, and all facial expressions can be defined using this system. Fig 1.2 illustrates some of the Action Units described in FACS.

Currently, these the above studies are used as a guide the research on facial expression recognition. People, in their works, focused on the classification of the six basic emotions, which have been assumed to be universal, or try to detect the facial muscle movements corresponding to AUs.

A general review of FER (including also 2D) can be found in the recent work by Corneanu et al. [Corneanu et al., 2016]. For surveys of earlier approaches, we can refer to to [Sandbach et al., 2012c, Fang et al., 2011a] for 3D and [Zeng et al., 2009, Pantic and Rothkrantz, 2000] for 2D domains.

We see that many researchers have been focused on the facial expression recognition, but most of them have been working on the

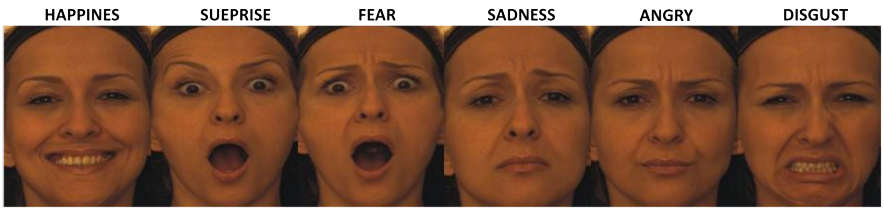

Figure 1.1: The six universal facial expressions.

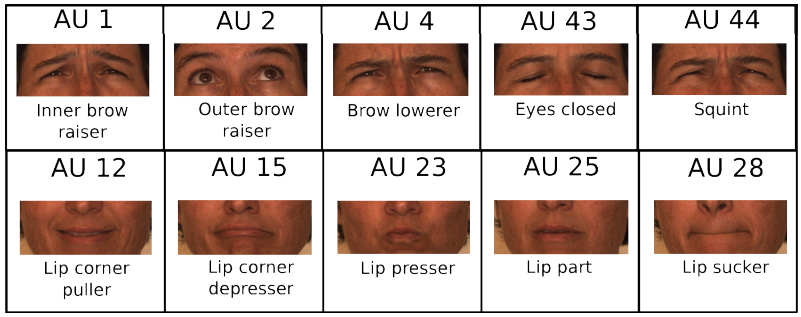

Figure 1.2: Examples of Action Units described in FACS. The samples are extracted from the Bosphorus 3D facial expression database [Savran et al., 2008]

2D domain i.e. using texture information [Sandbach et al., 2012c]. Despite their achievements, facial analysis methods based on still suffer from illumination and pose variations, which often occur in real conditions. The illumination problem is basically the variability of an object's appearance from one image to the next with slight changes in lighting conditions and viewpoints [Vishwakarma et al., 2007]. Due to this problem, it is difficult to handle subtle facial behavior in 2D domain.

With the rapid development of 3D imaging and scanning technologies, 3D data has appeared as a promising solution in face processing and analysis. The main reason for this is that 3D face scans contain detailed geometric shape information, without suffering from the problems of illumination and pose variations that are inherent to the 2D faces. Another important reason to use 3D data was presented by Savran et al. [Savran et al., 2012]. They conducted a comparative evaluation of 3D and 2D face modalities, and demonstrated that overall 3D data performs better, especially for lower face AUs.

Thus, this explains the recent increase in the number of studies dealing with 3D face information, and also attracted our interest to this field.

1.1.2 Head pose estimation

Another important branch in facial analysis for human-computer interaction is head pose estimation. Based on the fundamental assumption that head pose is highly correlated with the direction of visual gaze [Stiefelhagen et al., 1999], this issue addresses the following question – "where does the person look?". Despite, a person's gaze direction is strictly related to his/her eyes, physiological investigations reveal that gaze direction is strongly correlated by the orientation of a human head, i.e., head pose [Langton et al., 2004].

Based on the aforementioned findings, one of the applications can be a hands-free interface. For instance, knowledge of a person's head pose may directly control a device designed for disabled people, or act as a replacement of the mouse in human-computer interaction [Mateo et al., 2008]. It can be applied in human behavior understanding, such as analyzing inclination of passers-by to an outdoor advertisement or monitoring drivers' attention [Smith et al., 2008, Murphy-Chutorian et al., 2007]. Also head pose can be used to understand the real world in augmented reality [Wang and Yang, 2017]. Further, head pose could be useful together with facial expression analysis is social behavior modeling. Apart from verbal communication, analyzing head pose and facial expression of people is a good way to understand the interaction between them [Ba and Odobez, 2009]. In this case a head pose estimation system can be used, also, as a pre-processing step for pose independent face recognition [Blanz and Vetter, 2003], facial motion analysis [Wang et al., 2018] or stress indication [Giannakakis et al., 2018].

The goal of head pose estimation is to predict the relative orientation between the camera and a target head. This orientation is usually represented by three angles: rotation around vertical axis (yaw angle), around lateral axis (pitch angle), and around longitudinal axis (roll angle) as illustrated in Fig. 1.3.

Traditionally, head pose estimation has been performed on 2D images, but advances in 3D acquisition systems have led to a growing

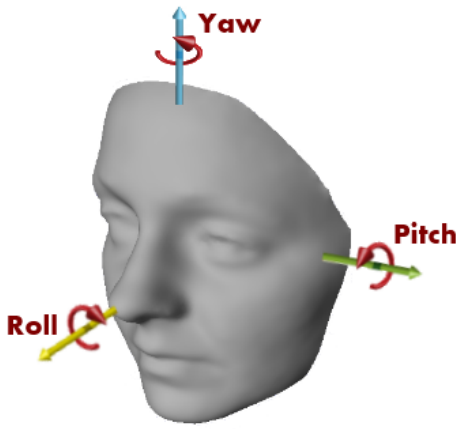

Figure 1.3: Orientation of the head in terms of pitch, roll, and yaw angles.

interest in methods that operate on 3D data [Seemann et al., 2004]. As with 3D facial expression analysis, these methods are less sensitive to changes in illumination which makes them more accurate and robust.

1.2 Contribution

The work carried out during my PhD is focused on two facial research areas: 3D facial expression recognition and 3D head pose estimation. And in both of these works we have used approaches based on spectral analysis.

Spectral analysis is one of several statistical techniques useful for characterizing and analyzing data. This analysis refers to the decomposition of data into oscillations of different lengths or scales. By this process, the observations in what is called the data domain are converted into the spectral domain [Smelser et al., 2001]. The reasons for doing this are that: (a) some forms of manipulation are easier in the spectral domain; and (b) the revealed scales are necessary

statistical descriptors of the data and may suggest important factors that affect or produce such data.

Spectral analysis or Spectrum analysis is the analysis in terms of a spectrum of frequencies or related quantities such as energies, eigenvalues, singular values etc. In mathematics, the spectrum of a matrix is the set of its eigenvalues [Golub et al., 1996] or singular values, which could be treated as the square roots of the eigenvalues. Thus, roughly, we can say that our work is based on the analysis of eigenvalues and singular values. In the next subsections, we briefly explain why and how this was done.

1.2.1 3D facial expression recognition

As mentioned before, 3D shape analysis has attracted increasing attention, although the availability of 3D information is not always fully exploited, i.e. 3D/depth information is treated analogously to a gray-scale image and the 3D information is simply extracted by directly applying 2D techniques. In order to take full advantage of depth information we need approaches that are truly 3D.

In this work, we investigate the problem of Facial Expression Recognition (FER) using 3D data and present an approach for automatic 3D facial expression recognition. Building from one of the most successful frameworks for facial analysis using exclusively 3D geometry, we extend the analysis from a curve-based representation into a spectral representation, which allows a complete description of the underlying surface that can be further tuned to the desired level of detail. Spectral representations are based on the decomposition of the geometry in its spatial frequency components, much like a Fourier transform, which are related to intrinsic characteristics of the surface. We propose the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace. The proposed approach is tested on the three most popular databases for 3D FER (BU-3DFE, Bosphorus and BU-4DFE) in terms of expression

and AU recognition. Our results show that the proposed GLFs consistently outperform the curves-based approach as well as the most popular alternative for spectral representation, Shape-DNA, which is based on the Laplace Beltrami Operator and cannot provide a stable basis to guarantee that the extracted signatures for the different patches are directly comparable. Furthermore, we demonstrate that the approach can work in a fully-automatic setting by integrating a state-of-the-art 3D landmark detector to guide the extraction of the features with no manual intervention, while still maintaining high expression recognition rates.

1.2.2 3D head pose estimation

For 3D head pose estimation, we also use a kind of spectral analysis, which is based on the analysis of elements obtained from tensor decomposition. This decomposition splits a tensor into one small core tensor and a set of matrices which consist of singular values. Thus we can treat it as one of the areas of spectrum analysis.

The head pose orientation is usually represented following three Euler angles:

- Yaw angle: rotation around the vertical (y) axis.
- Pitch angle: rotation around the horizontal side-to-side (x) axis.
- Roll angle: rotation around the horizontal back-to-front (z) axis.

Despite the fact that standard features used to represent 3D meshes lie in high-dimensional spaces, a key observation to solve this problem is that the aforementioned angles define a lower-dimensional manifold with only three degrees of freedom. This fact makes tensor decomposition and manifold learning appealing frameworks for the estimation of the orientation parameters. In particular, multi-linear decomposition are able to separate the variations produced by the different factors (i.e. angles) into separate subspaces, thus obtaining

specific parametrizations for each of them. On the other hand, manifold learning can be used to find the low-dimensional manifold structure defined by the orientation angles.

Based on the aforementioned findings, we propose an approach to learn the manifold, defined by 3D rotation, based on the coefficients obtained from tensor decomposition. For this purpose, we use multi-linear decomposition over 3D descriptors in order to split the pose variation factors (i.e. yaw, pitch and roll) and obtain a set of subspaces whose coefficients are governed by an unique parameter. These coefficients define a continuous curve in each of the sub-spaces that corresponds to the head pose variation along one of the rotation angles. We further show that these curves can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. We show that the proposed framework can achieve state-of-the-art performance for head pose estimation.

1.3 Outline of the thesis

The thesis is organized into 4 chapters. Chapters 2-4 are self-contained and each of them corresponds to a published or under review paper, while Chapter 5 summarizes the conclusions from this work.

Chapter 2. This chapter presents an approach for automatic 3D facial expression recognition. This approach based on the local shape spectrum representation, which allows a complete description of the underlying surface. In this chapter, we propose the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace. Experiments are carried out on three publicly available databases in terms of expression and Action Unit recognition.

Chapter 3. In this chapter we present a system that is able to estimate head pose using only depth information from

consumer RGB-D cameras such as Kinect 2. In contrast to most approaches addressing this problem, we do not rely on tracking and produce pose estimation in terms of pitch, yaw and roll angles using single depth frames as input. Our system combines three different methods for pose estimation: two of them are based on state-of-the-art landmark detection and the third one is a dictionary-based approach that is able to work in especially challenging scans where landmarks or mesh correspondences are too difficult to obtain. We evaluated our system on the SASE database, which consists of ∼30K frames from 50 subjects. We obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge.

Chapter 4. In this chapter, we also present an algorithm for 3D head pose estimation using only depth information from RGBD consumer cameras. We present an approach that allows modeling the underlying 3D manifold that results from rotation variations. To do so, we use tensor representation and higher order singular value decomposition to generate separate subspaces for each variation factor and show that each of them has a clear structure that can be modeled with cosine functions from a unique shared parameter per angle. Such representation provides a deep understanding of data behavior and angle estimations can be performed by optimizing combinations of these cosine functions. We evaluate our approach on two publicly available databases, and achieve top state-of-the-art performance.

Chapter 5. Finally, in this chapter we summarize this thesis by giving the most important ideas and contributions of the work in both 3D facial expression recognition and head pose estimation.

1.4 Publications

The research developed during this thesis has resulted in the following list of publications:

Journals

1. **D. Derkach**, A. Ruiz, F.M. Sukno, "Tensor Decomposition and Non-linear Manifold Modeling for 3D Head Pose Estimation", *International Journal of Computer Vision, (Under Review)*

2. **D. Derkach**, F.M. Sukno, "Automatic Local Shape Spectrum Analysis for 3D Facial Expression Recognition", *Image and Vision Computing, special issue "Best of FG2017" (Accepted)*, DOI: 10.1016/j.imavis.2018.09.007

International Conferences

1. **D. Derkach**, A. Ruiz, F.M. Sukno, "3D Head Pose Estimation Using Tensor Decomposition and Non-linear Manifold Modeling", *International Conference on 3D Vision*, 2018, pages 505-513, *(Oral Presentation)*, DOI: 10.1109/3DV.2018.00064

2. **D. Derkach**, F.M. Sukno, "Local Shape Spectrum Analysis for 3D Facial Expression Recognition", *International Conference on Automatic Face and Gesture Recognition* (FG2017), 2017, pages 41-47, *(Oral Presentation)*, DOI: 10.1109/FG.2017.143

3. **D. Derkach**, A. Ruiz, F.M. Sukno, "Head Pose Estimation Based on 3-D Facial Landmarks Localization and Regression", *International Conference on Automatic Face and Gesture Recognition* (FG2017), 2017, pages 820-827, *(Oral Presentation)*, DOI: 10.1109/FG.2017.104, *(Winner of the Head Pose Estimation Challenge)*

Chapter 2

AUTOMATIC LOCAL SHAPE SPECTRUM ANALYSIS FOR 3D FACIAL EXPRESSION RECOGNITION

Adapted from : D. Derkach, F.M. Sukno, "Automatic Local Shape Spectrum Analysis for 3D Facial Expression Recognition", *Image and Vision Computing, special issue "Best of FG2017"*, DOI: 10.1016/j.imavis.2018.09.007

D. Derkach, F.M. Sukno. "Local shape spectrum analysis for 3D facial expression recognition". *In Automatic Face & Gesture Recognition (FG 2017)*, 2017 12th IEEE International Conference on (pp. 41-47). IEEE., DOI: 10.1109/FG.2017.143

Abstract

We investigate the problem of Facial Expression Recognition (FER) using 3D data. Building from one of the most successful frameworks for facial analysis using exclusively 3D geometry, we extend the analysis from a curve-based representation into a spectral representation, which allows a complete description of the underlying surface that can be further tuned to the desired level of detail. Spectral representations are based on the decomposition of the geometry in its spatial frequency components, much like a Fourier transform, which are related to intrinsic characteristics of the surface. In this Chapter, we propose the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace. We extract patches around facial landmarks and include a state-of-the-art localization algorithm to allow for fully-automatic operation. The proposed approach is tested on the three most popular databases for 3D FER (BU-3DFE, Bosphorus and BU-4DFE) in terms of expression and AU recognition. Our results show that the proposed GLFs consistently outperform the curves-based approach as well as the most popular alternative for spectral representation, Shape-DNA, which is based on the Laplace Beltrami Operator and cannot provide a stable basis that guarantee that the extracted signatures for the different patches are directly comparable. Interestingly, the accuracy improvement brought by GLFs is obtained also at a lower computational cost. Considering the extraction of patches as a common step between the three compared approaches, the curves-based framework requires a costly elastic deformation between corresponding curves (e.g. based on splines) and Shape-DNA requires computing an eigen-decomposition of every new patch to be analyzed. In contrast, GLFs only require the projection of the patch geometry into the Graph Laplacian eigenspace, which is common to all patches and can therefore be pre-computed off-line. We also show that 14 automatically detected landmarks are enough to achieve high FER and AU detection rates, only slightly below those obtained when using sets of manually annotated landmarks.

2.1 Introduction

The human face plays an important role in expressing emotions such as happiness, satisfaction, surprise, fear, sadness or disgust. While there is consensus about the need to integrate multi-modal information for a complete understanding of human emotions, facial expressions are considered one of the most relevant channels for humans to regulate interactions both with the environment and with other persons [Pantic, 2009].

During the past two decades, the problem of facial expression recognition (FER) has become very relevant. The growing interest in improving the interaction and cooperation between people and computers makes it necessary that automatic systems are able to react to a user and his emotions, as it takes place in natural human intercourse. Many applications such as virtual reality [Riva, 2006, Parsons et al., 2017], video-conferencing [Eisert and Girod, 1998, Li et al., 2015c, Shih et al., 2017], user profiling [Arapakis et al., 2009] and customer satisfaction studies for broadcast and web services [McDuff et al., 2013a], require efficient FER in order to achieve the desired results [Girard et al., 2013]. Therefore, the impact of facial expression analysis on the above-mentioned application areas is constantly growing.

Early works on FER have focused primarily on the 2D domain (texture information) [Sandbach et al., 2012c] due to the prevalence of data. Despite their great achievements, facial analysis methods based on still suffer from illumination and pose variations, which often occur in real conditions.

With the rapid development of 3D imaging and scanning technologies, it becomes more and more popular using 3D face scans. Compared with 2D face images, 3D face scans contain detailed geometric shape information of facial surfaces, which removes the problems of illumination and pose variations that are inherent to the 2D modality. Thus, 3D-shape analysis has attracted increasing attention, although the availability of 3D information is not always

fully exploited: in many cases, 3D information is analyzed by directly applying 2D techniques to limited depth representations, such as depth maps (2.5D), where the depth information is treated analogously to a gray-scale image and the 3D information is simply extracted by computing popular 2D texture descriptors like LBPs [Guo et al., 2009, Wang and Meng, 2013, Wang et al., 2014b], SIFT [Berretti et al., 2010] or Gabor filters [Yun and Guan, 2010, Xie et al., 2010, D'Hose et al., 2007]. More recently, deep convolutional neural networks have also been explored in order to generate deep features [Li et al., 2015b] from this 2.5D representation.

In order to take full advantage of depth information we need approaches that are truly 3D. A notable approach in this direction, from Maalej et al., is based on the representation of surfaces with a finite number of *level curves* [Maalej et al., 2011]. Based on this curves, authors emphasized the importance of using local regions instead of the entire face and proposed a local geometric analysis of the surface. Further, they applied a Riemannian framework to derive 3D shape analysis and quantify similarity between corresponding patches on different 3D facial scans.

Motivation and Contributions

Despite the success of the level-curves framework from [Maalej et al., 2011], it could be argued that it is an incomplete representation of the 3D data, since it only captures part of the underlying surface, which is actually sampled by means of a finite number of curves. In contrast, spectral representations are based on the decomposition of the complete geometry in its (fundamental) frequency components, which are related to intrinsic characteristics of the surface, and correspond to the eigenvectors of the Laplace Beltrami Operator (LBO). The spectrum of the LBO is an isometric invariant, and it has been shown to be a powerful descriptor as a signature for (non-rigid) 3D shape matching and classification [Karni and Gotsman, 2000, Reuter et al., 2006]. The most popular of such

descriptors, baptized as "Shape-DNA", was proposed by Reuter et al. [Reuter et al., 2006], by taking the eigenvalues (i.e. spectrum) of the LBO. Because such spectrum captures intrinsic shape information, it was shown that it can be used like a DNA-test to identify 3D objects or to detect similarities in practical applications. Applications of Shape-DNA include object identification for the purpose of copyright protection [Reuter et al., 2005], shape analysis for medical applications [Niethammer et al., 2007] or smoothing and partitioning of complex structures [Qiu et al., 2006, Qiu et al., 2008]. Shape-DNA has also been used for statistical shape analysis with different purposes [Reuter et al., 2009], but, to the best of our knowledge, it has not been applied for facial expression analysis.

Based on the facts above, in this Chapter we explore the use of spectral methods as local shape descriptors for 3D FER. We show that the application of Shape-DNA is not the best way to deal with local face patches and that a fixed-graph basis, which we refer to as Graph Laplacian Features (GLFs), provides superior results. This is theoretically sound given the impossibility to ensure a fixed ordering of the spectral components under the Shape-DNA approach [Jain et al., 2007]. Compared to the curves-based framework, the proposed method constitutes a generalization to a full representation of the surface patches resulting in higher accuracy and reduced computational complexity. Preliminary results of this approach were presented in [Derkach and Sukno, 2017].

We perform experiments over the three most widely used databases for 3D facial expression analysis: Bosphorus [Savran et al., 2008], BU-3DFE [Yin et al., 2006] and BU-4DFE [Yin et al., 2008]. In all cases we show that the proposed GLFs approach still outperform the curves-based and Shape-DNA alternatives. Furthermore, we demonstrate that the approach can work in a fully-automatic setting by integrating a state-of-the-art 3D landmark detector [Sukno et al., 2015] to guide the extraction of GLFs with no manual intervention, while still maintaining high expression recognition rates.

The remainder of this Chapter is organized as follows. Section

2.2 introduces the existing approaches for 3D FER. In Section 2.3 we provide the required background on spectral mesh processing and Section 2.4 details the proposed GLFs for 3D facial analysis. Automatic landmarks detection is covered in Section 2.5 while Sections 2.6 and 2.7 detail our experiments on FER and Action Units detection, respectively. Section 2.8 concludes the chapter.

2.2 Related Work

The use of the 3D data for facial analysis is not so wide as the use of 2D, but still, there are considerable efforts toward solving the problem of 3D FER. In this section we will briefly review them, highlighting those that are most related to our approach and focusing only on approaches employing 3D features. A general review of FER (including also 2D) can be found in the recent work by Corneanu et al. [Corneanu et al., 2016]. For surveys of earlier approaches, the reader is referred to [Sandbach et al., 2012c], [Fang et al., 2011a] for 3D and [Zeng et al., 2009, Pantic and Rothkrantz, 2000] for 2D.

One of the most popular feature to encode facial information in 3D is curvature. For example, in [Wang et al., 2013b], Wang et al. described facial scans based on four curvature-based descriptors, namely, the two principal curvatures, mean curvature and shape index. This information was gathered on a regular grid (like a 2.5D representation) and it was later encoded by Local Binary Patterns (LBP). A similar strategy was followed by Zeng et al. [Zeng et al., 2013] who conformally mapped the 3D facial surface onto a 2D unit disk and then considered it as a 2D image for further processing.

Vretos et al [Vretos et al., 2011] were focused on the 3D FER using Zenrike moments on depth images. It is a set of complex polynomials, which form an orthogonal set over the interior of the unit circle. The Zenrike moments were proposed, in [Khotanzad and Hong, 1990], in order to tackle several problems arising from the use of raw moments in image processing such as redundancy of the moments, as well as,

difficulty in the recovery of the image from these moments, due to high computational burden.

A group of works closely related to our approach are those that represent the 3D geometry by means of sets of curves [Klassen and Srivastava, 2006, Srivastava et al., 2011, Maalej et al., 2011, Samir et al., 2009, Drira et al., 2013, Amor et al., 2014]. These start with the seminal work presented by Klassen et al., based on the representation of static 3D images with a finite number of *level curves* [Klassen and Srivastava, 2006]. They showed that curves can be used to represent surface regions, being able to capture quite subtle deformations. Thus, 3D shape analysis can be performed by comparisons of corresponding level curves. It should be noted, however, that such comparison is not trivial, given that distances between 3D level curves should be computed based on the geodesic paths of their underlying manifold. An important step forward in this direction was presented in [Srivastava et al., 2011, Maalej et al., 2011], by introducing a square-root velocity representation for analyzing curves in Euclidean spaces under a Riemannian metric. In particular, they computed geodesic paths between curves under this metric to obtain deformations between closed curves. Samir et al. [Samir et al., 2009] applied a similar curves-based approach for the analysis of facial surfaces. They represented a surface as an indexed collection of closed curves. These curves were extracted according to to their Euclidean distance from the tip of the nose, which is sensitive to deformations and, thus, can better capture differences related to variant expressions. Then, authors studied curves' differential geometry and endowed it with a Riemannian metric. In order to quantify differences between any two facial surfaces, the length of a geodesic was used. A similar framework was used in [Drira et al., 2013, Amor et al., 2014] for analyzing 3D faces, with the goal of comparing, matching and averaging faces, with the difference that surfaces were represented by radial curves outflowing from the nose tip.

Another popular strategy to work with 3D facial scans has been

to exploit the availability of texture information. For example, Jan et al. [Jan and Meng, 2015] explored different feature extraction methods on both 2D texture images and 3D geometric data, and further fused the two domains to increase performance. The explored features include all pair-wise distances between landmarks (using the 83 manually provided points), Edge Oriented Histogram (EOH), LBP and Local Phase Quantization (LPQ).

Advances in 3D imaging devices have made it possible the use of 3D dynamic sequences (also known as 3D + time or 4D data). Such 3D sequences make it possible to analyze the behaviour of the facial geometry over time. Efforts in this direction have often focused on the design of features that can adequately capture both the spatial and temporal variations present in 4D data [Sandbach et al., 2012b, Canavan et al., 2012, Reale et al., 2013]. For example, Sandbach et al. [Sandbach et al., 2012b] proposed 3D motion-based features (the Free-Form Deformation algorithm) between frames of 3D facial geometry sequences, Canavan et al. [Canavan et al., 2012] presented a dynamic curvature descriptor (dynamic shape-index) constructed from local regions as well as temporal domains and Reale et al. [Reale et al., 2013] presented the 4D spatio-temporal "Nebula" descriptor, which is a histogram of different facial regions using geometric features (i.e. curvatures and polar angles), after fitting the volume data to a cubic polynomial.

A shortage of some of the approaches mentioned above is their need for manual intervention, most often in terms of landmark position that need to be known before the method can be applied. In contrast, there exist some methods that can work fully automatically [Yang et al., 2015, Azazi et al., 2015, Li et al., 2015a, Li et al., 2012]. For example, Yang et al. [Yang et al., 2015] presented a method that is able to work without the need for facial landmarks, as long as the input data is aligned and cropped beforehand (e.g. as in the BU-3DFE database). The method starts by extracting a set of maps of shape features in terms of multiple order differential quantities (e.g. Normal Maps and the Shape Index Maps) to describe geometry

attributes of the facial surface. Then a scattering operator was introduced to further highlight expression related cues on these maps, thereby constructing geometric scattering representations of 3D faces for classification.

Full-automation has also been addressed by incorporating landmark detection within FER systems. Azazi et al.[Azazi et al., 2015] proposed a fully automatic system that starts by transforming the 3D faces into the 2D plane using conformal mapping. Then a Differential Evolution optimization algorithm was used to simultaneously select the optimal facial feature set and the classifier parameters. The optimal features were selected from a pool of Speed Up Robust Features (SURF) descriptors of all the prospective facial points, which were automatically detected using 2D texture information. Li et al. [Li et al., 2015a] also used automatic 2D landmark detection and project the located landmarks back to the 3D scan so that both 2D and 3D features can be extracted. Their approach combines multi-order gradient-based local texture and shape descriptors. A local image descriptor based on histogram of second order gradients (HSOG) along with first order gradient based SIFT descriptors were used to describe the local texture around each 2D landmark. Similarly, the local geometry around each 3D landmark was described by two local shape descriptors constructed using first and second order surface differential geometry quantities, i.e. meshHOG, meshHOS. Classification was based on SVMs, reporting results of all 2D and 3D descriptors fused at both feature-level and score-level to further improve the accuracy.

Fully-automatic systems have also been proposed in 4D [Zhen et al., 2016, Xue et al., 2015, Berretti et al., 2013, Fang et al., 2011b]. Zhen et al. [Zhen et al., 2016] investigated 4D FER based on a muscular movement model. They firstly segment each 3D frame in 11 muscular regions using the Iterative Closest Normal Point algorithm and extract features that include coordinate, normal and shape index values. Classification is performed both in 3D and 4D domains by using SVMs and Hidden Markov Models (HMMs). A different

strategy was followed by Xue et al. [Xue et al., 2015] who start by detecting a large number of 2D landmarks that are then projected into the 3D surface to extract Discrete Cosine Transform (DCT) features. In order to model 3D dynamics, they upgrade these features to the spatio-temporal domain by analyzing sequences of depth maps arranged as volumes, where time is the 3rd dimension.

It is interesting to note that, except [Zeng et al., 2013] and [Zhen et al., 2016], most efforts to develop fully-automatic FER systems in 3D have so far relied on the detection of landmarks in 2D. In contrast, in this Chapter we present a system that works fully automatically based solely on the 3D geometry of the face (e.g. without using any texture information).

2.3 Spectral Shape Analysis

Spectral methods have been applied to solve a variety of problems including mesh compression, correspondence, smoothing, watermarking, segmentation, surface reconstruction etc. [Reuter, 2010, Nealen et al., 2006, Zhang et al., 2010].

Spectral shape analysis relies on the decomposition of the surface geometry into its spatial frequency components (spectrum). Such representation allows to analyze the surface by examining the eigenvalues, eigenvectors or eigenspace projections of these fundamental frequencies. One of the advantages of these methods is that they are invariant with respect to isometric embeddings of the shape and robust to pose variations such as translation and rotation.

In this chapter, we will use the spectrum of the Laplace operator as local descriptors of the facial surface for expression recognition. The Laplacians are the most commonly used operators for spectral mesh processing. More rigorously, the Laplacian can be considered a special case of the more general Laplace-Beltrami Operator (LBO), which is defined on a manifold invariant to its parameterization, taking into account only its Riemannian metric [Reuter et al., 2006,

Bronstein and Bronstein, 2011]. The spectrum of the LBO is an isometric invariant, and it has been shown to be strongly linked to the geometry of the surface and its intrinsic structure [Qiu et al., 2008]. Results from spectral theory suggest that LBO eigenvalues are tightly related to almost all major invariants [Chung, 1997], and it has also been observed that level sets of LBO eigenfunctions follow geometric features [Lévy, 2006], highlight protrusions [Reuter, 2010] and reveal (global) symmetry [Ovsjanikov et al., 2008].

Several Laplacian operators have been proposed in the literature to compute the mesh spectrum. In this chapter we are especially interested in the two most popular ones:

1. Graph Laplacian, related to operators that have been widely studied in graph theory [Chung, 1997]. Despite this operator is based solely upon topological information, its eigenfunctions (i.e. eigenvectors) generally have a remarkable conformity to the mesh geometry [Isenburg et al., 2001]. On the other hand, the eigenfunctions of this operator are sensitive to aspects such as mesh resolution or triangulation.

2. Discretizations of the LBO from Riemannian geometry [Chavel, 1984, Rosenberg, 1997], which try to obtain a basis that depends only on the underlying geometry and not on its specific representation. This is the type of operator used in the Shape-DNA approach (see Section 2.3.2).

In the next subsections we detail how the above operators are computed and how they can be used to encode geometric information of 3D meshes.

2.3.1 Graph Laplacian

Mesh (graph) Laplacian operators are linear operators that act on functions defined on the mesh and depend purely on the mesh points (vertices) and their connectivity (e.g. triangulation).

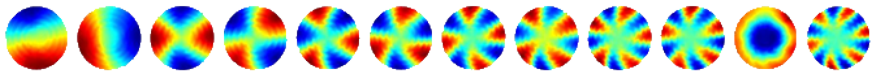

Figure 2.1: Example of the first 12 spatial patterns of the Graph laplacian eigenvectors of a disk mesh.

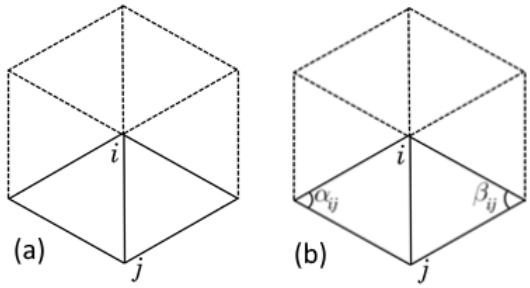

Figure 2.2: 1-ring neighbors (a) and angles opposite to an edge (b)

Given a mesh $\mathcal{M}$ with n vertices $\mathbf{V}$ and edges $\mathbf{E}$, $\mathcal{M} = (\mathbf{V}, \mathbf{E})$, the Graph Laplacian $\mathbf{L^G}(\mathcal{M})$ will be a $n \times n$ matrix defined as:

$$\mathbf{L^G}_{ij} = \begin{cases} -1 & if\ (i,j) \in \mathbf{E} \\ d_i & if\ i = j \\ 0 & otherwise \end{cases}$$

where d_i is the degree or valence of vertex i (Fig. 2.2(a)).

Since this operator is determined purely by the connectivity of the mesh, it does not explicitly encode geometric information. However, as shown in the seminal work from Taubin [Taubin, 1995], eigen-decomposition of the graph Laplacian produces an orthogonal basis whose components relate to spatial frequencies (Fig. 2.1), much like a Fourier Transform. Projections of a mesh into the eigenspace of graph Laplacian have been proposed and used to derive shape descriptors [Desbrun et al., 1999, Kim and Rossignac, 2005, Zahn and Roskies, 1972].

2.3.2 Shape-DNA

In Riemannian geometry, the Laplace operator can be generalized to operate on functions defined on a surface, resulting in the Laplace-Beltrami Operator (LBO).

Ovsjanikov in [Ovsjanikov et al., 2008] showed that the LBO can be defined entirely in terms of the metric tensor on the manifold, independently of the parameterization. Compared to the graph Laplacian, the LBO does not aim at operating on the mesh vertices, but rather on the underlying manifold itself. It depends continuously on the shape of the surface [Courant and Hilbert, 1965].

The Laplace operator based on the cotan formula represents the most popular discrete approximation to the LBO currently used for geometry processing [Meyer et al., 2003]. This operator can be presented as a product of a diagonal and a symmetric matrix $\mathbf{L^B} = \mathbf{B}^{-1}\mathbf{S}$, where $\mathbf{B}^{-1}$ is a diagonal matrix whose entries are Voronoi areas for all vertices and $\mathbf{S}$ is a symmetric matrix defined [Wang et al., 2012a]:

$$\mathbf{L^B}_{ij} = \begin{cases} -w_{ij} & if\ (i,j) \in \mathbf{E} \\ \sum_{k \in \mathbf{N}(i)} w_{ik} & if\ i = j \\ 0 & otherwise \end{cases}$$

where $w_{ij} = (\cot \alpha_{ij} + \cot \beta_{ij})$, α_{ij} and β_{ij} are the angles opposite if the edge (i, j) (see Fig. 2.2(b)), and $\mathbf{N}(i)$ is a set of vertices that are adjacent to vertex i.

A significant amount of geometric and topological information is known to be contained in the spectrum. Since the spectrum (i.e. the eigenvalues) of the LBO contains intrinsic shape information Reuter et al proposed to use them as shape signature or "Shape-DNA" [Reuter et al., 2006]. Shape-DNA can be used to identify shapes and detect similarities.

A disadvantage of the $\mathbf{L^B}$ operator with respect to the $\mathbf{L^G}$ is that the former is not symmetric. In order to extract appropriate eigenvalues, Laplacian matrices should be symmetric, so that they

possess real eigenvalues whose eigenvectors form an orthogonal basis [Bhatia, 2013]. However, although $\mathbf{L^B}$ itself is not symmetric in general, it is similar to the symmetric matrix $\mathbf{O} = \mathbf{B}^{-1/2}\mathbf{S}\mathbf{B}^{-1/2}$ since

$$\begin{aligned}\mathbf{L^B} &= \mathbf{B}^{-1}\mathbf{S} = \\ &= \mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{S}\mathbf{B}^{-1/2}\mathbf{B}^{1/2} = \\ &= \mathbf{B}^{-1/2}\mathbf{O}\mathbf{B}^{1/2}\end{aligned} \tag{2.1}$$

Thus, $\mathbf{L^B}$ and $\mathbf{O} = \mathbf{B}^{-1/2}\mathbf{S}\mathbf{B}^{-1/2}$ have the same real eigenvalues, which makes the mesh spectrum straight-forward to compute [Zhang et al., 2010].

2.4 Spectral Representation of Facial Patches

In this section we present our method for 3D FER based on the local representation of the facial surface by means of the Laplacian spectrum. To this end, we extract local face patches and project them into the spectrum of a fixed-graph basis, which we refer to as Graph Laplacian Features (GLFs). The proposed method is formulated as a generalization of the curves-based framework [Maalej et al., 2011] to a full representation of the local surface patches, which shall improve the descriptive power while keeping the advantages of being a fully-3D framework.

We adopt the widely used approach of representing the facial surface by means of a collection of local patches, centered at L reference points (or landmarks). These landmarks, $\{\mathbf{x}_\ell\}_{1\leq\ell\leq L}$, can be placed either manually or automatically (as discussed in Section 2.5). Following [Maalej et al., 2011], we define the local patches as sets of level curves, where each level curve consists of the surface points that are equidistant to a given landmark. An illustrative example is provided in Fig. 2.3.

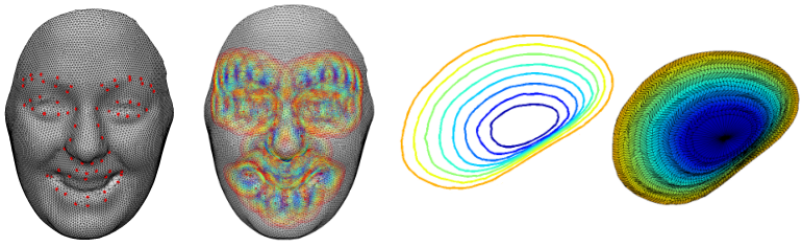

Figure 2.3: (a) 3D annotated facial shape model (68 landmarks); (b) closed curves extracted around the landmarks; (c) example of 8 level curves; (d) the mesh patch.

Formally, each curve $\mathbf{c}_\delta^\ell$ is composed by the set of surface vertices $\mathbf{v} \in \mathcal{M}$ whose distance from the reference landmark $\mathbf{x}_\ell$ equals the specified radius δ. Thus, if we use the Euclidean distance to compute the level curves, the ℓ-th local patch $\mathbf{p}^\ell$ is represented as the collection of level curves whose radii vary from δ_{min} to δ_{max}. That is:

$$\mathbf{p}^\ell = \{\mathbf{c}_\delta^\ell\}_{\delta_{min} \leq \delta \leq \delta_{max}} \tag{2.2}$$

$$\mathbf{c}_\delta^\ell = \{\mathbf{v} \in \mathcal{M} \,|\, \|\mathbf{v} - \mathbf{x}_\ell\| = \delta\} \tag{2.3}$$

Accordingly, each facial surface is represented by L patches that consist of sets of level curves around landmarks (Fig. 2.3(c)).

Once the patches are extracted, we aim to study their shape. Because we want to calculate the mesh spectra for the patches, we need to convert level curves to surface patches. Notice that, conceptually, we may directly extract the patches with no need to first extract the curves, but proceeding this way facilitates comparison to the curves framework from [Maalej et al., 2011] and, as we explain below, allows for using directly the graph Laplacian instead of the discretized LBO. To generate the mesh patches we re-sample the curves uniformly (as done in [Maalej et al., 2011]) and define a unique connectivity between them, which will be shared by all patches (Fig. 2.3(d)).

After these pre-processing steps, we extract spectral features for

facial expression analysis. We propose to do so using the Graph Laplacian, since this is the more theoretically sound approach under our settings. We also compare the results obtained by Shape-DNA, arguably the most widespread method to extract spectral features from 3D meshes. Specifically, spectral features are extracted as follows:

- Graph Laplacian: Whereas Graph Laplacian depends only on the connectivity between vertices, we calculate matrix $\mathbf{L^G}$ using formula (2.1) only once. Eigenvalues and eigenvectors are obtained from this matrix. Because we generate all our mesh patches with the same order of connectivity, the set of eigenvectors constitutes a common basis to represent the spatial spectrum of all patches. Therefore, we use these eigenvectors to project the mesh coordinates of each patch into the common eigenspace. These projections constitute our feature vectors, which capture the geometry of our mesh and are directly comparable between patches:

$$\mathbf{p}^{\ell}_{GLFs} = \mathbf{L^G}\mathbf{p}^{\ell} \tag{2.4}$$

- Shape-DNA: The second type of spectral features is obtained using the discretized LBO (2.1). This operator must be calculated separately on each mesh-patch, because it depends not only on the connectivity but also on the location of the vertices. Thus, the eigen-decomposition of each patch produces a different eigenspace, which is tuned to the geometry of that specific patch. Projections into the eigenspace are therefore no longer comparable, but the eigenvalues resulting from each decomposition have been proven discriminative [Reuter, 2010]. Hence we use the eigenvalues (which correspond to the diagonal elements of matrix Λ) as feature vectors:

$$\mathbf{p}^{\ell}_{DNA} = \{\mathbf{\Lambda}^{\ell}_{ii}\}_{\forall i} \tag{2.5}$$

$$\mathbf{L^B}(\mathbf{p}^{\ell})\mathbf{U}^{\ell} = \mathbf{\Lambda}^{\ell}\mathbf{U}^{\ell} \tag{2.6}$$

Recall that the eigen-decomposition in (2.6) serves to illustrate the concept but it is not the actual way the spectrum is computed (see Eq. 2.1) and the explanation in Section 2.3.2).

Once the spectral features are extracted, they can be directly fed to the classifier for expression recognition. We fix the dimensionality of both GLFs and Shape-DNA features to the first 50 components of the eigenspace, as this setting was shown to perform well for FER in [Derkach and Sukno, 2017]. For the experiments in this chapter we used Support Vector Machines (SVM) invoking the LIBSVM software [Chang and Lin, 2011]. A schematic diagram of the proposed framework is presented in Fig. 2.4.

2.5 3D Landmark Detection

In order to make our system fully automatic, we use Shape Regression with Incomplete Local Features (SRILF) [Sukno et al., 2015] to locate the following 14 facial landmarks: inner and outer eye corners, nose corners, mouth corners, nose root, nose tip, upper and lower middle lip points and chin tip. The SRILF algorithm combines the response from local feature detectors for each of the targeted landmarks with statistical constraints that ensure the plausibility of landmark positions on a global basis. The algorithm has three components: 1) selection of candidates through local feature detection; 2) partial set matching to infer possibly missing landmarks; 3) combinatorial search, which integrates the other two components.

2.5.1 Selection of candidates

The selection of candidates is performed independently for each targeted landmark. Given a mesh $\mathcal{M}$ and a landmark $\mathbf{x}_\ell$ to be targeted, a similarity score $s_\ell(\mathbf{v})$ is computed for every vertex

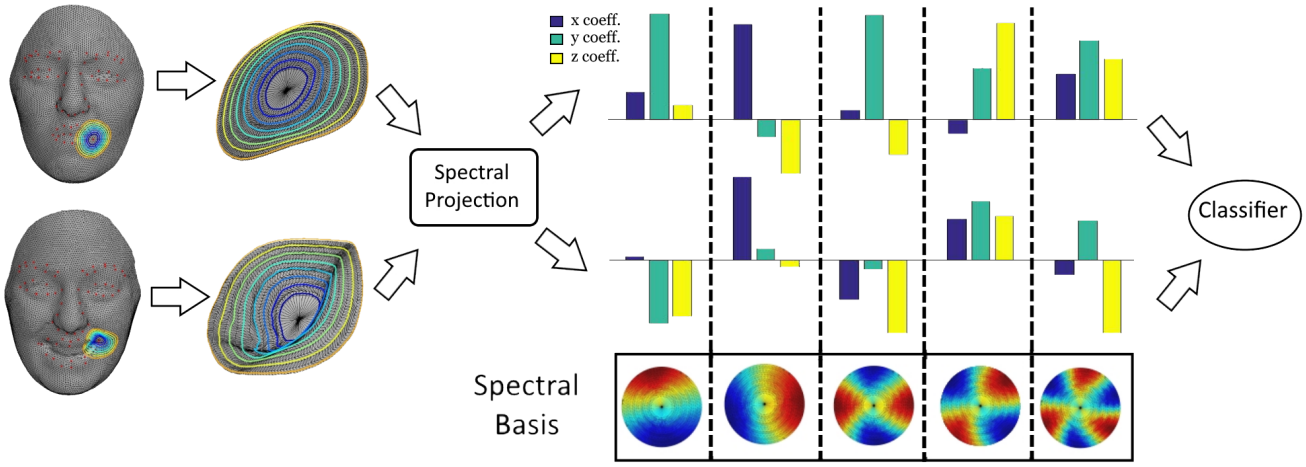

Figure 2.4: Schematic representation of the proposed approach. For each facial landmark, a surface patch is extracted to describe its local geometry. Each patch is projected into a common eigenspace to obtain a set of spectral coefficients that constitute our features. The eigenspace is computed off-line as the spectrum of the Graph Laplacian operator which depends exclusively on the connectivity of vertices and is therefore common for all patches. The spectral coefficients can be interpreted as loadings that weight the contribution of the spectral components. In the figure we display the coefficients of the first 5 spectral components, as well as the spatial patterns produced by their corresponding eigenvectors.

$\mathbf{v} \in \mathcal{M}$; the set of *candidates* $\mathcal{Y}_\ell$ for landmark $\mathbf{x}_\ell$ are the ϱ_ℓ highest scoring vertices:

$$\mathcal{Y}_\ell = \{\mathbf{v} \in \mathcal{M} \,|\, \mathcal{O}(s_\ell(\mathbf{v})) \leq \varrho_\ell\} \tag{2.7}$$

where $\mathcal{O}()$ is the (descending) order function. The score $s_\ell(\mathbf{v})$ is based on the similarity of local surface descriptors with respect to a descriptor template derived at training time. The current implementation of SRILF[1] uses Asymmetry Pattern Shape Contexts [Sukno et al., 2014] as local descriptors.

As in many other algorithms, it is expected that one of these candidates will be close enough to the correct position of the landmark. Nonetheless, the number of false positives (i.e. vertices that produce high similarity scores even though they are far from the correct landmark location) can change considerably for different landmarks, as well as from one facial scan to another, making it difficult to choose the number of candidates that should be retained.

While many approaches try to retain large numbers of candidates to deal with this issue, SRILF uses an upper outlier threshold from the distribution of false positives over a training set. This implies that, in the vast majority of cases, a candidate that is close enough to the target landmark will be detected, but a small proportion will be missed. Hence, for each targeted landmark there will be an initial set of candidates that may or may not contain a suitable solution and we need to match our set of targeted landmarks to a set of candidates that is potentially incomplete. This is analogous to the point-matching problem found in algorithms that search for correspondences. However, the human face is a non-rigid object and these point-matching algorithms are typically restricted to rigid transformations.

[1] A free implementation of the SRILF algorithm is available at http://fsukno.atspace.eu/Data.htm#SRILF_3dFL

2.5.2 Partial set matching

The second component of the algorithm aims at dealing with the above problem. Based on the priors encoded in a statistical shape model, it uses a subset of the landmarks (i.e. those with suitable candidates) to infer the most likely position of the ones that are missing.

Let $\mathbf{x} = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_L, y_L, z_L)^T$ be a shape vector, constructed by concatenating the coordinates of the L targeted landmarks in 3D, and let $\overline{\mathbf{x}}$, $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ be the mean shape, eigenvector and eigenvalue matrices, respectively. Given a shape for which we only know part of its landmarks, we could split it in the known (or fixed) part $\mathbf{x}^f$ and the unknown (to infer or guess) part $\mathbf{x}^g$. Thus, our objective is to infer the coordinates of landmarks $\mathbf{x}^g$ so that the probability that the resulting shape complies with the PCA model is maximized, ideally without modifying the coordinates in $\mathbf{x}^f$.

Let $Pr(\mathbf{x})$ be the probability that shape $\mathbf{x}$ complies with the model. Assuming that $Pr(\mathbf{x})$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ in PCA-space, this probability is proportional to the negative exponential of the Mahalanobis distance and it can be shown [Sukno et al., 2015] that maximization of $Pr(\mathbf{x})$ with respect to $\mathbf{x}^g$ yields:

$$\mathbf{x}^g = \overline{\mathbf{x}}^g - (\mathbf{\Psi}^{gg})^{-1}\mathbf{\Psi}^{gf}\left(\mathbf{x}^f - \overline{\mathbf{x}}^f\right) \tag{2.8}$$

where $\mathbf{\Psi}^{gg} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^g)^T$, $\mathbf{\Psi}^{gf} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^f)^T$ and $\mathbf{\Phi}$ is split in $\mathbf{\Phi}^f$ and $\mathbf{\Phi}^g$ according to $\mathbf{x}^f$ and $\mathbf{x}^g$ (see [Sukno et al., 2015]).

2.5.3 Combinatorial search

The third component of the algorithm integrates the two previous steps into a combinatorial search. It consists of analyzing subsets of candidates and completing the missing information by inferring the coordinates that maximize the probability of a deformable shape model.

Formally, let $\mathcal{F}$ and $\mathcal{G}$ be the sets of fixed and to-infer coordinates, respectively, with $\mathcal{F} \cap \mathcal{G} = \varnothing$ and $\mathcal{F} \cup \mathcal{G} = \{1, 2, \ldots, 3L\}$, where L is the number of targeted landmarks. The goal of the combinatorial search is to dynamically choose the splitting into $\mathcal{F}$ and $\mathcal{G}$ that minimizes the localization error:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} \tag{2.9}$$

where $\mathbf{x}$ are the *true* landmark coordinates and $\hat{\mathbf{x}}$ is the algorithm's estimate. The key concept here is that only the coordinates in $\mathcal{F}$ will be based on image evidence (e.g. the candidates) and the rest will be treated as *missing data*. Thus, $\hat{\mathbf{x}}^g$ will be obtained by inference and it can be expressed as a function of $\hat{\mathbf{x}}^f$, making more apparent that the minimization looks for the optimal subset $\mathcal{F}$:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x}^f - \hat{\mathbf{x}}^f\|^2 + \|\mathbf{x}^g - f(\hat{\mathbf{x}}^f)\|^2\} \tag{2.10}$$

with $f(\hat{\mathbf{x}}^f)$ as defined in Eq. 2.8. Because the true coordinates $\mathbf{x}$ are unknown, we cannot explicitly compute the above errors and need an indirect estimate instead. The SRILF algorithm does this by minimizing (subject to statistical plausibility):

$$\underset{\mathcal{F}}{\operatorname{argmin}} \left(-|\mathcal{F}| - \exp\left(-\sum_{\ell \in \mathcal{F}} \min_{\mathbf{y} \in \mathcal{Y}_\ell} \|\hat{\mathbf{x}}_\ell - \mathbf{y}\|^2 \right) \right) \tag{2.11}$$

where $\mathcal{Y}_\ell$ is the set of candidates for the ℓ-th landmark $\hat{\mathbf{x}}_\ell$. Intuitively, Eq. 2.11 can be understood by noticing that the main component of the cost is the cardinality of $\mathcal{F}$, i.e. the number of landmarks that can be successfully included in $\hat{\mathbf{x}}^f$ while keeping the statistically plausible of the shape. Upon equality of $|\mathcal{F}|$ the cost function increases with the distance from $\hat{\mathbf{x}}$ to the nearest candidate per landmark. These distances to the nearest candidates have a different meaning for fixed and inferred landmarks and help understand the way the algorithm works.

Fixed landmarks $\{\hat{\mathbf{x}}_\ell\}_{\ell\in\mathcal{F}}$ are directly sampled from candidates to guide the combinatorial search. Thus, their nearest candidates are known beforehand and their distance to them is just the reconstruction error of the statistical shape model. For the remaining landmarks, $\{\hat{\mathbf{x}}_\ell\}_{\ell\in\mathcal{G}}$, positions are statistically inferred from Eq. 2.8 independently from their candidate sets. It would be expected that better predictions generate inferred landmarks that are closer to their corresponding candidates, resulting in lower cost values.

An important aspect of the splitting between $\mathcal{F}$ and $\mathcal{G}$ is that it inherently provides tolerance to distorted or missing data (occlusions). Notice that there is no prior assumption regarding what landmarks can be in $\mathcal{F}$ or $\mathcal{G}$ nor the cardinality of the two sets and the splitting is performed dynamically on a case by case basis.

2.6 Facial Expression Recognition Experiments

In order to evaluate the proposed spectral features for local shape representation we conducted experiments on the three most widely used databases for 3D FER. We start by briefly describing each of these datasets and the corresponding experimental settings. Expression recognition results are reported later in this section, while Section 2.7 provides our results in Action Unit recognition.

2.6.1 BU-3DFE Database

The BU-3DFE database has been developed by Yin et al. [Yin et al., 2006]. from Binghamton University. This database consists of 3D face scans of 100 subjects with different facial expressions. There are also variations in race, gender and age. Scans are annotated according to the six prototypical facial expressions (anger, disgust, fear, happiness, sadness and surprise) at four different intensity levels.

For each model, the database provides a cropped image containing just the face; a non-cropped image that contains both side views of the face; 83 manually annotated landmarks of the main facial features and a single 3D mesh containing the coordinates of the face with a triangulation containing between 25 to 35 thousand polygons.

For our experiments, we used the 3D scans from all 100 subjects at the two highest intensity levels, Thus, our dataset consists of 1200 3D face scans, namely two intensity levels for each of the six facial expressions from 100 subjects.

Our first set of experiments were performed using the landmarks provided the database. As mentioned above, accompanying each facial scan there are 83 manually labeled landmarks. From these, 15 landmarks correspond to the silhouette contour and have arguably little validity in a 3D setting, hence we considered only the subset of $L = 68$ landmarks laying within the face area. Further, all facial scans have been represented by 68 patches $\mathbf{p}^\ell$, where, each patch consisted of 15 level curves $\{\mathbf{c}_\delta^\ell\}_{\delta_{min} \leq \delta \leq \delta_{max}}$ with $\delta_{min} = 5, \delta_{max} = 20$ (Fig. 2.3(c)).

The second set of experiments were performed with automatically detected landmarks (Section 2.5). In this case, we have $L = 14$ landmarks and, correspondingly, 14 local patches to represent each facial scan.

All experiments were performed following a 10-fold cross-validation. The dataset was arbitrarily divided into ten identity-disjoint sets; each of these (composed by 120 samples) was tested with models trained from the remaining nine sets (1080 samples). Recognition rates are obtained by averaging the results over the 10 sets.

2.6.2 BU-4DFE Database

The BU-4DFE database is a dynamic 3-D facial expression database, which has been created at Binghamton University [Yin et al., 2008]. This database contains a total of 101 subjects (58 female and 43 male, with an age range of 18–45 years old). Each subject performs the six

prototypical expressions (anger, disgust, fear, happiness, sadness and surprise) resulting in 606 sequences of 3D meshes. Similar to The BU-3DFE database, BU-4DFE provides 83 annotated landmarks.

Each facial expression sequence included in BU-4DFE database normally contains about 100 frames, each of which is a 3D face mesh. For our experiments, we have chosen two facial scans from the central frames of each sequence for all 101 subjects. Hence, our subdataset consists of 1212 3D face scans. Under these settings, the resulting dataset is comparable to the one of BU-3DFE described in the previous section, both in terms of number of scans, subjects and intensity levels, since the central frames of each sequence in BU-4DFE are expected to be near the expression apex.

Thus, in analogy to test in BU-3DFE, two experiments were performed in BU-4DFE. Firstly based on the the 68 provided landmarks silhouette contour, and secondly based on the 14 automatically detected landmarks. In both cases, experiments were performed under 10-fold cross-validation.

2.6.3 Bosphorus Database

The Bosphorus database [Savran et al., 2008] contains images from 105 subjects labeled in terms the Facial Action Coding System (FACS). There are up to 54 face images per subject and these images involve both the six prototypical expressions and instances of Action Units (AUs). For our FER experiments, we have chosen all 3D scans showing any of the six prototypical expressions for all 105 subjects, which amounts to 453 scans[2]. Each facial scan is provided with a set of 24 manual landmarks, from which we exclude two in the ears and the one in the chin for not being always visible, ending up with $L = 21$ landmarks that were used for our first set of experiments. As before, we also repeat our experiments with 14 automatically located landmarks and perform 10-fold cross validation experiments.

[2]Notice that not all subjects performed all six expressions. We included all subjects regardless of the number of expressions they performed.

2.6.4 Results

Our facial expression recognition experiments consist on a direct comparison of the proposed spectral features (GLFs) with respect to the curves-framework from [Maalej et al., 2011] and Shape-DNA, which constitute the straight-forward spectral alternative. This was done by targeting the six basic expressions present in the selected databases: anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA) and surprise (SU). Multi-class SVMs were used for classification. Table 2.1, summarizes the average accuracy obtained by each approach on over the three test databases. Notice that, in the case of GLFs, we also report results under a fully-automatic setting.

Method	BU-3DFE	BU-4DFE	Bosphorus
Curves	78.2%	66.94%	59.14%
Shape-DNA	73.62%	61.19%	56.67%
GLFs	**81.5%**	**74.47%**	**77.33%**
Automatic GLFs	**76.5%**	**71.43%**	**71.11%**

Table 2.1: Average accuracy of the three methods for facial expression recognition on the three database. The first three rows correspond to results using the manual landmarks provided for each database; the last row corresponds to fully-automatic experiments using 14 landmarks detected by the SRILF algorithm (Section 2.5).

It can be seen that the average accuracy of the spectral features based on the Graph Laplacian varies between approximately 75% and 81%, depending on the database. When using automatic landmarks, the accuracy drops only between 3% and 5%, even though the number of landmarks is considerably reduced. The table also shows that proposed method outperforms the curves-based approach on all databases. This is consistent with the hypothesized advantage of using GLFs, as these can capture a more complete description the facial patches when compared to the level curves.

It is also interesting to see that Shape-DNA features obtain the lowest accuracy among the three methods, with even lower accuracy than the fully-automatic GLFs. This confirms the theoretical limitations already highlighted with respect to the direct application of Shape-DNA to surface patches: given two shapes to compare under a spectral representation, small differences between them can modify the eigen-decomposition to the extent that the eigenvalues change their relative order producing a swapping of the extracted bases [Jain et al., 2007]. Such swaps make the direct comparison of eigenvalues used in Shape-DNA conceptually incorrect (Fig. 2.5). Fixing this would require matching algorithms to appropriately re-order the resulting eigenvalues. Our GLFs do not suffer from this issue as they result from a projection into a common basis, which only depends on the connectivity and is therefore shared by all patches.

To provide a more extensive review of our results, Fig. 2.6 shows the average accuracy per expression of each method on the three different databases. It can be seen that, among the six basic expressions, happiness, surprise and anger achieved the highest accuracy in all datasets. In contrast, fear and disgust were the most difficult expressions to predict. We also observe that GLFs consistently outperform both the curve-based and Shape-DNA approaches for most expressions. Moreover, even the results with fully-automatic GLFs compare favorably to those from curves and Shape-DNA with manual landmarks, performing similarly on BU-3DFE and Bosphorus and outperforming the alternative methods on the BU-4DFE database.

To put our results in a wider context, we also compare them to other methods reporting FER rates on the three aforementioned databases. Table 2.2 and Table 2.3 summarize the comparison to earlier results. Expression recognition rates vary between 70.9% and 86.3% on the BU-3DFE database and between 60% and 79% on Bosphorus, with our average recognition rates reaching 81.5% and 77.3% respectively. Notice that in our case we use a single type of

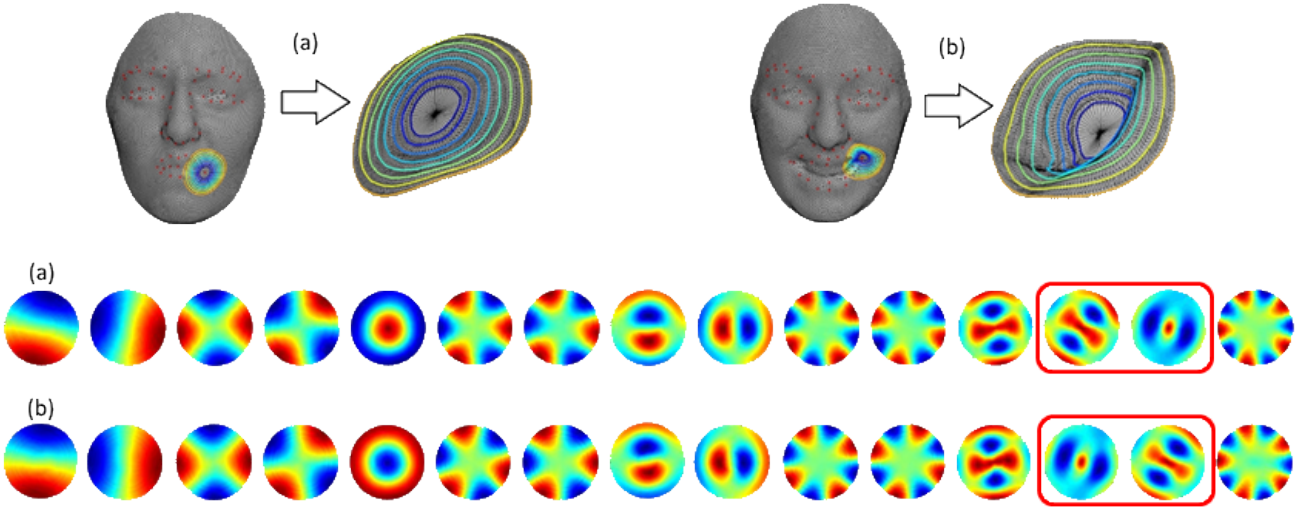

Figure 2.5: An example of the spectral bases extracted by Shape-DNA for patches of two different facial scans. The figure shows the bases (eigenvectors) extracted for patches centered at the left mouth corner, ordered from left to right by the magnitude of their eigenvalues. While the ordering of the first few eigenvectors usually matches for different patches (the sign-flip at the 5th basis is not a problem), such ordering is not preserved as we include more bases. In the example, the positions of the 13th and 14th bases are swapped, which makes the direct comparison of their eigenvalues conceptually incorrect. As more bases are included, swaps become more frequent.

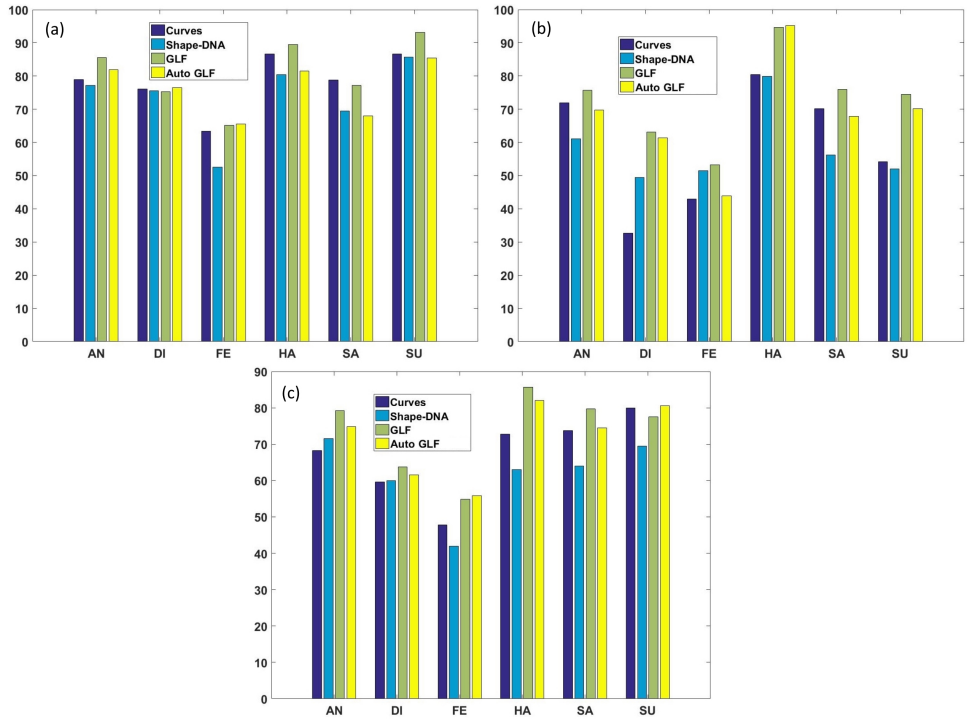

Figure 2.6: Average accuracy per expression for each method on the three database ((a) BU-3DFE database; (b) Bosphorus database; (c) BU-4DFE dattabase).

feature (GLFs), while most other works achieving high recognition rates use combinations of multiple features, sometimes including also texture (2D) and often considering only part of the available data.

Indeed, the comparison of recognition rates must be done carefully, as not all papers use the same number of subjects in their experiments. For example, tests reported on the BU-3DFE have followed two main strategies comprising 60 or 100 subjects, respectively. Another relevant aspect to consider is whether the results are reported under fully-automatic operation. With these

Methods	Automatic	2D/3D	Accuracy	
			60subj	100subj
Azazi et.al [Azazi et al., 2015]	Yes	2D + 3D	79.36%	-
Berretti et.al [Berretti et al., 2010]	No	3D	-	77.53%
Jan et.al [Jan and Meng, 2015]	No	3D	81.25%	-
Li et.al [Li et al., 2012]	Yes	3D	80.14%	-
Li et.al [Li et al., 2015a]	Yes	2D + 3D	86.32%	-
Vretos et.al [Vretos et al., 2011]	Yes	3D	-	73%
Yang et.al [Yang et al., 2015]	No	3D	82.7%	-
Zeng et.al [Zeng et al., 2013]	Yes	3D	-	70.93%
Zhang et.al [Zhen et al., 2016]	Yes	3D	83.2%	-
GLFs	No	3D	**76.94% – 86.25%**	**81.5%**
Automatic GLFs	Yes	3D	**73.39% – 81.86%**	**76.5%**

Table 2.2: Comparison of the proposed method to results from other 3D methods on the BU-3DFE database. Accuracy scores are separated according to the number of subjects that were used in each paper.

Methods	Automatic	2D/3D	Accuracy			
			60sub	65sub	100sub	105sub
Azazi et.al [Azazi et al., 2015]	Yes	2D+3D	79%	-	-	-
Jan et.al [Jan and Meng, 2015]	No	3D	-	-	-	75.68%
Vretos et.al [Vretos et al., 2011]	Yes	3D	-	60.5%	-	-
Wang et.al [Wang et al., 2013b]	No	3D	-	-	76.5%	-
GLFs	No	3D	-	-	-	**77.33%**
Automatic GLFs	Yes	3D	-	-	-	**71%**

Table 2.3: Comparison of the proposed method to results from other 3D methods on the Bosphorus database. Accuracy scores are separated according to the number of subjects that were used in each paper.

Methods	Automatic	3D/4D	Accuracy	
			60subj	100subj
Berretti et.al[Berretti et al., 2013]	Yes	4D	72.3%	-
Fang et.al[Fang et al., 2011b]	Yes	4D	-	75.82%
Reale et.al[Reale et al., 2013]	No	4D	-	76.1%
Sandbach et.al[Sandbach et al., 2012b]	Yes	4D	-	61.3%
Xue et.al [Xue et al., 2015]	Yes	4D	78.8%	-
GLFs	No	3D	-	**74.13%**
Automatic GLFs	Yes	3D	-	**71.43%**

Table 2.4: Comparison of the proposed method to results from other 3D and 4D methods on the BU-4DFE database. Accuracy scores are separated according to the number of subjects that were used in each paper.

considerations, we see that the proposed GLFs achieve the highest FER rates reported to date on both BU-3DFE and Bosphorus databases using all available subjects. This also holds under fully-automatic operation.

As stated in [Zeng et al., 2013], experimental settings considering the whole set of subjects facilitate fair comparison and should be preferred. Nevertheless, to facilitate a wide comparison to previous works we also report results using 60 subjects. Because in such case the accuracy depends on the specific subjects that are selected, we do not provide a single accuracy value but the range of recognition rates obtained in 1,000 independent experiments. For each experiment, 60 randomly selected subjects were considered under a similar 10-fold cross-validation strategy to the one followed for the 100 subject experiment.

Additionally, Table 2.4 shows a comparison of our method to previous works on the BU-4DFE database. Notice that all other methods compared in the table use 4D data, i.e. they take advantage of the temporal information available in this dataset, while our method only considers static information and performs FER by taking decisions on a per-frame basis. In spite of this, we see that the accuracy of GLFs is only about 2% below the top-performing method ($\sim$ 4.5% in the case of methods under fully-automatic operation).

2.7 Experiments on Action Unit Estimation

Action Units (AUs) are designed to characterize the facial surface under any anatomically feasible facial deformation [Ekman, 1994]; thereby combinations of AUs can be used to describe any of the six basic expressions [Ruiz et al., 2015], as well as any other anatomically feasible facial expression. Since our approach is based on the aggregation of localized descriptors of the facial surface, it is

reasonable to expect that it can also be applied to the estimation of Action Units (AU).

In this section we address AU estimation with the proposed GLFs. Most experimental settings are equivalent to those in the previous section, although the classification strategy had to be modified given the possibility of co-occurrence of AUs. In other words, while each facial scan is labelled with a single expression, it can contain several AUs. Thus, instead of a multi-class SVM classifier, we used multiple binary SVMs (one per AU).

From the three databases used in this chapter, only Bosphorus is provided with AU annotations. Thus, we have also manually annotated AUs on the BU-3DFE database and used these two databases to perform AU estimation experiments. These AU annotations have been made publicly available on-line[3].

2.7.1 BU-3DFE database

We used the same set of 1200 scans selected for our FER experiments in the BU-3DFE database. Each of these scans, containing expressions, was manually annotated with a corresponding set of AUs by two coders. The resulting annotations are summarized in Table 2.5, where we show the AU frequencies per expression (i.e. the percentage of times that each AU was found present in a given expression)[4]. The obtained AU frequencies per expression and co-occurrences of AUs are consistent with previous studies reported in the literature [Du et al., 2014, Wang et al., 2014a, Zhao et al., 2015]. Using these annotations as ground truth, experiments on AU recognition were performed under the same conditions as the expressions recognition tests.

Table 2.6 shows the F1-score for each AU, together with a weighted average (weighted proportionally to the number of samples

[3]http://fsukno.atspace.eu/Research.htm#FG2017a

[4]For clarity of the presentation we only indicate occurrence percentages if these are above 5%.

	Action Units																
	1	**2**	**4**	**5**	**6**	**7**	**9**	**10**	**12**	**15**	**16**	**17**	**20**	**23**	**24**	**25**	**26**
AN			75%			45%	8%					75%			45%	8%	
DI			77,5%			70,5%	40%	63%								83,5%	11%
FE	46,5%	37,5%	23,5%	39,5%		14%					6%		31,5%			72%	5,5%
HA	14%	13%			33%	25,5%			84,5%							95%	
SA	13,5%	5,5%	35,5%	7,5%		30%				32%	50,5%	45%		20,5%	12%		
SU	86,5%	88,5%		93,5%												98,5%	96%

Table 2.5: Percentage of times that each AU was found present for each facial expression in the BU-3DFE database

AU	# smpl	Curves	Shape-DNA	GLFs	Auto GLFs
1 - Inner Brow Raiser	333	0.74	0.73	0.75	0.68
2 - Outer Brow Raiser	302	0.77	0.73	0.78	0.71
4 - Brow Lowerer	423	0.77	0.74	0.79	0.74
5 - Upper Lid Raiser	304	0.76	0.71	0.80	0.75
6 - Cheek Raiser	68	0.42	0.45	0.46	0.41
7 - Lid Tightener	370	0.69	0.63	0.73	0.68
9 - Nose Wrinkler	99	0.55	0.47	0.56	0.57
10 - Upper Lip Raiser	136	0.64	0.57	0.67	0.69
12 - Lip Corner Puller	177	0.74	0.70	0.76	0.73
15 - Lip Corner Depr	69	0.37	0.30	0.34	0.41
16 - Lower Lip Depr.	122	0.50	0.39	0.52	0.52
17 - Chin Raiser	130	0.48	0.42	0.50	0.45
20 - Lip Stretcher	84	0.28	0.25	0.3	0.21
23 - Lip Tightener	134	0.42	0.38	0.50	0.44
24 - Lip Presser	125	0.57	0.62	0.61	0.54
25 - Lips Part	709	0.94	0.92	0.94	0.94
26 - Jaw Drop	230	0.85	0.86	0.88	0.85
Avrg	3815	**0.72**	**0.69**	**0.74**	**0.71**

Table 2.6: Average F1-score results of AUs recognition on BU-3DFE database

per AU) for the proposed GLFs as well as shape-DNA and curves. One common characteristic of all three approaches is that they recognized AU25 and AU26 better that any other AU. Also, analyzing the table, we can see that detection of AU1, AU2, AU4, AU5 and AU12 can be said reliable. The worst results correspond to AU15.

When comparing among features, our results show the same tendency observed in the expression recognition experiments. The best performance was obtained by GLFs, which clearly outperformed Shape-DNA and was also slightly better than the curves framework. Regarding the latter, while the average recognition accuracy of GLF and curves were rather similar, it should be noted that

non-automatic GLFs consistently outperformed curves in 15 out of the 17 tested AUs.

2.7.2 Bosphorus database

AU	# smpl	Curves	Shape-DNA	GLFs	Auto GLFs
1 - Inner Brow Raiser	46	0.44	0.14	0.56	0.38
2 - Outer Brow Raiser	105	0.69	0.46	0.76	0.73
4 - Brow Lowerer	105	0.49	0.48	0.72	0.68
9 - Nose Wrinkler	99	0.62	0.62	0.78	0.79
10 - Upper Lip Raiser	71	0.69	0.49	0.65	0.68
12 - Lip Corner Puller	305	0.54	0.49	0.73	0.66
14 - Dimpler	73	0.33	0.32	0.46	0.36
15 - Lip Corner Depr.	55	0.26	0.17	0.38	0.46
16 - Lower Lip Depr.	70	0.47	0.33	0.53	0.52
17 - Chin Raiser	71	0.54	0.37	0.65	0.62
18 - Lip Puckerer	71	0.71	0.51	0.68	0.60
20 - Lip Stretcher	65	0.44	0.25	0.59	0.44
22 - Lip Funneler	70	0.69	0.47	0.71	0.68
23 - Lip Tightener	72	0.47	0.27	0.59	0.61
24 - Lip Presser	70	0.47	0.29	0.49	0.37
25 - Lips Part	70	0.49	0.39	0.62	0.46
26 - Jaw Drop	72	0.56	0.38	0.58	0.69
27 - Mouth Stretch	105	0.79	0.70	0.81	0.88
28 - Lip Suck	105	0.78	0.66	0.80	0.65
34- Cheek Puff	105	0.66	0.51	0.89	0.92
43 - Eyes Closed	105	0.80	0.47	0.86	0.75
44 - Squint	71	0.32	0.15	0.57	0.53
Avrg	1981	**0.58**	**0.44**	**0.69**	**0.64**

Table 2.7: Average F1-score results of AUs recognition on Bosphorus database

As mentioned previously, the Bosphorus database is provided with AU annotations following the FACS. The annotated images can be divided in two categories: *i*) faces displaying a single AU; *ii*) faces

AU	GLFs	Auto GLFs	[Savran and Sankur, 2009]	[Sandbach et al., 2012a]
1 - Inner Brow Raiser	93.6	89.4	95	91.1
2 - Outer Brow Raiser	98.2	96.8	98	98.4
4 - Brow Lowerer	97.4	96.8	97	96.9
9 - Nose Wrinkler	97.0	98.8	96	97.9
10 - Upper Lip Raiser	98.2	97.2	99	97.7
12 - Lip Corner Puller	94.8	94.6	98	95.9
14 - Dimpler	90.4	91.5	96	91.4
15 - Lip Corner Depr.	91.9	91.5	89	83.6
16 - Lower Lip Depr.	96.7	96.6	97	96.7
17 - Chin Raiser	96.6	94.2	94	94.7
18 - Lip Puckerer	97.9	98.1	97	97.2
20 - Lip Stretcher	93.4	91.6	95	92.5
22 - Lip Funneler	97.6	97.3	98	99.3
23 - Lip Tightener	93.0	96.8	92	95.1
24 - Lip Presser	90.1	89.5	88	89.6
25 - Lips Part	95.0	95.4	95	94.8
26 - Jaw Drop	96.0	96.7	97	95.1
27 - Mouth Stretch	98.8	96.8	98	99.4
28 - Lip Suck	99.3	97.1	96	97.9
34 - Cheek Puff	99.5	99.8	98	99.1
43 - Eyes Closed	99.5	98.0	98	99.7
44 - Squint	93.9	94.2	-	94.1
Avr.	**95.85**	**95.40**	**95.76**	**95.37**

Table 2.8: Comparison of AuC values achieved with GLFs and previous works for 22 AUs on the Bosphorus database.

displaying combinations of AUs. To facilitate comparison to other works, we follow the settings from [Savran and Sankur, 2009] and [Sandbach et al., 2012a], and use only the scans containing a single AU, which amount to 1981 samples. As in all previous experiments, we followed a 10-fold cross-validation over the data set, ensuring that the folds were identity-disjoint sets. This ensures that subjects in the test set are unseen in the training set.

Table 2.7 shows the F1-scores for each AU and the weighted average. It can be seen that, as in BU-3DFE, the proposed GLFs outperform curves and Shape-DNA, consistently for most of the AUs. Moreover, the results of GLFs under fully-automatic operation are still better (on average) than both competing alternatives.

To put our results in a wider context, we provide in Table 2.8

a comparison of our method to other works targeting AU detection from 3D data on the Bosphorus database: the work by Savran et al. [Savran and Sankur, 2009], based on surface curvature features, and the work by Sandbach et al. [Sandbach et al., 2012a], based on a variant of LBPs applied to surface normals. Notice that, because the metric used by those works is not F1-score but Area under the Curve (AuC), we also adopt this metric for the comparison table. The AuC refers to the area under the Receiving Operating Characteristic (ROC) curve and can be interpreted as an estimate of the probability that a random positive is ranked higher than a random negative, without the need to choose a particular decision threshold [Ferri et al., 2011].

Table 2.8 shows the AuC values for each of the 22 AUs present in the Bosphorus database, as well as the average AuC for each method. We can see that, in terms of average results, GLFs are slightly better than both [Savran and Sankur, 2009] and [Sandbach et al., 2012a], but the differences between the three methods are very small. Indeed, each of the three compared methods outperforms the rest for a set of seven AUs, with a tie between GLFs and Savran's method [Savran and Sankur, 2009] for the remaining one (AU25) to complete the list of 22 AUs.

Results under fully automatic operation are also provided for GLFs (neither [Savran and Sankur, 2009] nor [Sandbach et al., 2012a] fall in this category). We can see the AuC values are again similar to those achieved by the non-automatic methods, even outperforming these for some AUs.

The detection performance of our nethod is further illustrated in Fig. 2.7 by means of ROC curves of some AUs. These AUs were selected to show the highest and lowest AuC values obtained in our experiments. In each curve, we have also indicated the operation threshold, and the corresponding F1-score that is obtained with it.

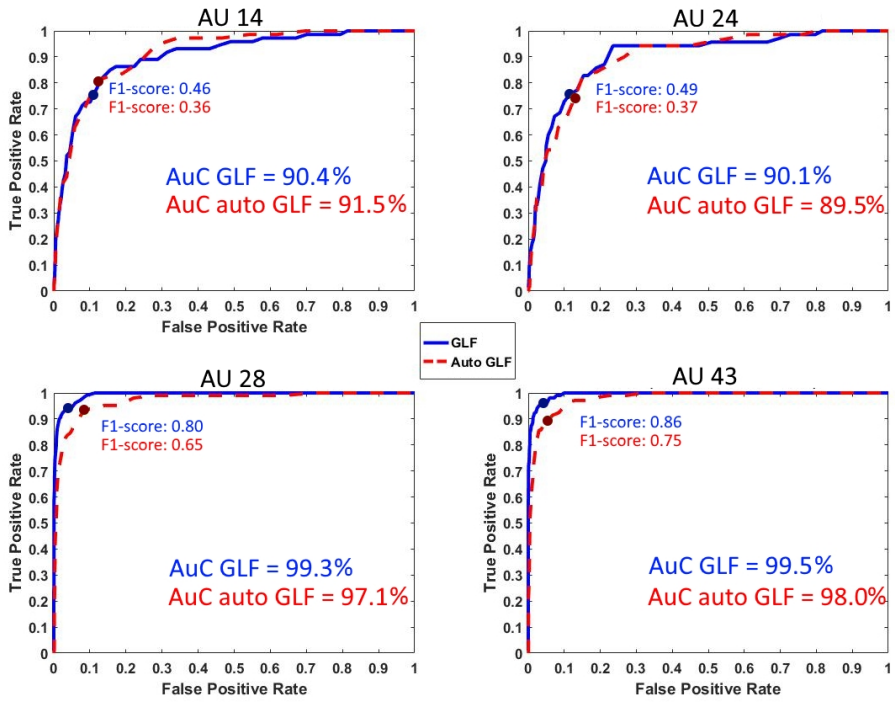

Figure 2.7: ROC curves of various AUs using GLFs on the Bosphorus database. The solid blue curve shows the performance of GLFs using manual landmarks while the dashed red line shows the performance under fully-automatic operation. For each curve we provide the AuC value and the operating point (marked with a dark dot), as well as the resulting F1-score.

2.8 Conclusions

In this chapter, we extend the analysis of 3D geometry from a curve-based representation into a spectral representation. This representation allows to build a complete description of the underlying surface while maintaining a fully-3D framework. We propose the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace, much like a Fourier transform into the spatial frequency bases of the surface patches. Further, we compare our approach with two others approaches. The first one is the curves-based framework and the second one is the straight-forward alternative for spectral representation, Shape-DNA, which is based on the Laplace Beltrami Operator. We show that the straight-forward application of Shape-DNA is not the best way to deal with local face patches, since it cannot provide a stable basis to guarantee that the extracted signatures for the different patches are directly comparable.

We tested the proposed approach in the three most popular databases for 3D FER (BU-3DFE, Bosphorus and BU-4DFE) in terms of FER rates and, additionally, in terms of AU recognition when AU labels were available (BU-3DFE and Bosphorus). Our results show that the proposed GLFs consistently outperform the curves-based and Shape-DNA alternatives, both in terms of expression recognition and AU recognition. Moreover, the recognition rates of Shape-DNA are even lower than those in the curves-based framework, as predicted by the theory: in spite of upgrading the curves-based representation to a full-surface description, similarly to GLFs, the instabilities of the bases extracted by Shape-DNA result in a decreased performance.

Interestingly, the accuracy improvement brought by GLFs is obtained also at a lower computational cost. Considering the extraction of patches as a common step between the three compared approaches, the curves-based framework requires a costly elastic

deformation between corresponding curves (e.g. based on splines) and Shape-DNA requires computing the eigen-decomposition of each new patch to be analyzed. In contrast, GLFs only require the projection of the patch geometry into the Graph Laplacian eigenspace, which is common to all patches and can thus be pre-computed off-line.

Comparison to other works reporting 3D FER and AU detection results confirmed that the proposed method allows achieving top performance by simply feeding GLFs to off-the-shelf SVM classifiers. A state-of-the-art algorithm for 3D landmark localization was also integrated, which enabled us to perform experiments under fully-automatic operation. We showed that 14 automatically detected landmarks were enough to achieve high FER and AU detection rates, only slightly below those obtained when using sets of manually provided landmarks.

Chapter 3

HEAD POSE ESTIMATION BASED ON 3-D FACIAL LANDMARKS LOCALIZATION AND REGRESSION

Adapted from: D. Derkach, A. Ruiz F.M. Sukno. "Head pose estimation based on 3-D facial landmarks localization and regression". *In Automatic Face & Gesture Recognition (FG 2017)*, 2017 12th IEEE International Conference on (pp. 820-827). IEEE. DOI: 10.1109/FG.2017.104

Abstract

In this Chapter we present an approach for accurate head pose estimation from a single depth frame of consumer RGB-D cameras, such as Kinect 2. In contrast to most existing approaches, we base our system in the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based pose estimators. A key aspect of the proposed system is the use of state of the art landmark localization with no need for initialization and tolerance to occlusions or missing data. Our system is complemented with a secondary pose estimator based purely on patches sampled randomly on the head region to account for potential failures of the landmark-based estimation.

We evaluated our system on the SASE database, which consists of $\sim$ 30K frames from 50 subjects. We obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge. Our experiments also confirmed the initial hypothesis that the landmark-based estimates would be more accurate than correspondence-free approaches, such as the dictionary-based one that was adopted. Landmark-based estimates were successfully produced for $\sim$ 90% of cases and the remaining ones were tackled by the dictionary-based approach. Our results compare well with those reported in the related literature, especially considering the added difficulty of not using tracking and RGB data to produce our estimates.

3.1 Introduction

Human head-pose estimation has attracted a lot of interest because it is usually the first step of many face analysis tasks. It is an important aspect in facial motion capture, human-computer interaction and video conferencing, as well as a prerequisite for face recognition or facial expression analysis. Head pose estimation has traditionally been performed on RGB images, but recent advances in 3D geometry acquisition have led to a growing interest in methods that operate on 3D data. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust [Seemann et al., 2004].

The goal of head pose estimation is to predict the relative orientation between the target head and the viewer or camera. It is usually parametrized by the head's pitch, yaw and roll angles. An early attempt to classify head pose estimation methods from a methodological perspective was presented by Murphy et al. [Murphy-Chutorian and Trivedi, 2009], who proposed 8 categories including appearance template methods, flexible models, non-linear regression and tracking. While that classification included both 2D and 3D methods, in this Chapter we focus on head estimation based exclusively on depth information. This considerably reduces the number of categories to: geometric methods [Sun and Yin, 2008, Li and Pedrycz, 2014], appearance methods [Papazov et al., 2015], [Breitenstein et al., 2008], [Tulyakov et al., 2014], regression methods [Fanelli et al., 2011], flexible models [Meyer et al., 2015], [Baltrušaitis et al., 2012] and tracking methods [Papazov et al., 2015].

An important aspect of 3D head pose estimation algorithms is whether RGB data or temporal information are used. Firstly, RGB data can provide complementary information to the one provided by depth data, especially at the detection stage, but it is likely to reduce the robustness to illumination that is inherent to 3D-only data. It is also very popular to make use of dynamic information to improve head pose orientation results. However, algorithms using

tracking often benefit from the fact that test sequences usually start with near-frontal head orientations and, therefore, it is not clear their robustness to detect initial head poses other than frontal, which are arguably more challenging.

In this Chapter, we present an approach for accurate static 3D head pose estimation which is able to perform head-pose estimation using only depth information from a single Kinect 2 frame of a person sitting in front of a camera. This setup has been specified in the FG2017 Head-Pose Estimation Challenge [Lüsi et al., 2017]. In contrast to most existing approaches, we base our system in the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based pose estimators. A key aspect of the proposed system is the use of Shape Regression with Incomplete Local Features (SRILF) [Sukno et al., 2015] for landmark localization. This algorithm provides state of the art landmark localization accuracy with no prior initialization and is inherently tolerant to occlusions or missing data. The latter is very important when capturing moderate or large head rotations with a single-view depth sensor such as Kinect 2 since, in such cases, large parts of the face become unavailable due to self-occlusions. Our system is complemented with a secondary pose estimator based purely on patches sampled randomly on the head region to account for potential failures of the landmark-based estimation. Our tests on the SASE database [Lüsi et al., 2016b] provided in the FG2017 Head-Pose Estimation challenge, showed average estimation errors of 7.82, 6.65 and 5.39 degree for pitch, yaw and roll angles, respectively.

3.2 Related Work

As aforementioned, an important aspect of 3D head pose estimation algorithms is whether or not they use RGB data and tracking. Only few of methods has addressed this problem without the use of temporal information.

For instance, Sun and Yin proposed a geometric feature based pose estimation approach based on 3D facial models [Sun and Yin, 2008]. The pose orientation was estimated using a symmetry plane. Li and Pedrycz [Li and Pedrycz, 2014] developed a central profile-based 3D face pose estimation algorithm. The central profile is the intersection curve, that starts from forehead center, goes down through nose ridge, nose tip, mouth center, and ends at a chin tip. It is also called symmetry plane. They defined an objective function for conducting the Hough transform in parameter space that maps face profile to an accumulator cell. The face profile corresponding to the maximum accumulator cell was regarded as the central profile. Once the symmetry plane had been completed, two angles (roll and yaw) were determined, since the objective function was based on three parameters. Based on the detection of central profile, nose tip was detected and pitch angle was estimated using the coordinates of three points nose tip, nose ridge point and nose bottom point. Valle et al. [Valle et al., 2016] also presented a free-tracking algorithm that estimates the head pose, but they estimated only one yaw angle from unrestricted 2D gray-scale images. In order to obtain a discrete head-pose estimation, they proposed a classification scheme, based on a random forest, where patches randomly extracted from the image cast votes for the corresponding discrete head-pose angle. Papazov et al. [Papazov et al., 2015] presented a real-time system for 3D head pose estimation using a commodity depth sensor such as Microsoft's Kinect. The proposed method consists of an offline training and an online testing phase. In both phases, 2D information was used for face detection. After that, a triangular surface patch (TSP) descriptor, which encodes the shape of the 3D face surface within a triangular area, was employed for final angle estimation. For testing, the authors utilized two approaches: tracking mode and detection mode (static).

Another free-tracking approach was presented in [Breitenstein et al., 2008]. Breitenstein et al. developed an error function that compares the input range image to precomputed pose images of an

average face model. In an offline step, range images of an average face were rendered for many poses, and the resulting reference pose range images were saved. For each pixel they computed signatures that are distinct for regions with high curvature, such as the nose tip. This yielded a set of candidate nose positions and orientations that were used as head pose hypotheses. Then they computed the error between the reference pose range images corresponding to the pose hypotheses and the input range image using a novel error function. The match with the lowest error yielded the final pose estimation and a confidence value. In [Wang et al., 2013a], an approach was presented to estimate the 3D position and orientation of head from single RGB and depth images. 2D Scale-invariant feature transform (SIFT) features were used together with 3D histogram of oriented gradients (HOG) features, which were extracted in a pair of RGB and depth images captured synchronously. Random forests approach were then applied in order to formulate pose estimation as a regression problem, due to their power for handling large training data and the high mapping speed. Finaly, the mean-shift method was employed to refine the result obtained by the random forests.

Similarly, Fanelli et al. [Fanelli et al., 2011] used random forests to handle large training datasets and formulated a real-time head pose estimation as a regression problem for tracking purposes. In [Tulyakov et al., 2014], authors proposed a fusion approach to address real-time head pose estimation. They constructed a system able to recover itself (in cases where the tracking was lost) by combining a frame independent decision tree based estimator with a personalized template tracker.

An alternate approach, using depth as well as intensity information, was presented by Baltrusaitis et al. [Baltrušaitis et al., 2012]. The authors presented 3D Constrained Local Model (CLM-Z) for the facial feature tracking under varying pose. A two-step CLM fitting strategy was employed: performing an exhaustive local search around the current estimate of feature points leading to a response map around every feature point, and then iteratively updating

the model parameters to maximize a posterior probability until a convergence metric is reached. For fitting, they used Regularised Landmark Mean-Shift (RLMS). Another relevant paper, by Padeleris et al. [Padeleris et al., 2012], estimated the pose of an input Kinect sensor depth map by finding the 3D rotation of a template that best matched the input. The proposed method searches for a view at which the rendered image matches the reference depth image obtained during an initialization phase. At run time, the method searches the 6-dimensional pose space to find a pose from which the head appears identical to the reference view. This registration was treated as an optimization problem that was solved through Particle Swarm Optimization (PSO). One more approache based on PSO was presented by Meyer et al. [Meyer et al., 2015]. They performed pose estimation by registering a morphable face model to the measured depth data, using a combination of particle swarm optimization (PSO) and the iterative closest point (ICP) algorithm.

Martin et al. [Martin et al., 2014] presented approach for head pose estimation on consumer depth cameras that works without prior knowledge of the tracked person and without prior training of detector. To achieve this, they combined an algorithm to generate and track a model of the head with feature based head pose estimation. This algorithm was based on tracking a head model using the iterative closest point algorithm.

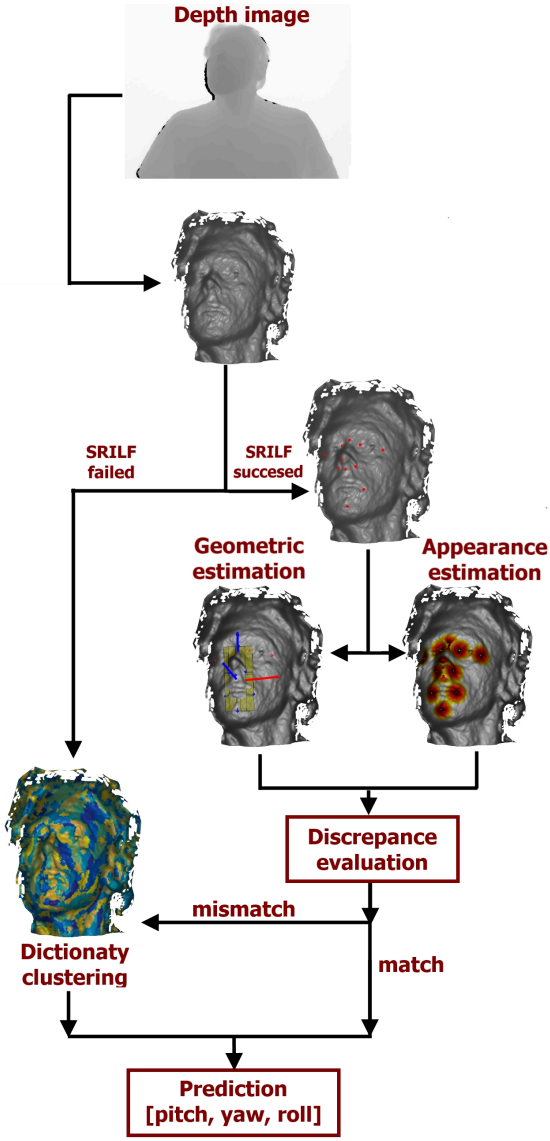

Figure 3.1: Block diagram of the proposed head pose estimation method

3.3 Proposed system

A block diagram of the proposed system is shown in Fig. 3.1. We start by approximately isolating the head region using clustering and use the obtained result to build a 3D mesh $\mathcal{M}$ that contains the head and a variable part of the shoulders. Mesh $\mathcal{M}$ is fed to the SRILF algorithm [Sukno et al., 2015] with the aim to automatically detect 12 prominent facial landmarks. The SRILF algorithm performs both detection of the visible landmarks and estimation of potentially occluded landmarks. Thus, if successful, the algorithm always returns an estimate of the coordinates for all 12 targeted points. Landmark detection details are provided in Section 3.3.1. Once facial landmarks are available, we use two complementary approaches to estimate the head pose (Section 3.3.2). Firstly, we perform a least-squares estimation of the eye-line and frontal-plane of the face which provide straight-forward *geometric* estimates of the head pose. The second estimate is based on regression over local surface descriptors (appearance) centered at the landmark points. While these two estimates are conceptually quite different, in practice, we will see that in practice they produce similar results (Section 3.4.1).

In a vast majority of cases ($\sim 90\%$) the above steps are sufficient to accurately estimate the head pose. The remaining 10% of cases are especially challenging scans, typically due to *i*) very large rotations, with self-occlusion of large portions of the face, and/or *ii*) low quality scans due to imaging artifacts. In such cases, we use an alternative estimate of the head pose based on dictionary learning (Section 3.3.3). It should be emphasized that the system automatically chooses whether to use the landmark-based or dictionary-based estimates on a case-by-case basis, with the following rationale:

- If landmarks are accurately detected, their estimate of the head pose is more precise than the dictionary-based estimate.

- If the SRILF algorithm cannot produce a reliable estimate of

landmark positions, the dictionary-based estimate is the only one available.

- If both landmark-based estimates (geometric and local descriptor regression) do not coincide, it is very likely that landmarks have been incorrectly detected. Thus, dictionary-based estimate should be used.

3.3.1 3D Landmark Detection

We use Shape Regression with Incomplete Local Features (SRILF) [Sukno et al., 2015] to locate the following 12 facial landmarks: inner and outer eye corners, nose corners, mouth corners, nose root, nose tip and chin tip. The SRILF algorithm combines the response from local feature detectors for each of the targeted landmarks with statistical constraints that ensure the plausibility of landmark positions on a global basis. The algorithm has three components: 1) selection of candidates through local feature detection; 2) partial set matching to infer possibly missing landmarks; 3) combinatorial search, which integrates the other two components.

Selection of candidates

The selection of candidates is performed independently for each targeted landmark. Given a mesh $\mathcal{M}$ and a landmark $\mathbf{x}_\ell$ to be targeted, a similarity score $s_\ell(\mathbf{v})$ is computed for every vertex $\mathbf{v} \in \mathcal{M}$; the set of *candidates* $\mathcal{C}_\ell$ for landmark $\mathbf{x}_\ell$ are the ϱ_ℓ highest scoring vertices:

$$\mathcal{C}_\ell = \{\mathbf{v} \in \mathcal{M} \,|\, \mathcal{O}(s_\ell(\mathbf{v})) \leq \varrho_\ell\} \tag{3.1}$$

where $\mathcal{O}()$ is the (descending) order function. The score $s_\ell(\mathbf{v})$ is based on the similarity of local surface descriptors with respect to a descriptor template derived at training time. The SRILF

implementation currently available[1] uses Asymmetry Pattern Shape Contexts [Sukno et al., 2014] as local descriptors.

As in many other algorithms, it is expected that one of these candidates will be close enough to the correct position of the landmark. Nonetheless, the number of false positives (i.e. vertices that produce high similarity scores even though they are far from the correct landmark location) can change considerably for different landmarks, as well as from one facial scan to another, making it difficult to choose the number of candidates that should be retained.

While many approaches try to retain large numbers of candidates to make sure that at least one will be reasonably close to the desired landmark position, SRILF determines the number of candidates as an upper outlier threshold from the distribution of false positives over a training set. This implies that, in the vast majority of cases, a candidate that is close enough to the target landmark will be detected, but a small proportion will be missed. Hence, for each targeted landmark there will be an initial set of candidates that may or may not contain a suitable solution and we need to match our set of target landmarks to a set of candidates that is potentially incomplete. This is analogous to the point-matching problem found in algorithms that search for correspondences. However, the human face is a non-rigid object and these point-matching algorithms are typically restricted to rigid transformations.

Partial set matching

The second component of the algorithm aims at dealing with the above problem. Based on the priors encoded in a statistical shape model, it uses a subset of the landmarks (i.e. those with suitable candidates) to infer the most likely position of the ones that are missing.

Let $\mathbf{x} = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_L, y_L, z_L)^T$ be a shape vector, constructed by concatenating the coordinates of the L targeted

[1] http://fsukno.atspace.eu/Data.htm#SRILF_3dFL

landmarks in 3D, and let $\overline{\mathbf{x}}$, $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ be the mean shape, eigenvector and eigenvalue matrices, respectively. Given a shape for which we only know part of its landmarks, we could split it in the known (or fixed) part $\mathbf{x}^f$ and the unknown (to infer or guess) part $\mathbf{x}^g$. Thus, our objective is to infer the coordinates of landmarks $\mathbf{x}^g$ so that the probability that the resulting shape complies with the PCA model is maximized, ideally without modifying the coordinates in $\mathbf{x}^f$.

Let $Pr(\mathbf{x})$ be the probability that shape $\mathbf{x}$ complies with the model. Assuming that $Pr(\mathbf{x})$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ in PCA-space, this probability is proportional to the negative exponential of the Mahalanobis distance and it can be shown [Sukno et al., 2015] that maximization of $Pr(\mathbf{x})$ with respect to $\mathbf{x}^g$ yields:

$$\mathbf{x}^g = \overline{\mathbf{x}}^g - (\mathbf{\Psi}^{gg})^{-1}\mathbf{\Psi}^{gf}\left(\mathbf{x}^f - \overline{\mathbf{x}}^f\right) \tag{3.2}$$

where $\mathbf{\Psi}^{gg} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^g)^T$, $\mathbf{\Psi}^{gf} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^f)^T$ and $\mathbf{\Phi}$ is split in $\mathbf{\Phi}^f$ and $\mathbf{\Phi}^g$ according to $\mathbf{x}^f$ and $\mathbf{x}^g$ (see [Sukno et al., 2015]).

Combinatorial search

The third component of the algorithm integrates the two previous steps into a combinatorial search. It consists of analyzing subsets of candidates and completing the missing information by inferring the coordinates that maximize the probability of a deformable shape model.

Formally, let $\mathcal{F}$ and $\mathcal{G}$ be the sets of fixed and to-infer coordinates, respectively, with $\mathcal{F} \cap \mathcal{G} = \varnothing$ and $\mathcal{F} \cup \mathcal{G} = \{1, 2, \ldots, 3L\}$. The goal of the combinatorial search is to dynamically choose the splitting into $\mathcal{F}$ and $\mathcal{G}$ to minimize the localization error:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} \tag{3.3}$$

where $\mathbf{x}$ are the *true* landmark coordinates and $\hat{\mathbf{x}}$ is the algorithm's estimate. The key concept here is that only the coordinates in $\mathcal{F}$ will

be based on image evidence (e.g. the candidates) and the rest will be treated as *missing data*. Thus, $\hat{\mathbf{x}}^g$ will be obtained by inference and it can be expressed as a function of $\hat{\mathbf{x}}^f$, making more apparent that the minimization looks for the optimal subset $\mathcal{F}$:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x}^f - \hat{\mathbf{x}}^f\|^2 + \|\mathbf{x}^g - f(\hat{\mathbf{x}}^f)\|^2\} \tag{3.4}$$

with $f(\hat{\mathbf{x}}^f)$ as defined in Eq. 3.2. Because the true coordinates $\mathbf{x}$ are unknown, we cannot explicitly compute the above errors and need an indirect estimate instead. The SRILF algorithm does this by minimizing (subject to statistical plausibility):

$$\underset{\mathcal{F}}{\operatorname{argmin}} \left(-|\mathcal{F}| - \exp\left(-\sum_{\ell \in \mathcal{F}} \min_{c \in \mathcal{C}_\ell} \|\hat{\mathbf{x}}_\ell - c\|^2 \right) \right) \tag{3.5}$$

where $\mathcal{C}_\ell$ is the set of candidates for the ℓ-*th* landmark $\hat{\mathbf{x}}_\ell$. Intuitively, Eq. 3.5 can be understood by noticing that the main component of the cost is the cardinality of $\mathcal{F}$, i.e. the number of landmarks that can be successfully included in $\hat{\mathbf{x}}^f$ while keeping the shape statistically plausible. Upon equality of $|\mathcal{F}|$ the cost function increases with the distance from $\hat{\mathbf{x}}$ to the nearest candidate per landmark. These distances to the nearest candidates have a different meaning for fixed and inferred landmarks and help understand the way the algorithm works.

Fixed landmarks $\{\hat{\mathbf{x}}_\ell\}_{\ell \in \mathcal{F}}$ are directly sampled from candidates to guide the combinatorial search. Thus, their nearest candidates are known beforehand and their distance to them is just the reconstruction error of the statistical shape model. For the remaining landmarks, $\{\hat{\mathbf{x}}_\ell\}_{\ell \in \mathcal{G}}$, positions are statistically inferred from Eq. 3.2 independently from their candidate sets (Fig. 3.3). It would be expected that better predictions generate inferred landmarks that are closer to their corresponding candidates, resulting in lower cost values.

The minimization in Eq. 3.5 is addressed by testing all possible combinations of 4 candidates, which constitute the initial $\hat{\mathbf{x}}^f$. The

shape is completed by inference of $\hat{\mathbf{x}}^g$ from Eq. 3.5 and is checked against the statistical constraints of the shape model. As long as the generated shape is statistically plausible, candidates are added to $\hat{\mathbf{x}}^f$ from the remaining landmarks in a sequential forward selection strategy looking for the maximum possible $|\mathcal{F}|$.

An important aspect of the splitting between $\mathcal{F}$ and $\mathcal{G}$ is that it inherently provides tolerance to distorted or missing data (occlusions). Notice that there is no prior assumption regarding what landmarks can be in $\mathcal{F}$ or $\mathcal{G}$ nor the cardinality of the two sets and the splitting is performed dynamically on a case by case basis. This is an advantage in applications such as head-pose estimation with sensors like Kinect, which capture depth information from a single view. Under large head rotations, the generated depth maps will have large parts of the face missing due to self-occlusions and it is crucial to be able to exploit partial information.

3.3.2 Landmark-based pose estimation

Once facial landmarks are extracted, we can estimate head pose, represented by three Euler angles also called as *yaw* (ϕ), *pitch* (θ) and *roll* (ψ) angles. Pitch (nodding) is the rotation around the horizontal axis, which in our case is the X axis. Yaw (shaking) is the rotation around the vertical axis of the body (Y axis). Roll (tilting) is the rotation around the axis perpendicular to two previous axes. In our case, this is the Z axis, which is perpendicular to the camera (Fig. 3.2).

We derive two different landmark-based pose estimates, a geometric estimate and an appearance estimate:

Geometric estimate

It is based on least-squares estimates of simple geometric entities that can approximately describe the head pose. Specifically, we estimate the eye-line to determine the *roll* angle and a frontal-face plane for *yaw* and *pitch*.

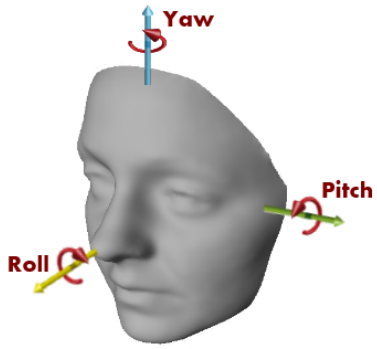

Figure 3.2: Orientation of the head in terms of pitch, roll, and yaw angles

The four landmarks of inner and outer eye corners are used to build the eye-line. Firstly, the eye-line is projected into the XY plane, where it can be expressed as a linear equation of two variables $y = mx + b$. The roll angle is calculated as $\psi = \tan^{-1}(m)$.

The remaining landmarks, except the nose tip, are used to estimate a plane that will be a good approximation of the frontal-face region (see Fig. 3.1). Let the normal vector to this plane be $\mathbf{n} = [x_n, y_n, z_n]$. Due to the fact that angles can be obtained by rotations about its principal axes, we can compute the *yaw* and *pitch* angles as: $\phi = \tan^{-1}\left(x_n/z_n\right)$, $\theta = \tan^{-1}\left(y_n/z_n\right)$.

Appearance estimate

It is based on regression over the local appearance around landmark points. Specifically, for each detected landmark $\hat{\mathbf{x}}_\ell$ we compute a local surface descriptor $d(\hat{\mathbf{x}}_\ell)$ that will be the input to a multi-linear regressor $\mathbf{A}_\ell$ yielding an estimate for ϕ, θ and ψ. Thus, differently from the geometric estimate, the appearance estimate requires a training set to derive the regressors.

We use 3D Shape Contexts (3DSC) [Frome et al., 2004] as local descriptors, slightly modified to increase their sensitivity to

viewpoint and robustness to noise. 3DSC are based on a spherical histogram computed on a neighbourhood of the interest point (in our case, landmark locations) and have been shown to perform well as descriptors of the facial surface [Sukno et al., 2012]. Similarly to other popular descriptors 3D geometry [Johnson and Hebert, 1999, Tombari et al., 2010, Rusu et al., 2009], 3DSC use the surface normal at the interest point to appropriately orient the reference system of the local neighbourhood, aiming for rotational invariance[2]. Because our objective is to identify viewpoint, such normal-based orientation is not convenient, hence we will orient the reference systems of all local neighbourhoods based on the normal to the camera sensor. This choice avoids also the computation of surface normals, which are known to be especially sensitive to noise [Papazov et al., 2015, Tombari et al., 2010].

Notice that, in principle, we will produce L different estimates for each angle (i.e. one per landmark). However, because of the potential presence of occlusions, it is not guaranteed that all estimated landmarks will actually lie on the mesh surface.[3] Indeed, when parts of the facial surface are missing, it is possible that some landmarks $\ell \in \mathcal{G}$ are estimated relatively far from the mesh $\mathcal{M}$, i.e. they are inferred in the position where we would statistically expect them to be, despite no surface has been captured there (Fig. 3.3).

Therefore, we use the indicator function $\mathbb{1}(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon)$ to filter out the estimates of landmarks that are estimated off the surface and produce our final appearance estimate as the average of the remaining ones:

$$(\phi, \theta, \psi)^T = \frac{\sum_k \mathbb{1}\Big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\Big)\mathbf{A}_\ell d(\hat{\mathbf{x}}_\ell)}{\sum_k \mathbb{1}\Big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\Big)} \tag{3.6}$$

where the distance from $\hat{\mathbf{x}}(\ell_k)$ to $\mathcal{M}$ is computed as the distance to

[2]Such invariance, however, is only partially achieved in 3DSC since the orientation of the surface normal still leaves one degree of freedom undefined (the sphere's azimuth [Sukno et al., 2013])

[3]We consider that a landmark is *on the surface* when its distance to it is relatively small as compared to the mesh resolution.

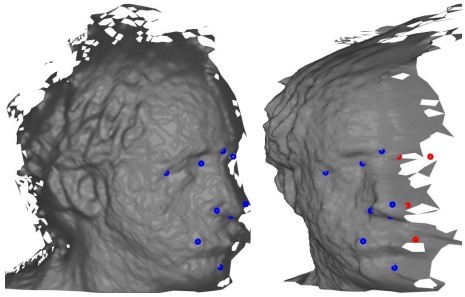

Figure 3.3: Positions of the landmarks estimated automatically by SRILF in a head scan showing large yaw rotation. Two views of the same scan are provided: the original view (as seen from the camera) is shown to the left and a rotated view (to simulate a *frontal shot*) is shown to the right. Landmarks lying on the surface are indicated in blue color, while those off-the-surface (estimated by inference) are displayed in red.

the nearest mesh vertex:

$$\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| = \min_{v_j \in \mathcal{M}} \|\hat{\mathbf{x}}_\ell - v_j\| \tag{3.7}$$

3.3.3 Dictionary-based pose estimation

As mentioned before, for some small percentage of scans SRILF has difficulties to correctly locate the facial landmarks and, thus, the approaches described in Section 3.3.2 are not applicable to estimate the corresponding head pose. Typically, these are especially challenging scans, with big rotations, large parts of the head self-occluded and/or very poor quality. These difficulties, together with the failure of a state-of-the-art landmarker as SRILF, suggest the need for a landmark-free approach to tackle these scans. Thus, we employ an alternative dictionary-based strategy for the scenario where no explicit vertex-landmarks correspondences are found.

Inspired by the success of Bag-of-Words approaches in 3D shape retrieval [Wang et al., 2012b], we represent each scan as a set of

descriptors $\mathcal{D} = \{d(\mathbf{x}_1), d(\mathbf{x}_2), ..., d(\mathbf{x}_N)\}$ extracted vertices $\mathbf{x}_n = \{x_n, y_n, z_n\}$ randomly sampled on the mesh $\mathcal{M}$. Concretely, we use again 3DSC descriptors with fixed orientation coinciding with the camera axis (Section 3.3.2) and a random sampling over the mesh with density of $7mm^{-1}$.

Given the sets $\mathcal{D}$ obtained from all the training scans, we use k-means clustering to learn a dictionary $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_K\}$ of 3D descriptors, where each $\mathbf{z_k}$ is a particular centroid and K is the total number of clusters considered. Intuitively, these clusters will represent different shapes typically appearing in face scans (e.g. nose tip, cheeks, eyes corners, etc.). This dictionary $\mathbf{Z}$ is then used to encode each 3D mesh as a vector $\mathbf{h} \in \mathcal{R}^K$ representing the frequency of each cluster $\mathbf{z}_k$ in the scan. For this purpose, we employ a Soft-Assignment approach [Lian et al., 2013], where each descriptor $\mathbf{x}_n$ is encoded as:

$$h_n^k = \frac{\exp(-||d(\mathbf{x}_n) - \mathbf{z}_k||^2)}{\sum_{j=1}^{K} \exp(-||d(\mathbf{x}_n) - \mathbf{z}_j||^2)}, \tag{3.8}$$

and the final vector representation is computed using a sum-pooling procedure as: $\mathbf{h} = \sum_{k=1}^{K} \mathbf{h}^k$. Finally, vectors $\mathbf{h}$ for all the training scans are used to train three different Least-Squares linear regressors for yaw, pitch and roll angles.

3.4 Experiments

We used the recently published SASE 3D head-pose database [Lüsi et al., 2016b], to assess the performance of our approach. The data in SASE has been acquired with Microsoft Kinect 2 camera and contains RGB and depth images in pairs. The entire database includes 50 subjects (32 male and 18 female) in the range of 7-35 years old, with more than 600 frames per subject. For each person, a large sample of head poses are included, with wide range of yaw, pitch and roll variations.

For the Head Pose Challenge [Lüsi et al., 2017] organized at the International Conference on Automatic Face and Gesture Recognition (FG 2017), the SASE data has been divided in three sets: *Training* (comprising 28 subjects with a total of $\sim$ 17K images), *Validation* (12 subjects, $\sim$ 7K images) and *Test* (10 subjects, $\sim$ 6K images). Only the *Training* data was made available to challenge participants in order to investigate the performance of their algorithms prior to the final evaluation phase. Therefore, we first present detailed results of our system using only Training data (Section 3.4.1) and use them to choose the parameters that will be used for Validation and Test sets (Section 3.4.2). Notice that, although the SASE data contains both RGB and depth images, we only used depth data in order to comply with the participation requirements of the Head Pose Challenge.

3.4.1 Training

We started by splitting the training data into two subsets: *Development* and *Pre-test*. The Development set was composed of 840 images, by randomly choosing 30 images from each of the 28 training subjects. This set was used to train the landmarking algorithm and the dictionary-based estimate. The remaining images ($\sim$ 16K) were used as a preliminary test-set to asses system's performance and validate system parameters.

As explained in Section 3.3, our system combines three different methods to estimate head pose: two of them are based on landmarks (geometric and appearance estimates) and the third one is dictionary-based. The three methods were developed independently and each has its advantages and shortcomings. Table 3.1 shows the results of each method applied separately on the entire Pre-test set. We can see that, as anticipated, landmark-based estimates are more accurate than dictionary-based estimates. On the other hand, for about 9% of the scans it was not possible to detect landmarks and only the dictionary-based estimates are available. Notice the

Approach	**Pitch (°)**	**Yaw (°)**	**Roll (°)**	**% scans**
Scans with landmarks successfully detected				
Landmark-based geometric	6.34	6.42	7.33	90.8%
Landmark-based appearance	6.17	6.04	5.57	90.8%
Landmark-based combined	5.50	5.44	5.28	90.8%
Dictionary-based	8.74	8.06	5.89	90.8%
Scans with landmarks not detected				
Dictionary-based	14.74	14.10	9.83	9.2%
All scans				
Dictionary-based	9.29	8.61	6.25	100%
Combination	**6.33**	**6.10**	**5.46**	**100%**

Table 3.1: Average pose estimation errors on the Training part of the SASE Database

comparatively large errors of the estimates in these scans, which confirm that these are especially challenging cases.

Within landmark-based methods, estimates based on appearance were slightly more accurate than geometric ones. However, we note that *i*) the geometric-based estimate is training-free while the appearance-based one requires a learning stage[4]; *ii*) combination of both estimates (by averaging) produced better results than each of them individually.

The dictionary-based approach was not as accurate as the landmark-based ones, but it was able to produce estimates in all cases. As explained in Section 3.3.3, this method also requires learning, which was performed on our development set, fixing the number of clusters to $K = 500$.

[4]For the experiments reported in Table 3.1, the appearance-based estimate was tested in a 10-fold cross-validation setting.

The last line of Table 3.1 shows the final results of the system, obtained by combining the three methods. As indicated in Fig. 3.1, landmark-based estimates were preferred over dictionary ones. However, if no landmarks were available or if geometric and appearance landmark-based estimates did not match, we used the dictionary approach. The rationale behind checking landmark-based estimates for agreement is that, if landmark are accurately located then both geometric and appearance estimates should produce similar results. On the other hand, if landmarks are located at incorrect positions, the estimates will also be incorrect but are unlikely to coincide among them, given the different nature of the estimators.

Therefore, given a head scan with geometric estimates $(\phi_G, \theta_G, \psi_G)$ and appearance estimates $(\phi_A, \theta_A, \psi_A)$, the system will use these estimates if and only if:

$$|\phi_G - \phi_A| + |\theta_G - \theta_A| + |\psi_G - \psi_A| < \tau \tag{3.9}$$

Otherwise, the dictionary-based approach is used. Fig. 3.4 shows the variation of the estimation errors of the landmark- and dictionary-based approaches for different values of τ. The errors are displayed for each angle taking into account only the scans for which landmark-based estimates failed to comply with Eq. 3.9. It can be seen that scans with larger differences between geometric and appearance estimates have larger pose estimation errors. Errors increase steadily $\forall\tau$ for landmark-based estimates and partially for the dictionary-based approach (approximately up to $\tau \leq 50$). The second observation from Fig. 3.4 is that, as expected, for large differences between geometric and appearance estimates the dictionary-based approach is more accurate. For the experiments reported in this Chapter, we have adopted a conservative value of $\tau = 50$ where Fig. 3.4 shows lower errors of the dictionary-based approach for all three estimated angles. As indicated by the black line in the plot, this represents replacing the landmark-based estimates by the dictionary ones in approximately 15% of the cases.

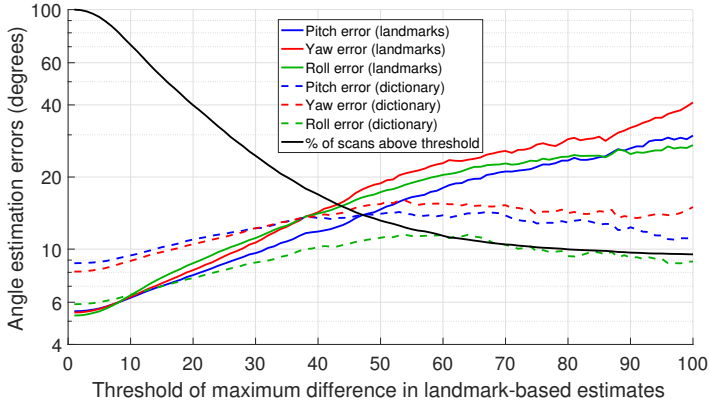

Figure 3.4: Variation of estimation errors for landmark- and dictionary-based estimates as a function of the difference between geometric and appearance estimates. The black solid line shows the percentage of scans for which the difference in Eq. 3.9 is above the value indicated in the horizontal axis. For those scans, the blue, red and green curves show the angle estimation errors based on landmarks (solid lines) and dictionary (dashed lines).

3.4.2 Validation and Test

Once we trained the necessary models and set the system parameters as described in the previous section, we submitted our estimates of head poses in the Validation and Test sets to the Head Pose Challenge. Table 3.2 summarizes our results, with which we obtained the first place in the challenge. It can be seen that the results at this stage were not too different from those obtained in training, indicating that there was not much over-fitting. Notice that, for the Test set, we can only provide the overall score (sum of average errors for all angles) since the ground truth for this part of the database was not made available by the challenge organizers.

Table 3.3 shows additional details about the performance of the proposed system in the validation and test sets. The average processing time reported correspond to tests on an Intel

Subset	Pitch	Yaw	Roll	Sum
Training	6.33	6.10	5.46	17.89
Validation	7.82	6.65	5.39	19.86
Test	n/a	n/a	n/a	19.02

Table 3.2: Average pose estimation errors on the SASE Database

Overall statistics	
Total number of scans	13,885
Average processing time	8.42 s
Automatic landmarks	
Successfully detected	91.1 %
Average detection time	5.77 s
Landmark-based pose estimates	
Successfully computed	91.1 %
Agreement within $\tau = 50$	86.8 %
Average processing time	6.08 s
Dictionary-based pose estimates	
Computed for	13.2 %
Average processing time	3.29 s

Table 3.3: Detailed information about the performance of the proposed system on the Validation and Test sets

i7-770 processor at 3.4 GHz with 16 Gb or RAM. Most of the implementation was done in Matlab and it has not been optimized for speed. Full code of the system used to produce the reported results is publicly available.[5]

[5]https://github.com/DmytroDerkach/CMTech

3.4.3 Comparison to other methods

Table 3.4 shows the results reported by most relevant previous works addressing head pose estimation based on 3D data. Together with the estimation errors for each angle, we indicate whether the corresponding methods use tracking, RGB and/or depth data. Moreover, we show the specific database(s) used for testing in each case. As explained in Section 3.2, most methods use temporal information to speed-up processing but also to avoid the need for initialization at every frame. This can considerably simplify the problem if sequences start from near-to-frontal shots, as is the case in most datasets used for head pose evaluation. However, this assumption do not need to be fulffiled in real-scenarios. Among the four methods listed in Table 3.4 that do not rely on tracking, only one exclusively uses depth information (as our system) and the other three methods use both depth and RGB data.

Comparisons with our work using results depicted in Table 3.4 are difficult and rather indirect considering the diversity of datasets and experimental setups that were used in the cited works. However, our results compares favorably to the best performing methods in the literature only relying on depth information. Moreover, performance of the proposed system is comparable to some algorithms that also use tracking for pose estimation. Finally, a detailed analysis of our results reveals that our average estimation errors are strongly influenced by the presence of outliers; e.g. our median absolute estimation errors were approximately 3.5 degrees per angle, considerably lower than the average absolute errors reported in Section 3.4. Analysis of these outlier cases revealed that they were typically scans with large rotation angles where the face was positioned quite oblique to the camera axis and the sensor could not capture it with sufficient quality.

Method	Errors			Tracking	Domain	Data
	ϕ	θ	ψ			
Valle [Valle et al., 2016]	12.6	-	-	No	RGB+ depth	AFLW/ AFW
Wang [Wang et al., 2013a]	8.8	8.5	7.4	No	RGB+ depth	Biwi Kinect
Li [Li and Pedrycz, 2014]	8	12	4	No	depth	FRGC v2.0
Papazov [Papazov et al., 2015]	2.5	1.8	2.9	No	RGB+ depth	Biwi Kinect
	3.0	2.5	3.8	Yes		
Meyer [Meyer et al., 2015]	2.1	2.1	2.4	Yes	depth	Biwi Kinect
	2.9	2.3	-	Yes	depth	ETH
Padeleris [Padeleris et al., 2012]	2.4	3.0	2.8	Yes	depth	Biwi Kinect
Martin [Martin et al., 2014]	2.6	2.5	3.6	Yes	depth	Own
Baltrusaitis [Baltrušaitis et al., 2012]	6.3	5.1	11.3	Yes	RGB+ depth	Biwi Kinect
	3.0	3.8	2.1	Yes		BU
	2.9	3.1	3.2	Yes		ICT-3DHP
Tulyakov [Tulyakov et al., 2014]	4.7	7.6	5.3	Yes	RGB+ depth	Dali 3DHP
Fanelli [Fanelli et al., 2011]	8.9	8.5	7.9	Yes	depth	Biwi Kinect

Table 3.4: Average angular errors(in degrees) for different existing head pose estimation algorithms

3.5 Conclusions

In this Chapter we present an approach for accurate head pose estimation from a single depth frame of consumer RGB-D cameras, such as Kinect 2. In contrast to most existing approaches, we base our system in the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based pose estimators. A key aspect of the proposed system is the use of state of the art landmark localization with no need for initialization and tolerance to occlusions or missing data. Our system is complemented with a secondary pose estimator based purely on patches sampled randomly on the head region to account for potential failures of the landmark-based estimation.

We evaluated our system on the SASE database, which consists of $\sim$ 30K frames from 50 subjects. We obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge. Our experiments also confirmed the initial hypothesis that the landmark-based estimates would be more accurate than correspondence-free approaches, such as the dictionary-based one that was adopted. Landmark-based estimates were successfully produced for $\sim$ 90% of cases and the remaining ones were tackled by the dictionary-based approach. Our results compare well with those reported in the related literature, especially considering the added difficulty of not using tracking and RGB data to produce our estimates.

Chapter 4

TENSOR DECOMPOSITION AND NON-LINEAR MANIFOLD MODELING FOR 3D HEAD POSE ESTIMATION

Adapted from: D. Derkach, F.M. Sukno, "Tensor Decomposition and Non-linear Manifold Modeling for 3D Head Pose Estimation", *International Journal of Computer Vision*;

D. Derkach, A. Ruiz, F.M. Sukno, "3D Head Pose Estimation Using Tensor Decomposition and Non-linear Manifold Modeling", *International Conference on 3D Vision*, 2018, pages 505-513, DOI: 10.1109/3DV.2018.00064

Abstract

Head pose estimation is a challenging computer vision problem with important applications in different scenarios such as human-computer interaction or face recognition. In this Chapter, we present an algorithm for 3D head pose estimation using only depth information collected from Kinect sensors. A key feature of the proposed approach is that it allows modeling the underlying 3D manifold that results from the combination of pitch, yaw and roll rotations. To do so, we use tensor decomposition to generate separate subspaces for each variation factor and show that each of them has a clear structure that can be modeled with cosine functions from a unique shared parameter per angle. Such representation provides a deep understanding of data behavior and angle estimations can be performed by optimizing combination of these cosine functions. We evaluate our approach on two publicly available databases, and achieve top state-of-the-art performance.

4.1 Introduction

Head pose estimation is a relevant problem for several computer vision applications, including human-computer interaction, video conferencing, face recognition and facial motion analysis [Wang et al., 2018]. Head pose estimation has traditionally been performed on 2D images, but advances in 3D acquisition systems have led to a growing interest in methods that operate on 3D data [Seemann et al., 2004]. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust. Therefore, in this Chapter we focus on head pose estimation from 3D data.

The goal of head pose estimation is to predict the relative orientation between the camera and a 3D mesh of the target head. This orientation is usually represented by three angles: rotation around vertical axis (yaw angle), around lateral axis (pitch angle), and around longitudinal axis (roll angle). Despite the fact that standard features used to represent 3D meshes lie in high-dimensional spaces, a key observation to solve this problem is that the aforementioned angles define a lower-dimensional manifold with only 3 degrees of freedom. This fact makes tensor decomposition and manifold learning appealing frameworks for the estimation of the orientation parameters. In particular, factorization methods such as multi-linear decomposition [De Lathauwer et al., 2000, Wang et al., 2017b], are able to separate the variations produced by the different factors (i.e. angles) into separate subspaces, thus obtaining specific parametrizations for each of them. On the other hand, manifold learning [Wang et al., 2017a] can be used to find the low-dimensional manifold structure defined by the orientation angles.

In this context, previous works have attempted to use the described frameworks for head pose estimation. Concretely, methods such as Isomap [Raytchev et al., 2004] or Local Linear Embedding [Fu and Huang, 2006] have been explored in order to learn the underlying manifold structure defined by the orientation

parameters. Even though the cited methods are able to learn generic low-dimensional data representations, the resulting manifold is only defined implicitly and, therefore, it is difficult to introduce specific constraints to model the inherent structure defined by rotation variations.

In order to address this limitation, we propose a novel approach to learn the manifold defined by 3D rotations. In particular, our method is able to explicitly model its underlying structure with an analytic form which takes into account the specific constraints imposed by orientation variations. For this purpose, we use multi-linear decomposition over 3D descriptors in order to split the pose variation factors (i.e. yaw, pitch and roll) and obtain a set of subspaces whose coefficients are governed by an unique parameter. These coefficients define a continuous curve in each of the sub-spaces that corresponds to the head pose variation along one of the rotation angles. We further show that these curves can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. Thus, we introduce a minimization framework for pose estimation based on tensor decomposition constrained by trigonometric functions so that the solutions obtained are always compatible with the underlying rotation manifold.

We start by investigating the structure of the subspace obtained by multi-linear decomposition when such subspace corresponds to one rotation angle. It is performed by using 2D images that capture rotations of simple objects along the vertical axis. We show that the obtained coefficients, indeed, describe the rotation effects, thus they can be modeled by a trigonometric function. Then, we generalize it to 3D rotations in any of the three axes and demonstrate its usefulness by applying it to head pose estimation. Preliminar results of this approach were presented in [Derkach et al., 2018]. We perform experiments over two large and publicly available 3D face corpora: the SASE [Lüsi et al., 2016b] and BIWI databases [Fanelli et al., 2013]. In contrast to previous work [Derkach et al., 2018], we introduce additional feature type, and show that the proposed

framework can achieve state-of-the-art performance for head-pose estimation using different type of features.

The rest of the chapter is organized as follows. Section 4.2 introduces a brief review of the existing approaches for head pose estimation. In Section 4.3 we provide the required background on tensor theory and Section 4.4 details the general idea of the proposed method using multi-linear decomposition and manifol modeling framework. Then in Section 4.5 we show a simple example based on the images with rotation about only one vertical axis. Further in Section 4.6 we show how the presented framework can be applied for 3D head pose estimation. All experimental and features extraction details with obtained results are presented in this section. Finally, Section 4.7 concludes the chapter.

4.2 Related work

4.2.1 Manifold-based methods

Many methods have considered the model the underlying manifold structure of head pose variations [Wang et al., 2017a]. The main idea behind these methods is that, regardless of the dimensionality of the input features representing the mesh, there should be at most 3 degrees of freedom for head pose variation, thus defining a 3D manifold [Raytchev et al., 2004]. However, in general, this manifold is embedded non-linearly in the ambient space defined by the features, which has led researchers to explore non-linear manifold learning methods such as Locally Linear Embedding [Fu and Huang, 2006], Isomap [Raytchev et al., 2004], Synchronized Submanifold Embedding [Zhu et al., 2014], Homeomorphic Manifold Analysis [Peng et al., 2014], Neighborhood Preserving Embedding or Locality Preserving Projection [BenAbdelkader, 2010] for head pose estimation from 2D images.

An interesting possibility to enhance the embedding results is to adopt a supervised strategy and use head pose labels in order

to learn the manifold structure. For example, in [Balasubramanian et al., 2007], the authors presented a Biased Manifold Embedding (BME) framework in which the distance metric between features is modified so that heads under similar poses are brought closer to each other than they would be under the unbiased (unsupervised) case. Similarly, Wang et al. [Wang and Song, 2014] consider head-pose information to constrain the distances between data points and present a regression variant of Fisher Discriminant Analysis (FDA), which they call supervised neighborhood-based FDA. An alternative approach is followed by Benabdelkader [BenAbdelkader, 2010], who firstly apply unsupervised manifold learning methods and then employ the head pose information to train regressors in the resulting low-dimensional manifolds.

Liu et al. [Liu et al., 2010] argue that a single manifold is not enough for head pose estimation and that appearance variations such as changes in identity, scale and illumination make it necessary the use of multiple different manifolds to model pose parameters. Thus, authors presented a clustering method to construct multiple manifolds, each of which characterizes the underlying subspace of some subjects. Peng et al. [Peng et al., 2014] also learn multiple manifolds; they use Homeomorphic Manifold Analysis to build a separate manifold for each subject and learn non-linear mappings to relate each subject-manifold with a common pose-manifold whose topology is predefined as a unit circle or sphere (for addressing rotations about one or two axes, respectively).

The most similar work to ours is probably the one from [Takallou and Kasaei, 2014], who learn a non-linear tensor model based on multi-linear decomposition for head pose estimation from 2D images. They build a three-way tensor to account for identity, pose and pixels information, targeting only yaw rotations. During training, they find individual-dependent mappings between each training pose and a unified pose manifold based on tensor decomposition. At test time, each query image is projected into pose and identity subspaces, which results in as many pose coefficients as identities in the training set.

The final pose estimate is obtained by validating the available pose coefficients in terms of compliance with the unified pose manifold (e.g. inversely to the distance to training samples).

In contrast to our work, all of the above methods use 2D images and most of them do not target rotations about the three spacial axes, they consider rotations about only one or two axis. Moreover, none of them provides an analytic formulation for the pose manifolds.

4.2.2 3D methods review

Head pose estimation has traditionally been performed on 2D images. Cheng et al. [Chen et al., 2016], in their work, proposed a non-linear regression method based on gradient-based features for the estimation of head pose from extremely low resolution images. Also, Lee et al. [Lee et al., 2015] showed a method based on the random projection forests algorithm using only 2D images. Some of the approaches have used neural networks for solving head pose estimation problem [Venturelli et al., 2016, Lathuilière et al., 2017]. For example, Liu et al. [Liu et al., 2016] presented a method for head pose estimation using convolutional neutral network (CNN). As the input of the network they use a head RGB image, and then CNN was trained to learn head features and solve the regression problem. The work of Ahn et al. [Ahn et al., 2014] used a CNN method to learn a mapping function between visual appearance and head pose angles. A common thing for all of the approaches based on the neural networks is that they use only 2D images.

But recent advances in 3D acquisition systems have led to a growing interest in methods that operate on 3D data. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust.

An important distinction between different approaches is the type of input data that is used. Firstly, very few methods use only depth information, typically relying on curvatures, symmetry planes

or most salient facial landmarks, such as the nose tip [Breitenstein et al., 2008, Sun and Yin, 2008, Li and Pedrycz, 2014]. There are also other head pose estimation algorithms that rely on 3D data. Martin et al. [Martin et al., 2014], presented a method based on building a 3D head model with the iterative closest point (ICP) algorithm.

In contrast, a majority of head pose estimation algorithms working in 3D, use also RGB data as additional source of information, facilitating aspects such as face detection and estimation of fiducial points. In this category we find approaches based on the fusion of 2D and 3D features (e.g. SIFT, HOG) to train regressors [Wang et al., 2013a], template fitting, such as 3D Morphable Models [Ghiass et al., 2015, Yu et al., 2017], or depth features initialized by 2D face detection [Papazov et al., 2015].

Finally, it is also common to take advantage of temporal information for tracking the head pose across sequences of frames [Tulyakov et al., 2014, Barros et al., 2018, Tan et al., 2018], which considerably improves performance. However, tracking-based algorithms often benefit from the fact that test sequences usually start with nearly frontal head poses and their accuracy to detect initial head poses other than frontal is not clear. Thus, when comparing our results, we will focus on methods that provide estimation results on a per-frame bases, without tracking.

Interestingly, we see that previous methods targeting head pose estimation from 3D data have not taken advantage of the underlying manifold structure of 3D head rotations. In contrast, we take into account the structure of the manifold, also we present a method that is able to explicitly model its underlying structure with an analytic form which takes into account the specific constraints imposed by orientation variations.

4.3 Technical background: Tensor decomposition

In this section, we give a review of tensor decomposition methods, especially focusing on the higher order SVD (HOSVD) [Bergqvist and Larsson, 2010, Comon, 2014, Kolda and Bader, 2009]. In many scenarios, data can be naturally represented as multidimensional arrays and, therefore, it is beneficial to take into account its inherent structure in order to analyze it. For this purpose, the use of tensors is a natural solution. In particular, a tensor is also known as a n-way array or a n-mode matrix. Vectors and matrices can be considered as first and second order tensors, respectively. First of all, we will start by reviewing the standard SVD decomposition for second order tensors.

For matrix $A \in \mathbb{R}^{m \times n}$ we recall the SVD as being:

$$A = U\Sigma V^T = \sum_{k=1}^{r} \sigma_k u_k v_k^T = \sum_{k=1}^{r} \sigma_k u_k \otimes v_k \tag{4.1}$$

and for the elements a_{ij} of A we have

$$a_{ij} = \sum_{k=1}^{r} u_{ik}\sigma_{kk}v_{ik} \tag{4.2}$$

Here $\otimes$ denotes the tensor (or outer) product $x \otimes y \triangleq xy^T$; Σ is a diagonal $(r \times r)$ matrix with nonzero singular values of A (the square roots of the eigenvalues of A^TA) on its diagonal; u_k and v_k are the orthonormal columns of the matrix U $(m \times r)$ and V $(n \times r)$, respectively, with v_k being the eigenvectors of A^TA and $u_k = Av_k/\sigma_k$ [Bergqvist and Larsson, 2010].

The SVD is useful whenever we have a two-dimensional data set a_{ij}, which is naturally expressed in term of a matrix A (second order tensor). In the application of this Chapter we will deal with cases where the dimension is bigger than two, particularly is equal five

(fifth order tensor). The SVD may be generalized to higher order tensors (or multiway arrays).

Given $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_5}$, the decomposition of the fifth order tensor can be expressed as

$$\mathcal{T} = \sum_{J_1} \cdots \sum_{J_5} g_{J_1 J_2 \ldots J_5} u^{(1)}_{J_1} \otimes u^{(2)}_{J_2} \otimes \cdots \otimes u^{(5)}_{J_5} \tag{4.3}$$

or as a mode product [De Lathauwer et al., 2000]

$$\mathcal{T} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_5 U^{(5)} \tag{4.4}$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_5}$ is the core tensor and $U^{(n)} \in \mathbb{R}^{I_n \times J_n}$ – are the factor matrices (which are orthogonal) and can be thought of as the principal components in each mode. The graphic representation of the Higher Order SVD (3D) is shown in Fig. 4.1(a);

The n-mode product of a tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ *by a matrix* $U \in \mathbb{R}^{I_n \times J_n}$ *denoted by* $\mathcal{G} \times_n U$ *is an* $(J_1 \times J_2 \times \cdots \times J_{n-1} \times I_n \times J_{n+1} \times \cdots \times J_N)$*-tensor of which the entries are given by*

$$(\mathcal{G} \times_n U)_{j_1 j_2 \cdots j_{n-1} i_n j_{n+1} \cdots j_N} = \sum_{j_n} g_{j_1 j_2 \cdots j_{n-1} j_n j_{n+1} \cdots j_N} u_{i_n j_n} \tag{4.5}$$

In HOSVD, all matrices $U^{(n)}$ can be calculated by performing a matrix SVD on the $I_n \times (I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)$ matrix obtained by a flattering or unfolding of $\mathcal{T}$ [Bergqvist and Larsson, 2010, De Lathauwer et al., 2000].

The n-mode matricization (or unfolding) of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ *is denoted by* $\mathcal{T}_{(n)}$ *and arranges the n-mode fibers to be columns of the resulting matrix. Tensor element* $(i_1, i_2, \ldots, i_N)$ *maps to matrix element* (i_n, j)*, where*

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k; \quad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m \tag{4.6}$$

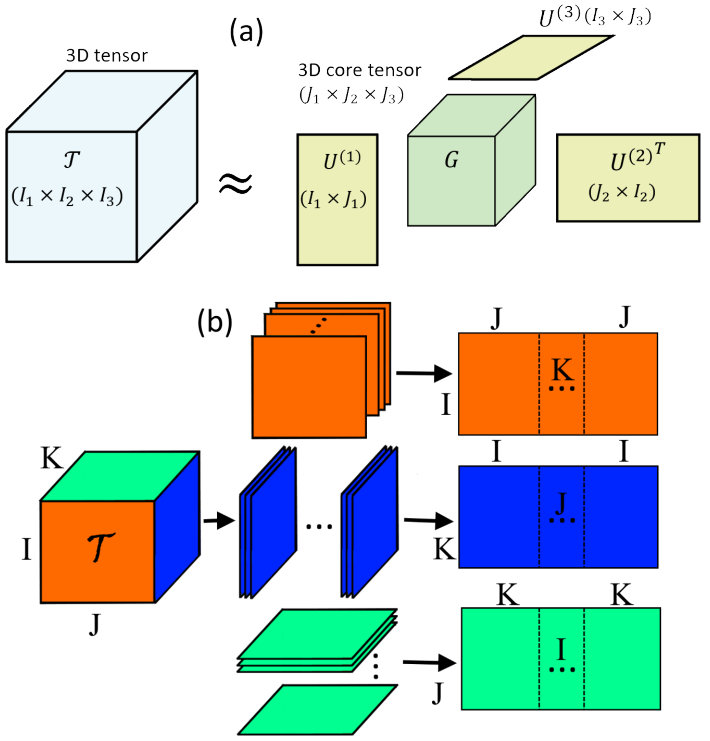

Figure 4.1: (a) Illustration of a 3D tensor decomposition. (b) Unfolding of the $(I \times J \times K)$-tensor $\mathcal{T}$ to the $(I \times JK)$-matrix, the $(J \times KI)$-matrix and the $(K \times IJ)$-matrix

An example of unfolding of the third order tensor $\mathcal{T}$ is shown in Fig. 4.1(b)

Since $U^{(n)}$ matrices are orthogonal, $\mathcal{G}$ from equation (4.4) is easily calculated from (4.7) and it is called the core tensor which shows the interactions of $U^{(n)}$ matrices – factor matrices [De Lathauwer et al., 2000].

$$\mathcal{G} = \mathcal{T} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \cdots \times_5 U^{(5)^T} \tag{4.7}$$

4.4 Proposed method

4.4.1 Multilinear decomposition and estimation of 3D rotations

In the following, we explain how the tensor decomposition framework described in Sec. 4.3 can be used to model data variations caused by rotations. Consider a training set composed by N samples $\mathbf{x}_n \in \mathbb{R}^{D_f}$. For instance, in head pose estimation, $\mathbf{x}_n$ refers to a D_f-dimensional vector representing a 3D-descriptor extracted from a mesh. Moreover, we assume that each $\mathbf{x}_n$ is labelled according to its identity plus 3 values defining its corresponding rotation angles (i.e yaw, pitch and roll). By discretizing these values into D_y, D_p and D_r bins respectively, we can represent the whole dataset as a 5-way tensor $\mathcal{T} \in \mathbb{R}^{N \times D_y \times D_p \times D_r \times D_f}$.

By using Eq. 4.3, we can decompose $\mathcal{T}$ as:

$$\mathcal{T} = \mathcal{G} \times U^{(id)} \times U^{(y)} \times U^{(p)} \times U^{(r)} \times U^{(f)} \tag{4.8}$$

where $\mathcal{G}$ is the core tensor that governs the interaction between the five different factors defining the dataset: identity, rotation in the three angles and the appearance (or features) of a sample. Specifically, note that each $U^{(*)}$ is a matrix spanning a subspace for a given factor. Therefore, its rows $\mathbf{u}^{(*)}$ can be seen as vectors representing the data behavior for each parameter of the factor subspace. For example, the rows of matrix $U^{(id)}$ encode the distinctive characteristics that define the shape of the object $\mathbf{x}_n \in \mathbb{R}^{D_f}$. At the same time, each row in the matrices $U^{(y)}, U^{(p)}$ and $U^{(r)}$ provides coefficients that define the rotation of the object by a particular angle about each axis. And finally, the product by $\mathcal{G}$ and $U^{(f)}$ can be interpreted as a mapping of the shaped and rotated object into feature space.

Thus, from our 5-way tensor, we have 4 modes that correspond to factor subspaces plus another one that corresponds to our input features and is usually combined with the core in the auxiliary

variable $\mathcal{W} = \mathcal{G} \times U^{(f)}$. This variable can be understood as a basis that represents the principal axes of variation in feature space across the various factors (the other tensor modes) and how they interact with each other to reconstruct the input features [Vasilescu and Terzopoulos, 2002].

After obtaining the decomposition of the tensor $\mathcal{T}$, a sample $\mathbf{x}$ can be reconstructed as:

$$\mathbf{x} = \mathcal{W} \times \mathbf{u}^{(id)} \times \mathbf{u}^{(y)} \times \mathbf{u}^{(p)} \times \mathbf{u}^{(r)}, \tag{4.9}$$

where $\mathcal{W} = \mathcal{G} \times U^{(f)}$ and $\{\mathbf{u}^{(y)}, \mathbf{u}^{(p)}, \mathbf{u}^{(r)}\}$ are row vectors from matrices $\{U^{(y)}, U^{(p)}, U^{(r)}\}$. Therefore, it is also theoretically possible to estimate the rotation angles for a given test sample $\mathbf{x} \in \mathbb{R}^{D_f}$ by minimizing the reconstruction error: [Tenenbaum and Freeman, 2000, Zhang et al., 2015]:

$$\underset{\mathbf{u}^{(y)},\mathbf{u}^{(p)},\mathbf{u}^{(r)},\mathbf{u}^{(id)}}{\operatorname{argmin}} \|\mathbf{x} - \mathcal{W} \times \mathbf{u}^{(y)} \times \mathbf{u}^{(p)} \times \mathbf{u}^{(r)} \times \mathbf{u}^{(id)}\| \tag{4.10}$$

Unfortunately, this becomes a minimization problem in which we need to simultaneously solve for 3 viewpoint parameterizations vectors (yaw $\mathbf{u}^{(y)}$, pitch $\mathbf{u}^{(p)}$ and roll $\mathbf{u}^{(r)}$), and the identity vector $\mathbf{u}^{(id)}$. There exist approaches to solve the above minimization, e.g. iterative estimates of one factor at a time or gradient-based optimization [Bakry and Elgammal, 2014]. However, they cannot guarantee accurate estimates and the resulting solutions are often not compliant with the manifold structure of the different subspaces. Thus, the final estimates are typically obtained by applying some correction to the results from the minimization of Eq. 4.10 (e.g. nearest neighbor search [Takallou and Kasaei, 2014]) so that they become compatible with the manifold structure implicitly defined by the training examples.

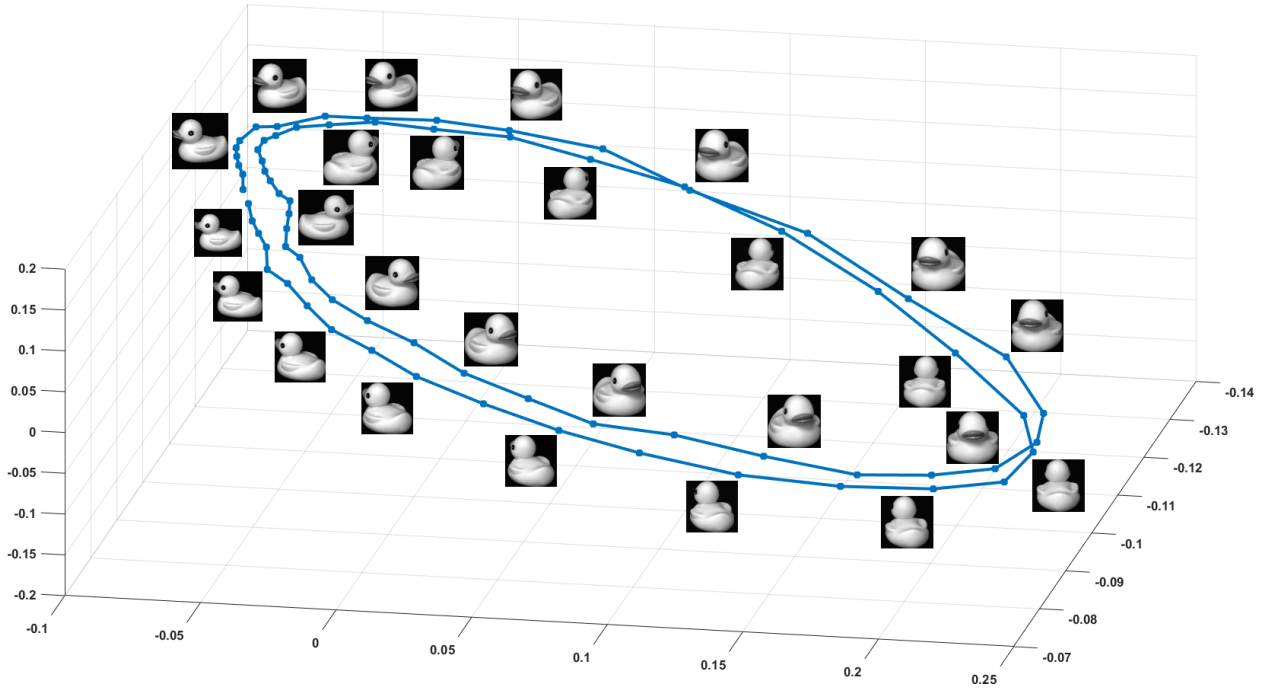

Figure 4.2: Visualization of the first three coefficients of the pose variation subspace.

4.4.2 Introducing rotation manifold constraints

In Sec. 4.4.1, we have described the use of tensor decomposition to model the effect of 3D rotations much like any other factor; e.g. notice how rotation factors and identity are treated in an equivalent manner. However, such general treatment does not take advantage of the special structure that can be expected for rotation subspaces leading to the generic formulation in Eq. 4.10, whose practical use is limited.

Indeed, notice that regardless of the resulting dimensionality of the yaw, pitch and roll vectors ($\mathbf{u}^{(y)}$, $\mathbf{u}^{(p)}$ and $\mathbf{u}^{(r)}$), each of them shall theoretically encode only the rotation about a specific axis. Hence, the coefficients of each of these vectors are expected to be governed by a single parameter (their corresponding rotation angle), thus describing a one-dimensional manifold embedded in the higher dimensional ambient space obtained by the tensor decomposition.

To illustrate the above, consider a simple example with a dataset composed of 2D images of different objects rotating with respect to a single axis (e.g, yaw angle). In this scenario, the data depends on three factors: yaw angle, object identity, and features (for which we can directly use the pixels). Using this dataset, we can build a tensor and decompose it analogously to Eq. 4.8. By following this procedure, we obtain three different matrices: $U^{(f)}$, $U^{(y)}$ and $U^{(id)}$, spanning the sub-spaces corresponding to features, rotation and identity, respectively. In Fig. 4.2 we show the values of the first three columns of matrix $U^{(y)}$, corresponding to different images of a "duck" object. We can see that the values displayed in Fig. 4.2 approximately describe a spiral curve, making apparent that the coefficients of the rotation subspace follow a uni-dimensional manifold structure. This is consistent with the fact that the variations captured by this subspace correspond to a single parameter: the rotation angle about the vertical axis. As we will show later in the experiments, this behaviour is not specific for rotations about a single axis but holds for general 3D-rotations and

also for different types of features.

Based on this observation, we propose to introduce explicit constraints over the rotation coefficients $\{\mathbf{u}^{(y)}, \mathbf{u}^{(p)},\mathbf{u}^{(r)}\}$ so that we can carry out their joint estimation directly over the underlying rotation manifold. For this purpose, we re-write Eq. 4.9 as follows:

$$\begin{aligned}\mathbf{x} &= \mathcal{W} \times \mathbf{u}^{(id)} \times \mathbf{u}^{(y)} \times \mathbf{u}^{(p)} \times \mathbf{u}^{(r)} \\ &= \mathcal{W} \times \mathbf{u}^{(id)} \times \mathbf{f}^{(y)}(\omega^{(y)}) \times \mathbf{f}^{(p)}(\omega^{(p)}) \times \mathbf{f}^{(r)}(\omega^{(r)})\end{aligned} \tag{4.11}$$

where $\mathbf{f}^{(*)} : \mathbb{R} \rightarrow \mathbb{R}^{D_*}$ are parametric functions taking as input an angle $\omega^{(*)}$ and giving as output a vector of coefficients $\mathbf{u}^{(*)}$. In this way, we explicitly force the rotation subspaces to be one-dimensional manifolds governed by $\omega^{(*)}$.

By considering the reparametrization defined in Eq. 4.11, the estimation of the rotation angles given a test sample $\mathbf{x}$ can be obtained by minimizing the following reconstruction error:

$$\begin{aligned}&\underset{\omega^{(y)},\omega^{(p)},\omega^{(r)},\mathbf{u}^{(id)}}{\operatorname{argmin}} \|\mathbf{x} - \hat{\mathbf{x}}\| \\ &\hat{\mathbf{x}} = \mathcal{W} \times \mathbf{u}^{(id)} \times f^{(y)}(\omega^{(y)}) \times f^{(p)}(\omega^{(p)}) \times f^{(r)}(\omega^{(r)})\end{aligned} \tag{4.12}$$

where $\mathcal{W} = \mathcal{G} \times U^{(f)}$ and the optimized variables are the angles ω^* and the vector $\mathbf{u}^{(id)}$.

This re-formulation of Eq. 4.10 allows to minimize the reconstruction error directly fulfilling the manifold structure of the rotation subspaces and reducing the number of coefficients to optimize from $(N + D_y + D_p + D_r)$ to $(N + 3)$. As we will show in the experimental results, this offers a crucial advantage with respect to the one described in Sec. 4.4.1.

4.4.3 Constraints definition using trigonometric functions

In the previous section, we have discussed the advantages of incorporating specific constraints into the rotation coefficients $\mathbf{u}^{(*)}$ (see Eq. 4.9) in order to explicitly model the pose changes of samples $\mathbf{x}$. For this purpose, we have parametrized each $\mathbf{u}^{(*)}$ by using a function $\mathbf{f}^{(*)}$ with a single angle $\omega^{(*)}$ as input. However, we have not discussed yet the specific definition of $\mathbf{f}^{(*)}$ used in this work.

To do so it is important to remember that we are looking for functions that transform a rotation angle into a set of coefficients that produce the desired rotation effect (once appropriately combined by means of $\mathcal{W}$). In this sense we may draw an analogy between the vectors $\mathbf{u}^{(*)} = \mathbf{f}^{(*)}(\omega^{(*)})$ and 3D rotation matrices, which also have multiple coefficients that in reality depend on a single parameter.[1] Therefore, it is reasonable to hypothesize that the relation between the coefficients of the rotation subspaces and their corresponding rotation angles can be modeled by means of trigonometric functions [Brannon, 2018].

Going back to the example from Fig. 4.2, we can see that the displayed rotation parameters describe an elliptical trajectory, compatible with the above hypothesis. This becomes more evident in Fig. 4.3, where we show the values of these coefficients separately against the rotation angle. It can be seen that the resulting wave-forms strongly resemble those from the cosine functions. It should be mentioned that, while we only display the first three dimensions of the rotation subspace (corresponding to the first three columns of matrix), the remaining columns follow a similar pattern. Therefore, we will model $\mathbf{f}^{(*)}$ as vectors of real functions based on cosines parameterized as follows:

[1] Of course this applies to rotations about a single axis; the general 3D-rotation case, depending on 3 parameters, could be anyway decomposed in the product of 3 such rotation matrices, analogous to the product of $\mathbf{f}^{(y)}(\omega^{(y)}) \times \mathbf{f}^{(p)}(\omega^{(p)}) \times \mathbf{f}^{(r)}(\omega^{(r)})$ in Eq. 4.11.

$$\mathbf{f}^{(*)}(\omega^{(*)}) = \Big(f_1(\omega^{(*)}),\, f_2(\omega^{(*)}),\, \ldots\, f_{D_*}(\omega^{(*)})\Big) \tag{4.13}$$

$$f_j(\omega^{(*)}) = \alpha_j^{(*)}\, cos(\beta_j^{(*)}\omega^{(*)} + \gamma_j^{(*)}) + \varphi_j^{(*)} \tag{4.14}$$

$$1 \leq j < D_*$$

where $\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}$ and $\varphi_j^{(*)}$ are parameters defining each specific cosine function. Note that, given a rotation subspace, there will be a different set of parameters for each dimension of the subspace, thus defining a spiral-like structure that analytically represents the underlying manifold.

To obtain the values of the parameters $\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}$ and $\varphi_j^{(*)}$ that define the analytic curves for each dimension of the rotation subspaces we solve the following minimization:

$$\underset{\alpha_j^{(*)},\beta_j^{(*)},\gamma_j^{(*)},\varphi_j^{(*)}}{\operatorname{argmin}} \|u_{ij}^{(*)} - f_j(\omega_i^{(*)})\| \tag{4.15}$$

where $u_{ij}^{(*)}$ are the elements of matrices $U^{(y)}, U^{(p)}, U^{(r)}$ obtained from the decomposition of the training tensor and $\omega_i^{(*)}$ is the value corresponding to the i-th bin of the discretized rotation angles used to construct the tensor. For example, given a rotation range of $[-\Theta, \Theta]$ and a uniform discretization:

$$\omega_i^{(*)} = \frac{2\,i - D_* - 1}{D_* - 1}\,\Theta \qquad 1 \leq i \leq D_* \tag{4.16}$$

4.4.4 Implementation

The central elements of our formulation are the minimizations defined in Eq. 4.12 and 4.15. In both cases, we note that the function to minimize is differentiable with respect to the target parameters. Therefore, we chose to solve these optimization problems by using a gradient descent procedure. Specifically we use the L-BFGS method [Byrd et al., 1994], which is formally a Quasi-Newton

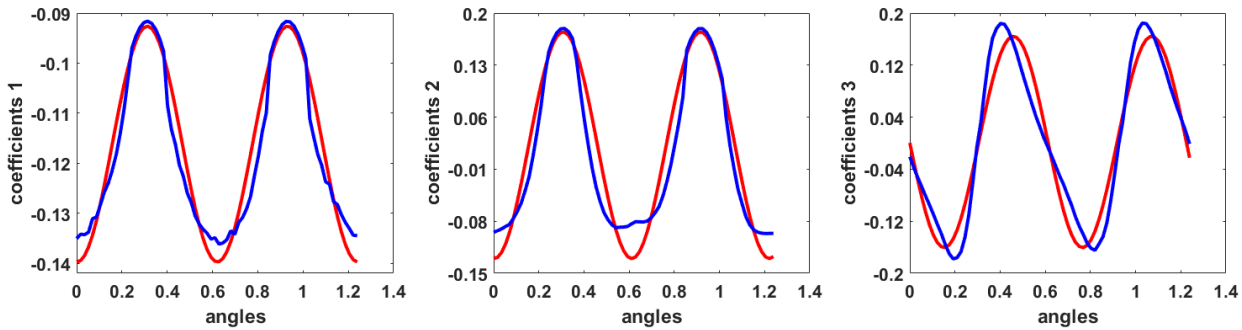

Figure 4.3: Values of the first three coefficients of the viewpoint subspace for the example from Fig. 4.2. The blue curves show the actual values of the first three columns of matrix $U^{(y)}$ and the red curves show their least-squares approximation with cosine functions.

method. L-BFGS applies an iterative gradient-descent procedure to minimize the objective function but, differently from other Quasi-Newton methods, it approximates the inverse Hessian matrix with a low-rank compact form that results in important savings in terms of memory and computational cost [Ruiz Ovejero et al., 2017]. For completeness, the required derivatives to use L-BFGS for the minimization of Eq. 4.12 and 4.15 are provided in the Appendix.

The different steps involved to train the proposed 3D pose estimation framework are summarized in Algorithm 1. We start from a set of feature vectors $\{X\}$, each labeled with an identity its yaw, pitch and roll angles. Depending on the input data, these angles may need to be discretized so that we obtain angular bins that are consistent across all identities. This allows to organize the input data into the 5D tensor $\mathcal{T}$, which is decomposed to obtain the core tensor $\mathcal{G}$ and the subspace matrices $U^{(*)}$ for each of the 5 factors in the tensor.

The feature-subspace matrix $U^{(f)}$, representing the input space, is combined with the core to form the auxiliary tensor $\mathcal{W}$; the identity-subspace matrix $U^{(id)}$ is kept unchanged[2] and the rotation-subspace matrices $U^{(y)}$, $U^{(p)}$ and $U^{(r)}$ are reparameterized in terms of cosine functions. Note that, for each rotation angle, there are as many cosine functions as dimensions in the yaw, pitch and roll subspaces, but all the functions in a given subspace are governed by the same unique free variable, respectively $\omega^{(y)}, \omega^{(p)}, \omega^{(r)}$.

During testing, the unknown identity and rotation angles of a feature vector $\mathbf{x}$ are estimated (Algorithm 2). This is done by solving the minimization in Eq. 4.12 using the auxiliary tensor $\mathcal{W}$ and the parameters learned during training for each function $\mathbf{f}^{(*)}$. Note that, because the factor subspaces are obtained by means of HOSVD, their parameters are sorted in terms of their eigenvalue. Therefore, even though each angular subspace results in $D_* - 1$ cosine functions, those with the smallest eigenvalues are typically discarded due to

[2]Actually, because in this Chapter we do not address identity recognition, the values in this matrix are not used during testing.

Algorithm 1: Training Phase

Input : $\{X\}$ – set of feature vectors for each of the subjects and each of the rotation angle labeled with $\omega^{(y)}, \omega^{(p)}, \omega^{(r)}$;
$D^{(*)}$ – number of the bins to discretize rotation;

Output: $\mathcal{W}$ and set of parameters $\alpha^{(*)}, \beta^{(*)}, \gamma^{(*)}, \varphi^{(*)}$ for each of the rotation;

Build 5D tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_5}$
Decompose $\mathcal{T}$ using HOSVD (Eq. 4.8)
  $\mathcal{T} = \mathcal{G} \times_1 U^{(id)} \times_2 U^{(y)} \times_3 U^{(p)} \times_4 U^{(r)} \times_5 U^{(f)}$
Compute $\mathcal{W}$:
  $\mathcal{W} = \mathcal{G} \times_5 U^{(f)}$
foreach $\omega^{(*)} \in \{\omega^{(y)}, \omega^{(p)}, \omega^{(r)}\}$ **do**
  foreach *j-th column in matrix* $U^{(*)}$*;* $(1 \leq j < D_*)$ **do**
    foreach *i-th bin of the discretized rotation angles* $(1 \leq i \leq D_*)$ **do**
      $\text{argmin}_{\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)}} \| u_{ij}^{(*)} - f_j(\omega_i^{(*)}) \|$;
      where $f_j(\omega_i^{(*)})$ is from Eq. 4.13;
    end
  end
end

their sensitivity to noise.

4.5 Experiments

In order to show that the proposed framework can be applied to a wide variety of input features, we perform several experiments. The first experiment is performed on the COIL-20 database [Nene et al., 1996], where we simply use the pixels of 2D images as features and use our framework to impose analytic constraints to out-of/-plane

Algorithm 2: Test Phase

Input : $\mathbf{x}$ – feature vector of unknown subject
Output: estimated angles – $\omega^{(y)}, \omega^{(p)}, \omega^{(r)}$

Initialize :
  $\omega^{(y)} = 0; \omega^{(p)} = 0; \omega^{(r)} = 0;$
  $\mathbf{u}^{(id)}$ as vector with zeros;
Define functions using Eq. 4.13:
foreach *j-th column in matrix* $U^{(*)}$*;* $(1 \leq j < D_*)$ **do**
  $f_j^{(y)}(\omega^{(y)}) = \alpha_j^{(y)} cos(\beta_j^{(y)} \omega^{(y)} + \gamma_j^{(y)}) + \varphi_j^{(y)}$
  $f_j^{(p)}(\omega^{(p)}) = \alpha_j^{(p)} cos(\beta_j^{(p)} \omega^{(p)} + \gamma_j^{(p)}) + \varphi_j^{(p)}$
  $f_j^{(r)}(\omega^{(r)}) = \alpha_j^{(r)} cos(\beta_j^{(r)} \omega^{(r)} + \gamma_j^{(r)}) + \varphi_j^{(r)}$
end
Estimate angles $\omega^{(y)}, \omega^{(p)}, \omega^{(r)}$:
$\text{argmin}_{\omega^{(y)}, \omega^{(p)}, \omega^{(r)}} \|\mathbf{x} - \hat{\mathbf{x}}\|$
where $\hat{\mathbf{x}} = \mathcal{W} \times \mathbf{f}^{(y)}(\omega^{(y)}) \times \mathbf{f}^{(p)}(\omega^{(p)}) \times \mathbf{f}^{(r)}(\omega^{(r)}) \times \mathbf{u}^{(id)}$

rotations of a variety of objects. Further, in Section 4.6, we perform 3D head pose estimation experiments using other two types of features: automatic landmarks and histogram-based 3D descriptors.

4.5.1 Image rotation manifold

We start our experiments using 2D images that capture rotations of simple objects along only one axis: the vertical axis (yaw angle). We consider the Columbia University Image Library (COIL-20) data-set [Nene et al., 1996]. The COIL-20 is an often-used dataset that contains a total of 1440 grayscale images from 20 different objects. Each image, of size 128 × 128 pixels, shows one of the objects at a particular rotation angle over a black background. There are 72 images for each object, taken at intervals of approximately 5°, thus covering the full range of rotations between 0 and 360 degrees. As the only pre-processing step, each image was downsampled to 32 ×

Figure 4.4: The sample images for 20 subjects in COIL-20 dataset

32 pixels and these were concatenated in row vectors of 1024 values which constituted our input features. Fig. 4.4 shows some sample images of this dataset.

Following Algorithm 1, the input data was arranged in a tensor. Due to the fact that images have only rotation about one axis (yaw), we built a 3-D tensor of size $20 \times 72 \times 1024$ ($\mathcal{T} \in \mathbb{R}^{20\times72\times1024}$). Then we applied tensor decomposition to obtain the core tensor and coefficients for each of the data factors subspaces. We can consider this a special case of Eq. 4.8 that reduces to:

$$\mathcal{T} = \mathcal{G} \times U^{(id)} \times U^{(y)} \times U^{(f)} \tag{4.17}$$

where $U^{(id)}$, $U^{(y)}$ and $U^{(f)}$ span the identity, rotation and feature subspaces, respectively. In particular, we are interested in $U^{(y)} \in \mathbb{R}^{72\times72}$; each element $u_{ij}^{(y)}$ contains the coefficient of the j-th subspace dimension for the i-th rotation angle. Thus, each column $U^{(y)}$ shows the behaviour of a particular dimension of the rotation subspace when the yaw angle varies and, as hypothesized in Section 4.4.3, it should approximately generate the waveform of a trigonometric function. Fig. 4.6 shows the coefficients of a few columns of matrix $U^{(y)}$ (in blue color) together with their cosine-based approximations (in red). The latter were computed by minimizing Eq. 4.13 for each column of matrix $U^{(y)}$, yielding an analytic representation $\mathbf{f}^{(y)}(\omega^{(y)})$ of the

underlying manifold structure of the rotation subspace, as defined in Eq. 4.13.

We can observe in Fig. 4.6 that the curves described by the coefficients of $U^{(y)}$ are indeed quite similar to cosine functions. On the other hand, we also see that the cosine-based approximations do not produce a perfect fit of the curves. This fact that is reasonably explained by the fact that, apart from rotations, the COIL-20 dataset includes also rather important variations in size, which for our settings can be considered spurious effect. Nevertheless, taking into account that we are trying to model out-of-plane rotations with 2D pictures using directly their pixels as input features, the curves in Fig. 4.6 comply strikingly well with the hypothesized behavior.

After rotation coefficients have been modeled, another way to assess the behavior of the proposed framework is by trying to synthesize rotated versions from an object from which we see only one image at a specific angle.[3] Specifically, given an input image (represented by feature vector $\mathbf{x}$) of an object whose rotation is $\omega_0^{(y)}$ we can obtain its identity vector $\mathbf{u}^{(id)}$ and combine it with $\mathbf{f}^{(y)}$ to synthesize images of the same object at other rotation angles $\omega^{(y)} \neq \omega_0^{(y)}$ using Eq. 4.11. Fig. 4.5 illustrates the images obtained by means of this procedure for a few objects, as well as the corresponding actual images from the database. We can see, again, that the quality of the synthesized images is not perfect and we can identify some minor artifacts not present in the original images; yet, in all cases we can clearly identify the object as well as the specific rotation angle that was synthesized, indicating that $\mathbf{f}^{(y)}$ has successfully modeled the effect of the rotation angle in this dataset.

[3]This process is coined *translation* in the seminal work by [Tenenbaum and Freeman, 2000].

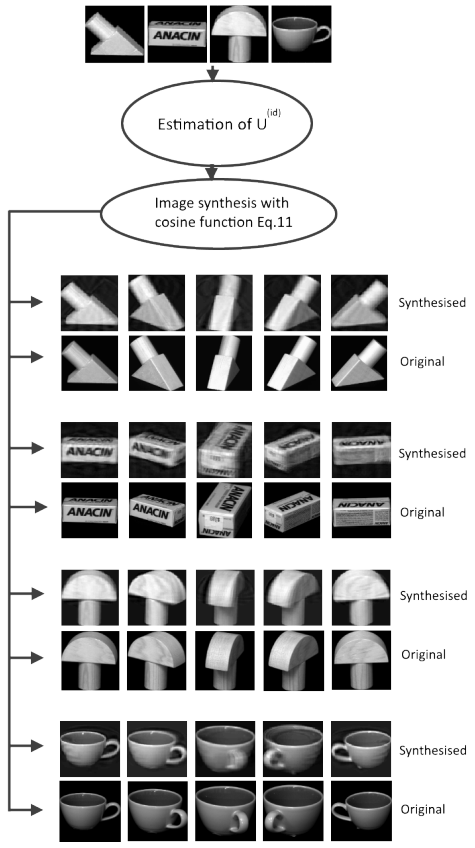

Figure 4.5: The images obtained by means of the synthesis procedure (Eq. 4.11) for a few objects, as well as the corresponding actual images from the database. In the very first line illustrated the sample images for several objects from COIL-20 database which representing training set. The two next blocks depict the process of obtaining identity vectors $\mathbf{u}^{(id)}$ for each object and combination it with trigonometric function $\mathbf{f}^{(y)}$ in order to synthesize images. In the following blocks, we can see the results of synthesized images (top line) and the corresponding original images (bottom line) for each object.

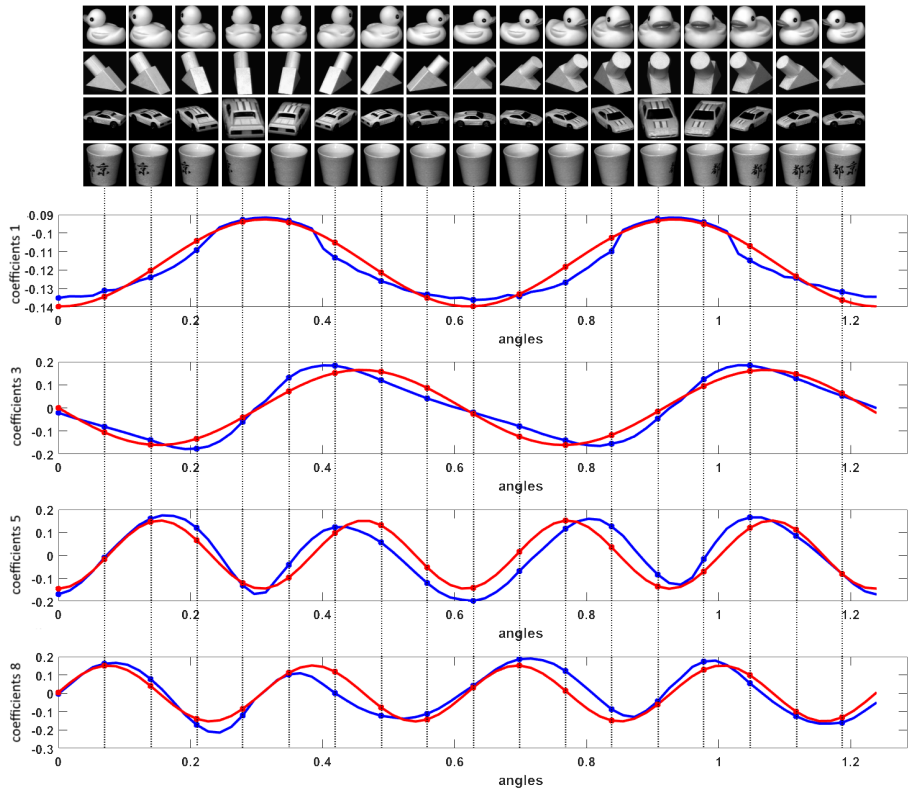

Figure 4.6: The first, third, fifth and eighth set of coefficients of the viewpoint subspace for entire range of angles. On the top of figure illustrated the sample images for several subject with different point of view. On the bottom, the set of coefficients, that corresponds to each of the presented point view, obtained from the tensor decomposition. The blue curves show the actual values of the first three columns of matrix $U^{(y)}$ and the red curves show their least-squares approximation with cosine functions.

4.6 3D head pose estimation experiments

In this section we demonstrate the application of the proposed framework to the problem of 3D head pose estimation from depth data. In contrast to the tests from Section 4.5.1, experiments of the present section imply full deployment of our framework, given that we address datasets where head rotations are not constrained to a single axis and contain combinations of yaw, pitch and roll rotations at the same time.

Given that our framework focuses on analytically modeling the underlying structure of the rotation manifold in general, it can be applied to a wide variety of input features. Nevertheless, the appropriate selection of input features is important to achieve quantitative results that demonstrate the relevance and applicability of the proposed method. Thus, we have selected the two main features from the system presented in Chapter 3, which obtained the first place in the recent Head Pose Estimation Challenge organized on the SASE database [Lüsi et al., 2017]. These features are composed by: 1) automatically detected landmarks (Section 3.3.1) and 2) histogram-based descriptors extracted around the landmarks (Section 3.3.3). We perform experiments over two large and publicly available 3D face corpora: the SASE [Lüsi et al., 2016b] and BIWI databases [Fanelli et al., 2013].

4.6.1 3D head pose estimation using SASE database

The data in SASE has been acquired with Microsoft Kinect 2 camera and contains RGB and depth images in pairs. The entire database includes 50 subjects (32 male and 18 female) in the range of 7-35 years old, with more than 600 frames per subject. For each person, a wide range of yaw, pitch and roll variations are included. Specifically, yaw and pitch angles vary within ± 75 °, while roll angles vary within

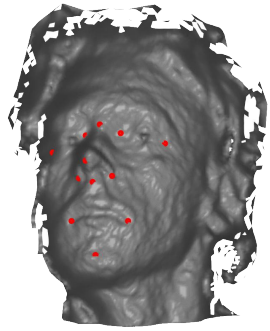

Figure 4.7: The example of the 3D mesh of the face with obtained landmarks

± 45 ° [Lüsi et al., 2016a].

As aforementioned in Chapter 3, the SASE database is distributed divided in two sets: *Training* (comprising 28 subjects with a total of $\sim$ 17K images) and *Validation* (12 subjects, $\sim$ 7K images) [Derkach et al., 2017]. Thereby, we have used each of these sets for training and testing, respectively. As mentioned before, we base our tests on the system described in Chapter 3, which is used as baseline, thus we use two type of features: automatically detected landmarks and local appearance around landmark points.

Following Chapter 3, we start by isolating the head region using clustering and use the obtained result to build a 3D mesh $\mathcal{M}$ that contains the head and a variable part of the shoulders. Then, mesh $\mathcal{M}$ is fed to the SRILF algorithm (Section 3.3.1) with the aim to automatically detect 12 prominent facial landmarks. An example of the 3D mesh of the face with the obtained landmarks is illustrated in Fig. 4.7. Once the facial landmarks are available, we use their coordinates as input features to train and test our approach as described in Section 4.4. It is worth to mention that the use of SRILF to extract the input features provides robustness to both expression changes and missing parts. The latter is especially

important in databases such as SASE and BIWI, because large pose variations induce self-occlusions that are likely to affect the visibility of some landmarks. SRILF deals with this problem by statistically inferring missing landmarks, thus providing a complete set of coordinates even under occlusions.

Pose estimation from landmarks

During the training phase, following Algorithm 1, we build a 5D tensor $\mathcal{T} \in \mathbb{R}^{28\times40\times40\times30\times36}$, that is: 28 subjects, 40 bins to discretize yaw and pitch angles in the range of $[-75°..75\circ]$, 30 bins for roll in the range of $[-55°..55°]$, and 36-dimensional features (12 landmarks $\times$ 3 coordinates). In order to fill all cells in this 5D tensor we need around 1.3 million samples, and it is obvious that the SASE database does not have this amount; it provides only about 5% of them. Thus, $\sim$ 95% of the data had to be generated synthetically. Specifically, if there is not a sample with i-th identity and target angles yaw $\omega^{(y)}$, pitch $\omega^{(p)}$ and roll $\omega^{(r)}$, we look for the closest sample with the same identity i from the training set and rotate it to the target angles (the amount of rotation is easily computed as the difference between the target angles and the ground-truth angles from the selected sample).

After the tensor is built, we decompose it using Eq. 4.8, obtaining the core tensor $\mathcal{G} \in \mathbb{R}^{28\times3\times3\times3\times10}$, matrix $U^{(id)} \in \mathbb{R}^{28\times28}$ for the identity subspace matrices $U^{(y)} \in \mathbb{R}^{40\times3}$, $U^{(p)} \in \mathbb{R}^{40\times3}$ and $U^{(r)} \in \mathbb{R}^{30\times3}$ for yaw, pitch and roll subspaces, and matrix $U^{(F)} \in \mathbb{R}^{36\times10}$ for the features subspace.

Next, we fit cosine functions to the pose coefficients (Eq. 4.13) and obtain four parameters$(\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}$ and $\varphi_j^{(*)})$ for each of the j-th coefficient of the three rotation subspaces (yaw, pitch and roll), thus achieving an analytic representation of the structure of the rotation manifolds. The results of the approximated coefficients for 3D pose variations are illustrated in Fig. 4.8. For each of the rotation subspaces (yaw, pitch and roll), the first, second and third coefficients of all angle variations are plotted with two colors. The

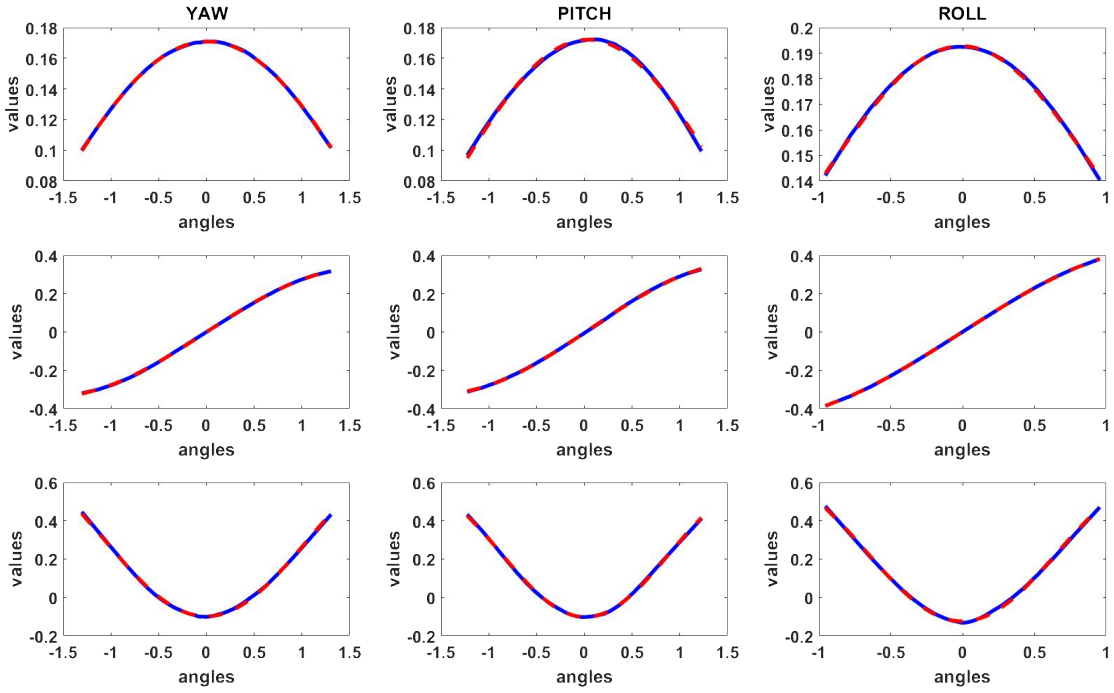

Figure 4.8: Curves defined by the coefficients in each of the subspaces corresponding to the head pose variation along one of the rotation axes. The first column corresponds to yaw rotation and shows the curves built from the coefficients of the first three columns of matrix $U^{(y)}$ (blue) and their approximation with a cosine function (red). The second and third columns correspond to pitch and roll angles, respectively

blue curves are the original values from the first three columns of the matrices $U^{(y)}, U^{(p)}$ and $U^{(r)}$ and the red curves are the approximated values obtained with cosine functions. It can be observed that the trigonometric approximation provides an excellent fit for the three rotation angles, with only minor deviations that could be easily attributed to noise in the data or in the extracted features.

Based on the obtained coefficients, similarly to the case with 2D images, we can synthetically generate sets of landmarks by sampling our analytic manifold. For example, given a particular subject and target angles $\omega^{(y)}$, $\omega^{(p)}$ and $\omega^{(r)}$, we can synthesize the corresponding set of landmarks using Eq. 4.11. If this procedure is repeated while varying one of the angles, we can get a graphical illustration of how the effect of this angle has been captured by our framework. Fig. 4.9 shows an example in which identity, pitch and roll angles are fixed, while yaw varies from $-75°$ to $75°$.

After all function parameters were obtained, we used the estimation approach based on the minimization of the reconstruction error (Eq. 4.12). For the test stage, $\sim$ 7K facial images from the Validation subset of SASE were used. We compared the obtained results of the proposed framework with respect to the approach based on minimizing the reconstruction error without constraints [4]. Table 4.1 summarizes the average pose estimation errors obtained by each approach. It can be seen that the approach based on the minimization without constraints obtained considerably higher estimation errors, confirming the usefulness of imposing manifold-compliant constraints.

[4]Note that, while the minimization does not include constraints, the obtained results are forced to comply with the manifold after each iteration using nearest-neighbour search. Results without such *correction* would be worse than those reported and not meaningful for comparison, since this is a widespread practice.

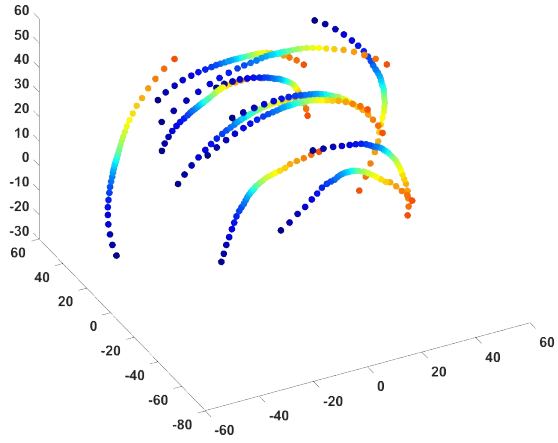

Figure 4.9: The example of generated landmarks using trigonometric function (Eq. 4.11). Landmarks coordinates change with rotation about vertical axis (yaw angle), i.e. during each step of the iteration, pitch and roll angles are fixed as frontal, and yaw angle varies from $-75°$ to $75°$. thus, according to the position of the landmarks, we can see how face moving from left ($-75°$, the first blue dot) to right ($75°$, the last orange dot)

Pose estimation from local surface appearance

In this section we perform experiments with the local surface descriptors presented in Chapter 3 Section 3.3.3. An interesting aspect of using these descriptors is that, because they are based on spatial histograms of local patches from the surface, rotations will have a non-linear effect on the descriptor values. An illustration of a 3D mesh of the face with the local descriptors is provided in Fig. 4.10.

Similarly to the experiment with landmarks from the previous section, for the training phase we build a 5D tensor $\mathcal{T} \in \mathbb{R}^{28\times40\times40\times30\times512}$, i.e. now the features dimension is 512. Then we decompose it using Eq. 4.8, obtain coefficients for each

	Yaw	Pitch	Roll
Without constraint	12.18	13.51	10.38
With constraint (proposed)	**6.50**	**7.07**	**6.06**

Table 4.1: Average pose estimation errors tested on the SASE database using coordinates of landmarks

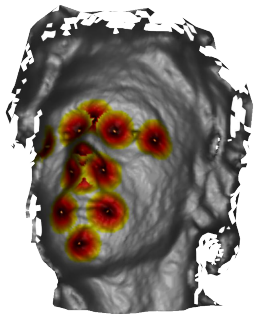

Figure 4.10: Illustration of a 3D mesh of the face with the neighbourhoods used to compute local descriptors.

of the factor subspaces and achieve an analytic representation of the structure of the rotation manifolds by fitting cosine functions (Eq. 4.13). The results of the approximated coefficients for 3D pose variations are illustrated in Fig. 4.11. Finally, we use the modeled coefficients to estimate the 3D head pose based on the minimization of the reconstruction error (Eq. 4.12).

Note that, following Chapter 3 Section 3.3.3, we build 12 different tensors and produce different estimates for each angle (i.e. one per landmark descriptor). However, because of the potential presence of occlusions, it is not guaranteed that all estimated landmarks will

	Yaw	Pitch	Roll
[Lüsi et al., 2016a]	22	19	18
[Derkach et al., 2017]	6.51	7.49	6.52
Proposed LMK	**6.50**	**7.07**	**6.06**
Proposed DESC	**6.21**	**6.64**	**4.6**

Table 4.2: Average pose estimation errors of the proposed framework and previous works on the SASE database

actually lie on the mesh surface [5]. Indeed, when parts of the facial surface are missing, it is possible that some landmarks are estimated relatively far from the mesh $\mathcal{M}$, i.e. they are inferred in the position where we would statistically expect them to be, despite no surface has been captured there (Fig. 3.3).

Therefore, we use the indicator function $\mathbb{1}(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon)$ to filter out the estimates from landmarks $\hat{\mathbf{x}}_\ell$ that are estimated off the surface and produce our final appearance-based estimate as the average of the remaining ones:

$$\omega^{(*)} = \frac{\sum_\ell \mathbb{1}\Big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\Big)\omega_\ell^{(*)}}{\sum_\ell \mathbb{1}\Big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\Big)} \tag{4.18}$$

where $\omega_\ell^{(*)}$ is the estimated angles from ℓ-th landmark; the distance from $\hat{\mathbf{x}}_\ell$ to $\mathcal{M}$ is computed as the distance to the nearest mesh vertex:

$$\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| = \min_{v_j \in \mathcal{M}} \|\hat{\mathbf{x}}_\ell - v_j\| \tag{4.19}$$

Table 4.2 summarizes the average pose estimation errors of the proposed framework using both landmarks and appearance features on the SASE database. In this table, we also compare our results

[5] We consider that a landmark is *on the surface* when its distance to it is relatively small as compared to the mesh resolution.

to other methods reporting head pose estimation error on the SASE database. Since this database is rather new, only a few papers have reported results on it. We can see that the proposed method performs well compared with state-of-the-art methods on the same dataset.

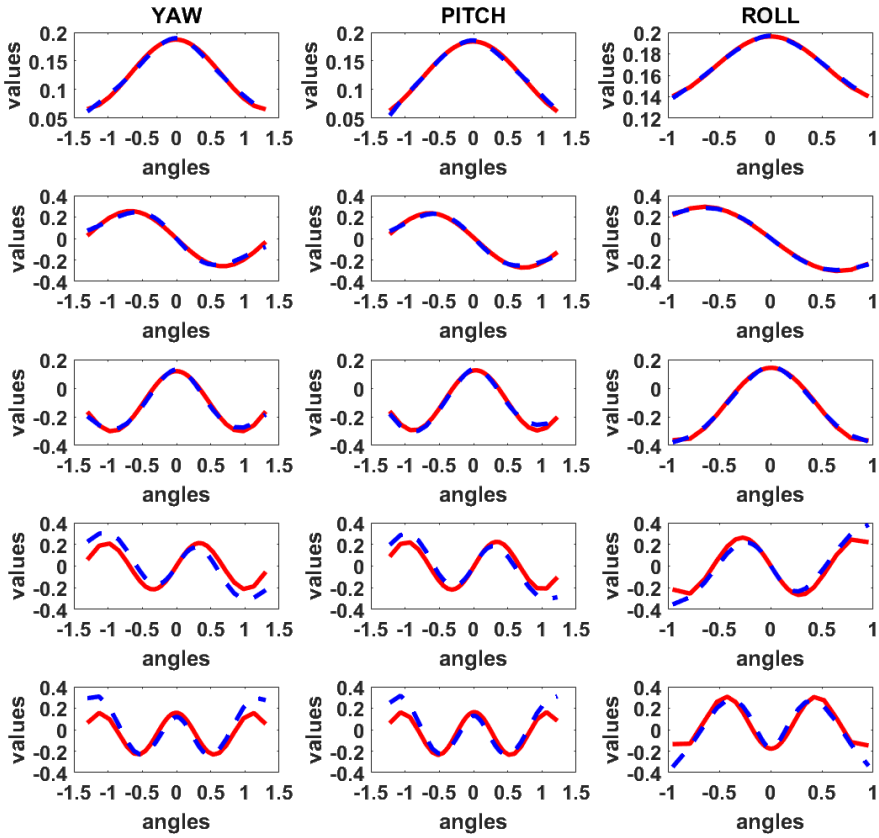

Figure 4.11: Curves defined by the coefficients in each of the subspaces corresponding to the head pose variation along one of the rotation axes using local descriptors as features. The first column corresponds to yaw rotation and shows the curves built from the coefficients of the first five columns of matrix $U^{(y)}$ (dashed blue lines) and their approximation with a cosine function (red solid lines). The second and third columns correspond to pitch and roll angles, respectively.

4.6.2 3D head pose estimation using BIWI database

The BIWI Database [Fanelli et al., 2013], acquired with a Kinect 1 sensor, contains 24 sequences of RGB-D images of subjects moving their heads over a range of roughly $\pm 75^\circ$ for yaw, $\pm 60^\circ$ for pitch and $\pm 50^\circ$ for roll. In total this database consists of around 17K images. Because there is no standard experimental protocol for this database, we perform our experiments under a leave-one-sequence-out strategy, so that no sequence is used for training and test at the same time. All other settings were kept as described in the previous section for the SASE database.

Table 4.3 summarizes our results, as well as those presented by previous works reporting pose estimation errors on this database. For each method, we show the average absolute error per angle together with the respective standard deviations (when provided by the authors). We also indicate the type of input data that is used (depth, RGB or both) and if pose estimations are done per-frame or using tracking.

The first thing we notice is that, despite our approach is the only one using just depth information without tracking, our results are quite competitive. Indeed, we clearly outperform other methods not doing tracking (except [Papazov et al., 2015] who report smaller averages but considerably higher standard deviations). Additionally, we achieve results that are comparable or better than four out the the seven tracking-based methods listed in Table 4.3, even though tracking-based algorithms benefit from the fact that test sequences start with nearly frontal head poses; thus, the accuracy of these algorithms to detect initial head poses other than frontal is not clear.

Another interesting aspect is that, among methods reporting standard deviations, our approach obtains the second-best results, only behind those from [Padeleris et al., 2012], who use tracking.

Method	Tracking	Errors ± Std			Domain
		Yaw	Pitch	Roll	
[Wang et al., 2013a]	No	8.8 ± 14.3	8.5 ± 11.1	7.4 ± 10.8	RGB + depth
[Chen et al., 2016]	No	9.9 ± 12.4	12.8 ± 17.2	6.9 ± 9.8	RGB
[Papazov et al., 2015]	No	2.5 ± 8.3	1.8 ± 4.3	2.9 ± 12.8	RGB + depth
	Yes	3.0 ± 9.6	2.5 ± 7.4	3.8 ± 16.0	
[Padeleris et al., 2012]	Yes	2.4 ± 1.8	3.0 ± 2.16	2.8 ± 2.1	depth
[Fanelli et al., 2011]	Yes	8.9 ± 13.0	8.5 ± 9.9	7.9 ± 8.3	depth
[Meyer et al., 2015]	Yes	2.1	2.1	2.4	depth
[Baltrušaitis et al., 2012]	Yes	6.3	5.1	11.3	RGB + depth
[Li et al., 2016]	Yes	3.0	3.2	5.3	RGB + depth
[Yu et al., 2017]	Yes	2.5	1.5	2.2	RGB + depth
Proposed LMK	**No**	**3.6 ± 4.6**	**3.8 ± 4.8**	**5.2 ± 5.8**	**depth**
Proposed DESC	**No**	**3.3 ± 4.2**	**3.4 ± 4.4**	**3.3 ± 3.7**	**depth**

Table 4.3: Average pose estimation errors and standard deviations of the proposed frame-work and previous works on the BIWI database

4.7 Conclusions

In this Chapter we address 3D head pose estimation from depth data by proposing a novel approach to learn the manifold defined by 3D rotations. In particular, our method is able to explicitly model the underlying structure of the rotation manifold with an analytic form that takes into account the specific constraints imposed by orientation variations. For this purpose, we use multi-linear decomposition to split the pose variation factors into separate sub-spaces accounting for yaw, pitch and roll effects. We show that the coefficients within each of these subspaces define a continuous curve that can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. We exploit this fact to introduce a minimization framework for pose estimation based on tensor decomposition constrained by trigonometric functions so that the obtained solutions are always compliant with the underlying manifold structure.

We show that the proposed modeling based on trigonometric functions can accurately capture the behaviour observed in the coefficients from the pose subspaces, by means of qualitative examples on 2D and 3D datasets. We also provide quantitative results of head pose estimation in two public database, which demonstrate the advantages introduced by the proposed constraints. Firstly, on the challenging SASE database, we show that directly applying existing multi-linear decomposition approaches yields poor pose estimation errors, which dramatically improve when introducing the proposed trigonometric constraints, reaching top state-of-the-art estimates. Later, we also report results on the widely used BIWI database, showing that the proposed framework is not only of theoretical interest but it can be translated into a practical system to produce competitive pose estimation results.

Chapter 5

CONCLUSIONS

5.1 Research summary

In **Chapter 2** we have been focused on 3D facial expression recognition. We have extended the analysis of 3D geometry from one of the promising methods – curve-based representation, into a spectral representation. Based on this representation, a complete description of the underlying surface was built with maintaining a fully-3D framework. We have proposed the use of Graph Laplacian Features (GLFs), which result from the projection of local surface patches into a common basis obtained from the Graph Laplacian eigenspace, much like a Fourier transform into the spatial frequency bases of the surface patches. The proposed approach was compared with two others approaches. The first one was the curves-based framework and the second one was the straight-forward alternative for spectral representation, Shape-DNA, which is based on the Laplace Beltrami Operator. We have shown that the straight-forward application of Shape-DNA is not the best way to deal with local face patches, since it cannot provide a stable basis to guarantee that the extracted signatures for the different patches are directly comparable.

Further, a state-of-the-art algorithm for 3D landmark localization

was also integrated, which enabled us to perform experiments under fully-automatic operation. The three most popular databases for 3D FER (BU-3DFE, Bosphorus and BU-4DFE) have been used for testing the proposed approach. It was performed in terms of FER rates and, additionally, in terms of AU recognition when AU labels were available (BU-3DFE and Bosphorus). Our results have shown that the proposed GLFs consistently outperform the curves-based and Shape-DNA alternatives under any of the experimental setings (non-automatic and fully-automatic), both in terms of expression recognition and AU recognition. Moreover, the recognition rates of Shape-DNA were even lower than those in the curves-based framework, as predicted by the theory.

Interestingly, the accuracy improvement brought by GLFs was obtained also at a lower computational cost. Considering the extraction of patches as a common step between the three compared approaches, the curves-based framework requires a costly elastic deformation between corresponding curves (e.g. based on splines) and Shape-DNA requires computing the eigen-decomposition of each new patch to be analyzed. In contrast, GLFs only require the projection of the patch geometry into the Graph Laplacian eigenspace, which is common to all patches and can thus be pre-computed off-line.

Comparison to other works reporting 3D FER and AU detection results confirmed that the proposed method allows achieving top performance by simply feeding GLFs to off-the-shelf SVM classifiers. Also, we showed that 14 automatically detected landmarks were enough to achieve high FER and AU detection rates, only slightly below those obtained when using sets of manually provided landmarks.

In **Chapter 3** we have presented a head pose estimation system which is able to obtain an accurate head pose estimation from a single depth frame of consumer RGB-D cameras, such as Kinect 2. We have based our system on the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based

pose estimators. A state-of-the-art landmark localization has provided us an accurate landmark coordinates with no need for initialization and tolerance to occlusions or missing data. Our system combines three different methods for pose estimation: two of them are based on state-of-the-art landmark detection and the third one is a dictionary-based approach that is able to work in especially challenging scans where landmarks or mesh correspondences are too difficult to obtain. It was evaluated on the SASE database, which consists of $\sim$ 30K frames from 50 subjects and obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge. Our experiments also confirmed the initial hypothesis that the landmark-based estimates would be more accurate than correspondence-free approaches, such as the dictionary-based one that was adopted. Landmark-based estimates were successfully produced for $\sim$ 90% of cases and the remaining ones were tackled by the dictionary-based approach. Our results compare well with those reported in the related literature, especially considering the added difficulty of not using tracking and RGB data to produce our estimates.

Based on the success obtained in Chapter 3, we have addressed 3D head pose estimation from depth data by proposing a novel approach to learn the manifold defined by 3D rotations. In particular, in **Chapter 4**, we have presented a method that is able to explicitly model the underlying structure of the rotation manifold with an analytic form that takes into account the specific constraints imposed by orientation variations. For this purpose, we have used multi-linear decomposition to split the pose variation factors into separate sub-spaces accounting for yaw, pitch and roll effects. We have shown that the coefficients within each of these subspaces define a continuous curve that can be modeled in terms of trigonometric functions, which are indeed the bases to explain rotation effects. We have exploited this fact to introduce a minimization framework for pose estimation based on tensor decomposition constrained by

trigonometric functions so that the obtained solutions are always compliant with the underlying manifold structure.

We have shown that the proposed modeling based on trigonometric functions can accurately capture the behaviour observed in the coefficients from the pose subspaces, by means of qualitative examples on 2D and 3D datasets, and using different types of features. We also provided quantitative results of head pose estimation in two public databases, which demonstrate the advantages introduced by the proposed constraints. Firstly, on the challenging SASE database (used in Chapter 3), we have shown that directly applying existing multi-linear decomposition approaches yields poor pose estimation errors, which dramatically improve when introducing the proposed trigonometric constraints, reaching top state-of-the-art estimates. Later, we also reported results on the widely used BIWI database, showing that the proposed framework is not only of theoretical interest but it can be translated into a practical system to produce competitive pose estimation results.

Appendix A

TECHNICAL DETAILS

For completeness of Section 4.4.4 in Chapter 4, the required derivatives to use L-BFGS for the minimization of Eq. 4.12 and 4.15 are provided in this Appendix.

For the objective function Eq. 4.15, partial derivatives should be computed with respect to variables $\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)}$. Let's rewrite this equation as:

$$\underset{\alpha_j^{(*)},\beta_j^{(*)},\gamma_j^{(*)},\varphi_j^{(*)}}{\operatorname{argmin}} \|u_{ij}^{(*)} - f_j(\omega_i^{(*)})\| = \underset{\alpha_j^{(*)},\beta_j^{(*)},\gamma_j^{(*)},\varphi_j^{(*)}}{\operatorname{argmin}} E(\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)})$$

Where error function E can be written in element form as:

$$E(\alpha_j^{(*)}, \beta_j^{(*)}, \gamma_j^{(*)}, \varphi_j^{(*)}) = \frac{1}{2} \sum_i \left(u_{ij}^{(*)} - (\alpha_j^{(*)} \cos(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}) + \varphi_j^{(*)})\right)^2$$

Thus, partial derivatives of the function E looks as:

$$\begin{aligned}\frac{\partial E}{\partial \alpha_j^{(*)}} =& \sum_i \big(\cos(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}) \cdot \\ & \cdot (\alpha_j^{(*)} \cos(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}) - u_{ij}^{(*)} + \varphi_j^{(*)})\big)\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial \beta_j^{(*)}} =& \alpha_j^{(*)} \sum_i \big(\omega_i^{(*)} \sin(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}) \cdot \\ & \cdot (u_{ij}^{(*)} - \alpha_j^{(*)} \cos(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}) - \varphi_j^{(*)})\big)\end{aligned}$$

$$\frac{\partial E}{\partial \gamma_j^{(*)}} = \alpha_j^{(*)} \sum_i \big(\sin\left(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}\right) \cdot \\ \cdot \left(u_{ij}^{(*)} - \alpha_j^{(*)} \cos\left(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}\right) - \varphi_j^{(*)}\right)\big)$$

$$\frac{\partial E}{\partial \varphi_j^{(*)}} = \sum_i \left(\alpha_j^{(*)} \cos\left(\beta_j^{(*)} \omega_i^{(*)} + \gamma_j^{(*)}\right) + \varphi_j^{(*)} + u_{ij}^{(*)}\right)$$

Similarly to previous case, error function in Eq. 4.12 can be written as:

$$E(\omega^{(y)}, \omega^{(p)}, \omega^{(r)}, \mathbf{u}^{(id)}) = \frac{1}{2} \sum_n \left(\mathbf{x}_n - \hat{\mathbf{x}}_n\right)^2$$

where $\mathbf{x}_n$ using Eq. 4.13 looks as:

$$\mathbf{x}_n = \sum_i \sum_j \sum_k \sum_l \left(\mathcal{W}_{ijkln} \cdot f_i^{(y)}(\omega^{(y)}) \cdot f_j^{(p)}(\omega^{(p)}) \cdot f_k^{(r)}(\omega^{(r)}) \cdot u_l^{(id)}\right)$$

Now we can compute partial derivatives with respect to variables $\omega^{(y)}, \omega^{(p)}, \omega^{(r)}$ and each l-th element in vector $\mathbf{u}^{(id)}$:

$$\frac{\partial E}{\partial \omega^{(y)}} = -\sum_n \left((\mathbf{x}_n - \hat{\mathbf{x}}_n) \cdot \frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(y)}}\right)$$

$$\frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(y)}} = -\sum_i \sum_j \sum_k \sum_l \left(\mathcal{W}_{ijkln} \cdot \alpha_i^{(y)} \sin(\beta_i^{(y)} \omega^{(y)} + \gamma_i^{(y)}) \beta_i^{(y)} \cdot \\ \cdot f_j^{(p)}(\omega^{(p)}) \cdot f_k^{(r)}(\omega^{(r)}) \cdot u_l^{(id)}\right)$$

$$\frac{\partial E}{\partial \omega^{(p)}} = -\sum_n \left((\mathbf{x}_n - \hat{\mathbf{x}}_n) \cdot \frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(p)}}\right)$$

$$\frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(p)}} = -\sum_i \sum_j \sum_k \sum_l \left(\mathcal{W}_{ijkln} \cdot f_i^{(y)}(\omega^{(y)}) \cdot \right.$$
$$\left. \cdot \left(\alpha_j^{(p)} \sin(\beta_j^{(p)} \omega^{(p)} + \gamma_j^{(p)}) \beta_j^{(p)}\right) \cdot f_k^{(r)}(\omega^{(r)}) \cdot u_l^{(id)}\right)$$

$$\frac{\partial E}{\partial \omega^{(r)}} = -\sum_n \left((\mathbf{x}_n - \hat{\mathbf{x}}_n) \cdot \frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(r)}}\right)$$

$$\frac{\partial \hat{\mathbf{x}}_n}{\partial \omega^{(r)}} = -\sum_i \sum_j \sum_k \sum_l \left(\mathcal{W}_{ijkln} \cdot f_i^{(y)}(\omega^{(y)}) \cdot f_j^{(p)}(\omega^{(p)}) \cdot \right.$$
$$\left. \cdot \left(\alpha_k^{(r)} \sin(\beta_k^{(r)} \omega^{(r)} + \gamma_k^{(r)}) \beta_k^{(r)}\right) \cdot u_l^{(id)}\right)$$

$$\frac{\partial E}{\partial \mathbf{u}_l^{(id)}} = -\sum_n \left((\mathbf{x}_n - \hat{\mathbf{x}}_n) \cdot \frac{\partial \hat{\mathbf{x}}_n}{\partial \mathbf{u}_l^{(id)}}\right)$$

$$\frac{\partial \hat{\mathbf{x}}_n}{\partial \mathbf{u}_l^{(id)}} = \sum_i \sum_j \sum_k \left(\mathcal{W}_{ijkln} \cdot f_i^{(y)}(\omega^{(y)}) \cdot f_j^{(p)}(\omega^{(p)}) \cdot f_k^{(r)}(\omega^{(r)})\right)$$

Bibliography

[Ahn et al., 2014] Ahn, B., Park, J., and Kweon, I. S. (2014). Real-time head orientation from a monocular camera using deep neural network. In *Asian Conference on Computer Vision*, pages 82–96. Springer.

[Ammar et al., 2010] Ammar, M. B., Neji, M., Alimi, A. M., and Gouardères, G. (2010). The affective tutoring system. *Expert Systems with Applications*, 37(4):3013–3023.

[Amor et al., 2014] Amor, B. B., Drira, H., Berretti, S., Daoudi, M., and Srivastava, A. (2014). 4-D facial expression recognition by learning geometric deformations. *IEEE transactions on cybernetics*, 44(12):2443–2457.

[Arapakis et al., 2009] Arapakis, I., Moshfeghi, Y., Joho, H., Ren, R., Hannah, D., and Jose, J. M. (2009). Integrating facial expressions into user profiling for the improvement of a multimodal recommender system. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1440–1443. IEEE.

[Azazi et al., 2015] Azazi, A., Lutfi, S. L., Venkat, I., and Fernández-Martínez, F. (2015). Towards a robust affect recognition: Automatic facial expression recognition in 3D faces. *Expert Systems with Applications*, 42(6):3056–3066.

[Ba and Odobez, 2009] Ba, S. O. and Odobez, J.-M. (2009). Recognizing visual focus of attention from head pose in natural meetings. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):16–33.

[Bakry and Elgammal, 2014] Bakry, A. and Elgammal, A. (2014). Untangling object-view manifold for multiview recognition and pose estimation. In *European Conference on Computer Vision*, pages 434–449. Springer.

[Balasubramanian et al., 2007] Balasubramanian, V. N., Ye, J., and Panchanathan, S. (2007). Biased manifold embedding: A framework for person-independent head pose estimation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE.

[Baltrušaitis et al., 2012] Baltrušaitis, T., Robinson, P., and Morency, L.-P. (2012). 3D constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2610–2617. IEEE.

[Barros et al., 2018] Barros, J. M. D., Mirbach, B., Garcia, F., Varanasi, K., and Stricker, D. (2018). Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2028–2037. IEEE.

[BenAbdelkader, 2010] BenAbdelkader, C. (2010). Robust head pose estimation using supervised manifold learning. In *European Conference on Computer Vision*, pages 518–531. Springer.

[Bergqvist and Larsson, 2010] Bergqvist, G. and Larsson, E. G. (2010). The higher-order singular value decomposition: Theory and an application [lecture notes]. *IEEE Signal Processing Magazine*, 27(3):151–154.

[Berretti et al., 2013] Berretti, S., Del Bimbo, A., and Pala, P. (2013). Automatic facial expression recognition in real-time from dynamic sequences of 3D face scans. *The Visual Computer*, 29(12):1333–1350.

[Berretti et al., 2010] Berretti, S., Del Bimbo, A., Pala, P., Amor, B. B., and Daoudi, M. (2010). A set of selected sift features for 3D facial expression recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4125–4128. IEEE.

[Bhatia, 2013] Bhatia, R. (2013). *Matrix analysis*, volume 169. Springer Science & Business Media.

[Blanz and Vetter, 2003] Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3D morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074.

[Brannon, 2018] Brannon, R. (2018). *Rotation, Reflection, and Frame Changes*. IOP Publishing.

[Breitenstein et al., 2008] Breitenstein, M. D., Kuettel, D., Weise, T., Van Gool, L., and Pfister, H. (2008). Real-time face pose estimation from single range images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[Bronstein and Bronstein, 2011] Bronstein, M. M. and Bronstein, A. M. (2011). Shape recognition with spectral distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1065–1071.

[Byrd et al., 1994] Byrd, R. H., Nocedal, J., and Schnabel, R. B. (1994). Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3):129–156.

[Canavan et al., 2012] Canavan, S., Sun, Y., Zhang, X., and Yin, L. (2012). A dynamic curvature based approach for facial activity analysis in 3D space. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 14–19. IEEE.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27.

[Chavel, 1984] Chavel, I. (1984). *Eigenvalues in Riemannian geometry*, volume 115. Academic press.

[Chen et al., 2016] Chen, J., Wu, J., Richter, K., Konrad, J., and Ishwar, P. (2016). Estimating head pose orientation using extremely low resolution images. In *Image Analysis and Interpretation (SSIAI), 2016 IEEE Southwest Symposium on*, pages 65–68. IEEE.

[Chung, 1997] Chung, F. R. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

[Cohn et al., 2009] Cohn, J. F., Kruez, T. S., Matthews, I., Yang, Y., Nguyen, M. H., Padilla, M. T., Zhou, F., and De la Torre, F. (2009). Detecting depression from facial actions and vocal prosody. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–7. IEEE.

[Comon, 2014] Comon, P. (2014). Tensors: a brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53.

[Corneanu et al., 2016] Corneanu, C. A., Simon, M. O., Cohn, J. F., and Guerrero, S. E. (2016). Survey on rgb, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1548–1568.

[Courant and Hilbert, 1965] Courant, R. and Hilbert, D. (1965). *Methods of mathematical physics*, volume 1. CUP Archive.

[De Lathauwer et al., 2000] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278.

[Derkach et al., 2017] Derkach, D., Ruiz, A., and Sukno, F. M. (2017). Head pose estimation based on 3-d facial landmarks localization and regression. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 820–827. IEEE.

[Derkach et al., 2018] Derkach, D., Ruiz, A., and Sukno, F. M. (2018). 3d head pose estimation using tensor decomposition and non-linear manifold modeling. In *3D Vision (3DV), 2018 International Conference on*, pages 505–513. IEEE.

[Derkach and Sukno, 2017] Derkach, D. and Sukno, F. M. (2017). Local shape spectrum analysis for 3D facial expression recognition. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 41–47.

[Desbrun et al., 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co.

[D'Hose et al., 2007] D'Hose, J., Colineau, J., Bichon, C., and Dorizzi, B. (2007). Precise localization of landmarks on 3D faces using gabor wavelets. In *Biometrics: Theory, Applications, and Systems. First International Conference on*, pages 1–6. IEEE.

[Drira et al., 2013] Drira, H., Amor, B. B., Srivastava, A., Daoudi, M., and Slama, R. (2013). 3D face recognition under expressions, occlusions, and pose variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9):2270–2283.

[Du et al., 2014] Du, S., Tao, Y., and Martinez, A. M. (2014). Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462.

[Eisert and Girod, 1998] Eisert, P. and Girod, B. (1998). Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, 18(5):70–78.

[Ekman, 1994] Ekman, P. (1994). Strong evidence for universals in facial expressions: a reply to russell's mistaken critique.

[Ekman and Friesen, 1971] Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

[Ekman et al., 1978] Ekman, P., Friesen, W. V., and Hager, J. C. (1978). Facial action coding system (facs). *A technique for the measurement of facial action. Consulting, Palo Alto.*

[Fanelli et al., 2013] Fanelli, G., Dantone, M., Gall, J., Fossati, A., and Van Gool, L. (2013). Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458.

[Fanelli et al., 2011] Fanelli, G., Weise, T., Gall, J., and Van Gool, L. (2011). Real time head pose estimation from consumer depth cameras. In *Joint Pattern Recognition Symposium*, pages 101–110. Springer.

[Fang et al., 2011a] Fang, T., Zhao, X., Ocegueda, O., Shah, S. K., and Kakadiaris, I. A. (2011a). 3D facial expression recognition: A perspective on promises and challenges. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 603–610. IEEE.

[Fang et al., 2011b] Fang, T., Zhao, X., Shah, S. K., and Kakadiaris, I. A. (2011b). 4d facial expression recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1594–1601. IEEE.

[Ferri et al., 2011] Ferri, C., Hernández-Orallo, J., and Flach, P. A. (2011). A coherent interpretation of auc as a measure of aggregated classification performance. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 657–664.

[Frome et al., 2004] Frome, A., Huber, D., Kolluri, R., Bülow, T., and Malik, J. (2004). Recognizing objects in range data using regional point descriptors. In *European conference on computer vision*, pages 224–237. Springer.

[Fu and Huang, 2006] Fu, Y. and Huang, T. S. (2006). Graph embedded analysis for head pose estimation. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 6–pp. IEEE.

[Ghiass et al., 2015] Ghiass, R. S., Arandjelović, O., and Laurendeau, D. (2015). Highly accurate and fully automatic head pose estimation from a low quality consumer-level rgb-d sensor. In *Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication*, pages 25–34. ACM.

[Giannakakis et al., 2018] Giannakakis, G., Manousos, D., Simos, P., and Tsiknakis, M. (2018). Head movements in context of speech during

stress induction. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 710–714. IEEE.

[Girard et al., 2013] Girard, J. M., Cohn, J. F., Mahoor, M. H., Mavadati, S., and Rosenwald, D. P. (2013). Social risk and depression: Evidence from manual and automatic facial expression analysis. In *Automatic Face and Gesture Recognition, International Conference and Workshops on*, pages 1–8. IEEE.

[Golub et al., 1996] Golub, G. H. et al. (1996). Cf van loan, matrix computations. *The Johns Hopkins*.

[Guo et al., 2009] Guo, Z., Zhang, Y., Lin, Z., and Feng, D. (2009). A method based on geometric invariant feature for 3D face recognition. In *Image and Graphics, Fifth International Conference on*, pages 902–906. IEEE.

[Isenburg et al., 2001] Isenburg, M., Gumhold, S., and Gotsman, C. (2001). Connectivity shapes. In *Proceedings of the conference on Visualization'01*, pages 135–142. IEEE Computer Society.

[Jain et al., 2007] Jain, V., Zhang, H., and van Kaick, O. (2007). Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling*, 13(01):101–124.

[Jan and Meng, 2015] Jan, A. and Meng, H. (2015). Automatic 3D facial expression recognition using geometric and textured feature fusion. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 5, pages 1–6. IEEE.

[Johnson and Hebert, 1999] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449.

[Karni and Gotsman, 2000] Karni, Z. and Gotsman, C. (2000). Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co.

[Khotanzad and Hong, 1990] Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):489–497.

[Kim and Rossignac, 2005] Kim, B. and Rossignac, J. (2005). Geofilter: Geometric selection of mesh filter parameters. In *Computer Graphics Forum*, volume 24, pages 295–302. Wiley Online Library.

[Klassen and Srivastava, 2006] Klassen, E. and Srivastava, A. (2006). Geodesics between 3D closed curves using path-straightening. In *European conference on computer vision*, pages 95–106. Springer.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

[Langton et al., 2004] Langton, S. R., Honeyman, H., and Tessler, E. (2004). The influence of head contour and nose angle on the perception of eye-gaze direction. *Perception & psychophysics*, 66(5):752–771.

[Lathuilière et al., 2017] Lathuilière, S., Juge, R., Mesejo, P., Muñoz-Salinas, R., and Horaud, R. (2017). Deep mixture of linear inverse regressions applied to head-pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, page 7.

[Lee et al., 2015] Lee, D., Yang, M.-H., and Oh, S. (2015). Fast and accurate head pose estimation via random projection forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1958–1966.

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

[Li and Pedrycz, 2014] Li, D. and Pedrycz, W. (2014). A central profile-based 3D face pose estimation. *Pattern Recognition*, 47(2):525–534.

[Li et al., 2012] Li, H., Chen, L., Huang, D., Wang, Y., and Morvan, J.-M. (2012). 3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2577–2580. IEEE.

[Li et al., 2015a] Li, H., Ding, H., Huang, D., Wang, Y., Zhao, X., Morvan, J.-M., and Chen, L. (2015a). An efficient multimodal 2D+3D feature-based approach to automatic facial expression recognition. *Computer Vision and Image Understanding*, 140:83–92.

[Li et al., 2015b] Li, H., Sun, J., Wang, D., Xu, Z., and Chen, L. (2015b). Deep representation of facial geometric and photometric attributes for automatic 3D facial expression recognition. *arXiv preprint arXiv:1511.03015*.

[Li et al., 2015c] Li, R., Curhan, J., and Hoque, M. E. (2015c). Predicting video-conferencing conversation outcomes based on modeling facial expression synchronization. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE.

[Li et al., 2016] Li, S., Ngan, K. N., Paramesran, R., and Sheng, L. (2016). Real-time head pose tracking with online face template reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1922–1928.

[Lian et al., 2013] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Van Nguyen, H., Ohbuchi, R., et al. (2013). A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition*, 46(1):449–461.

[Liu et al., 2016] Liu, X., Liang, W., Wang, Y., Li, S., and Pei, M. (2016). 3d head pose estimation with convolutional neural network trained on synthetic images. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1289–1293. IEEE.

[Liu et al., 2010] Liu, X., Lu, H., and Li, W. (2010). Multi-manifold modeling for head pose estimation. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 3277–3280. IEEE.

[Lucey et al., 2011] Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., and Matthews, I. (2011). Painful data: The UNBC-McMaster shoulder pain expression archive database. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 57–64. IEEE.

[Lüsi et al., 2016a] Lüsi, I., Escalera, S., and Anbarjafari, G. (2016a). Human head pose estimation on sase database using random hough regression forests. In *Video Analytics. Face and Facial Expression Recognition and Audience Measurement*, pages 137–150. Springer.

[Lüsi et al., 2016b] Lüsi, I., Escarela, S., and Anbarjafari, G. (2016b). SASE: RGB-Depth database for human head pose estimation. In *Computer Vision–ECCV 2016 Workshops*, pages 325–336. Springer.

[Lüsi et al., 2017] Lüsi, I., Jacques Junior, J. C. S., Gorbova, J., Baró, X., Escalera, S., Demirel, H., Allik, J., Ozcinar, C., and Anbarjafari, G. (2017). Joint challenge on dominant and complementary emotion recognition using micro emotion features and head-pose estimation: Databases. In *Automatic Face and Gesture Recognition, 2017. Proceedings. 12th IEEE International Conference on*. IEEE.

[Maalej et al., 2011] Maalej, A., Amor, B. B., Daoudi, M., Srivastava, A., and Berretti, S. (2011). Shape analysis of local facial patches for 3D facial expression recognition. *Pattern Recognition*, 44(8):1581–1589.

[Martin et al., 2014] Martin, M., Van De Camp, F., and Stiefelhagen, R. (2014). Real time head model creation and head pose estimation on consumer depth cameras. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 641–648. IEEE.

[Mateo et al., 2008] Mateo, J. C., San Agustin, J., and Hansen, J. P. (2008). Gaze beats mouse: hands-free selection by combining gaze and emg. In *CHI'08 extended abstracts on Human factors in computing systems*, pages 3039–3044. ACM.

[McDuff et al., 2013a] McDuff, D., El Kaliouby, R., Demirdjian, D., and Picard, R. (2013a). Predicting online media effectiveness based on smile responses gathered over the internet. In *Automatic Face and Gesture*

Recognition, International Conference and Workshops on, pages 1–7. IEEE.

[McDuff et al., 2013b] McDuff, D., Kaliouby, R., Senechal, T., Amr, M., Cohn, J., and Picard, R. (2013b). Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 881–888.

[Meyer et al., 2015] Meyer, G. P., Gupta, S., Frosio, I., Reddy, D., and Kautz, J. (2015). Robust model-based 3D head pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3649–3657.

[Meyer et al., 2003] Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer.

[Murphy-Chutorian et al., 2007] Murphy-Chutorian, E., Doshi, A., and Trivedi, M. M. (2007). Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 709–714. IEEE.

[Murphy-Chutorian and Trivedi, 2009] Murphy-Chutorian, E. and Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626.

[Nealen et al., 2006] Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M. (2006). Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389. ACM.

[Nene et al., 1996] Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (COIL-20).

[Niethammer et al., 2007] Niethammer, M., Reuter, M., Wolter, F.-E., Bouix, S., Peinecke, N., Koo, M.-S., and Shenton, M. E. (2007).

Global medical shape analysis using the laplace-beltrami spectrum. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 850–857. Springer.

[Ovsjanikov et al., 2008] Ovsjanikov, M., Sun, J., and Guibas, L. (2008). Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library.

[Padeleris et al., 2012] Padeleris, P., Zabulis, X., and Argyros, A. A. (2012). Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 42–49. IEEE.

[Pantic, 2009] Pantic, M. (2009). Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3505–3513.

[Pantic and Rothkrantz, 2000] Pantic, M. and Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1424–1445.

[Papazov et al., 2015] Papazov, C., Marks, T. K., and Jones, M. (2015). Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4722–4730.

[Parsons et al., 2017] Parsons, T. D., Gaggioli, A., and Riva, G. (2017). Virtual reality for research in social neuroscience. *Brain sciences*, 7(4):42.

[Peng et al., 2014] Peng, X., Huang, J., Hu, Q., Zhang, S., and Metaxas, D. N. (2014). Head pose estimation by instance parameterization. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1800–1805. IEEE.

[Qiu et al., 2006] Qiu, A., Bitouk, D., and Miller, M. I. (2006). Smooth functional and structural maps on the neocortex via orthonormal

bases of the laplace-beltrami operator. *IEEE Transactions on Medical Imaging*, 25(10):1296–1306.

[Qiu et al., 2008] Qiu, A., Younes, L., and Miller, M. I. (2008). Intrinsic and extrinsic analysis in computational anatomy. *Neuroimage*, 39(4):1803–1814.

[Raytchev et al., 2004] Raytchev, B., Yoda, I., and Sakaue, K. (2004). Head pose estimation by nonlinear manifold learning. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 462–466. IEEE.

[Reale et al., 2013] Reale, M., Zhang, X., and Yin, L. (2013). Nebula feature: A space-time feature for posed and spontaneous 4D facial behavior analysis. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE.

[Reuter, 2010] Reuter, M. (2010). Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2-3):287–308.

[Reuter et al., 2005] Reuter, M., Wolter, F.-E., and Peinecke, N. (2005). Laplace-spectra as fingerprints for shape matching. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 101–106. ACM.

[Reuter et al., 2006] Reuter, M., Wolter, F.-E., and Peinecke, N. (2006). Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design*, 38(4):342–366.

[Reuter et al., 2009] Reuter, M., Wolter, F.-E., Shenton, M., and Niethammer, M. (2009). Laplace–beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design*, 41(10):739–755.

[Riva, 2006] Riva, G. (2006). Virtual reality as communication tool: A sociocognitive analysis. *Virtual Reality*, 8(4).

[Rosenberg, 1997] Rosenberg, S. (1997). *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*, volume 31. Cambridge University Press.

[Ruiz et al., 2015] Ruiz, A., Van de Weijer, J., and Binefa, X. (2015). From emotions to action units with hidden and semi-hidden-task learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3703–3711.

[Ruiz Ovejero et al., 2017] Ruiz Ovejero, A. et al. (2017). *Weakly-supervised learning for automatic facial behaviour analysis*. PhD thesis, Universitat Pompeu Fabra.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. Citeseer.

[Samir et al., 2009] Samir, C., Srivastava, A., Daoudi, M., and Klassen, E. (2009). An intrinsic framework for analysis of facial surfaces. *International Journal of Computer Vision*, 82(1):80–95.

[Sandbach et al., 2012a] Sandbach, G., Zafeiriou, S., and Pantic, M. (2012a). Local normal binary patterns for 3D facial action unit detection. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1813–1816. IEEE.

[Sandbach et al., 2012b] Sandbach, G., Zafeiriou, S., Pantic, M., and Rueckert, D. (2012b). Recognition of 3D facial expression dynamics. *Image and Vision Computing*, 30(10):762–773.

[Sandbach et al., 2012c] Sandbach, G., Zafeiriou, S., Pantic, M., and Yin, L. (2012c). Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697.

[Savran et al., 2008] Savran, A., Alyüz, N., Dibeklioğlu, H., Çeliktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3D face analysis. *Biometrics and identity management*, pages 47–56.

[Savran and Sankur, 2009] Savran, A. and Sankur, B. (2009). Automatic detection of facial actions from 3D data. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1993–2000. IEEE.

[Savran et al., 2012] Savran, A., Sankur, B., and Bilge, M. T. (2012). Comparative evaluation of 3D vs. 2D modality for automatic detection of facial action units. *Pattern recognition*, 45(2):767–782.

[Seemann et al., 2004] Seemann, E., Nickel, K., and Stiefelhagen, R. (2004). Head pose estimation using stereo vision for human-robot interaction. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 626–631. IEEE.

[Shih et al., 2017] Shih, F.-Y., Fan, C.-L., Wang, P.-C., and Hsu, C.-H. (2017). A scalable video conferencing system using cached facial expressions. In *International Conference on Multimedia Modeling*, pages 37–49. Springer.

[Smelser et al., 2001] Smelser, N. J., Baltes, P. B., et al. (2001). *International encyclopedia of the social & behavioral sciences*, volume 11. Elsevier Amsterdam.

[Smith et al., 2008] Smith, K., Ba, S. O., Odobez, J.-M., and Gatica-Perez, D. (2008). Tracking the visual focus of attention for a varying number of wandering people. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1212–1229.

[Srivastava et al., 2011] Srivastava, A., Klassen, E., Joshi, S. H., and Jermyn, I. H. (2011). Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428.

[Stiefelhagen et al., 1999] Stiefelhagen, R., Finke, M., Yang, J., and Waibel, A. (1999). From gaze to focus of attention. In *International Conference on Advances in Visual Information Systems*, pages 765–772. Springer.

[Sukno et al., 2013] Sukno, F., Waddington, J., and Whelan, P. (2013). Rotationally invariant 3D shape contexts using asymmetry patterns. In *GRAPP13*, pages 7–17.

[Sukno et al., 2014] Sukno, F., Waddington, J., and Whelan, P. (2014). Asymmetry patterns shape contexts to describe the 3D geometry of craniofacial landmarks. *Computer Vision, Imaging and Computer Graphics - Theory and Applications. Communications in Computer and Information Science*, 458:19–35.

[Sukno et al., 2012] Sukno, F. M., Waddington, J. L., and Whelan, P. F. (2012). Comparing 3D descriptors for local search of craniofacial landmarks. In *International Symposium on Visual Computing*, pages 92–103. Springer.

[Sukno et al., 2015] Sukno, F. M., Waddington, J. L., and Whelan, P. F. (2015). 3-D facial landmark localization with asymmetry patterns and shape regression from incomplete local features. *IEEE transactions on cybernetics*, 45(9):1717–1730.

[Sun and Yin, 2008] Sun, Y. and Yin, L. (2008). Automatic pose estimation of 3D facial models. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.

[Takallou and Kasaei, 2014] Takallou, H. M. and Kasaei, S. (2014). Head pose estimation and face recognition using a non-linear tensor-based model. *IET Computer Vision*, 8(1):54–65.

[Tan et al., 2018] Tan, D. J., Tombari, F., and Navab, N. (2018). Real-time accurate 3D head tracking and pose estimation with consumer RGB-D cameras. *International Journal of Computer Vision*, 126(2-4):158–183.

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

[Tenenbaum and Freeman, 2000] Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283.

[Tombari et al., 2010] Tombari, F., Salti, S., and Stefano, L. D. (2010). Unique signature of histograms for local surface description. In *European conference on computer vision*, pages 356–369.

[Tulyakov et al., 2014] Tulyakov, S., Vieriu, R.-L., Semeniuta, S., and Sebe, N. (2014). Robust real-time extreme head pose estimation. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2263–2268. IEEE.

[Valle et al., 2016] Valle, R., Buenaposada, J. M., Valdés, A., and Baumela, L. (2016). Head-pose estimation in-the-wild using a random forest. In *International Conference on Articulated Motion and Deformable Objects*, pages 24–33. Springer.

[Vasilescu and Terzopoulos, 2002] Vasilescu, M. A. O. and Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer.

[Venturelli et al., 2016] Venturelli, M., Borghi, G., Vezzani, R., and Cucchiara, R. (2016). Deep head pose estimation from depth data for in-car automotive applications. In *International Workshop on Understanding Human Activities through 3D Sensors*, pages 74–85. Springer.

[Vishwakarma et al., 2007] Vishwakarma, V. P., Pandey, S., and Gupta, M. (2007). A novel approach for face recognition using dct coefficients re-scaling for illumination normalization. In *Advanced Computing and Communications, 2007. ADCOM 2007. International Conference on*, pages 535–539. IEEE.

[Vretos et al., 2011] Vretos, N., Nikolaidis, N., and Pitas, I. (2011). 3D facial expression recognition using zernike moments on depth images. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 773–776. IEEE.

[Vural et al., 2007] Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., and Movellan, J. (2007). Drowsy driver detection through facial movement analysis. In *International Workshop on Human-Computer Interaction*, pages 6–18. Springer.

[Wang et al., 2013a] Wang, B., Liang, W., Wang, Y., and Liang, Y. (2013a). Head pose estimation with combined 2d sift and 3d hog features. In *Image and Graphics (ICIG), 2013 Seventh International Conference on*, pages 650–655. IEEE.

[Wang et al., 2017a] Wang, C., Guo, Y., and Song, X. (2017a). Head pose estimation via manifold learning. In *Manifolds-Current Research Areas*. InTech.

[Wang and Song, 2014] Wang, C. and Song, X. (2014). Robust head pose estimation via supervised manifold learning. *Neural Networks*, 53:15–25.

[Wang et al., 2012a] Wang, H., Su, Z., Cao, J., Wang, Y., and Zhang, H. (2012a). Empirical mode decomposition on surfaces. *Graphical Models*, 74:173–183.

[Wang et al., 2014a] Wang, J., Wang, S., Huai, M., Wu, C., Gao, Z., Liu, Y., and Ji, Q. (2014a). Capture expression-dependent AU relations for expression recognition. In *Multimedia and Expo Workshops, International Conference on*, pages 1–6. IEEE.

[Wang et al., 2018] Wang, K., Wu, Y., and Ji, Q. (2018). Head pose estimation on low-quality images. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 540–547. IEEE.

[Wang et al., 2017b] Wang, M., Panagakis, Y., Snape, P., Zafeiriou, S., et al. (2017b). Learning the multilinear structure of visual data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4592–4600.

[Wang et al., 2014b] Wang, W., Zheng, W., and Ma, Y. (2014b). 3D facial expression recognition based on combination of local features and globe information. In *Intelligent Human-Machine Systems and Cybernetics, Sixth International Conference on*, volume 2, pages 20–25. IEEE.

[Wang et al., 2012b] Wang, X., Wang, L., and Qiao, Y. (2012b). A comparative study of encoding, pooling and normalization methods for action recognition. In *Asian Conference on Computer Vision*, pages 572–585. Springer.

[Wang and Meng, 2013] Wang, Y. and Meng, M. (2013). 3D facial expression recognition on curvature local binary patterns. In *Intelligent Human-Machine Systems and Cybernetics, International Conference on*, volume 2, pages 123–126. IEEE.

[Wang et al., 2013b] Wang, Y., Meng, M., and Zhen, Q. (2013b). Learning encoded facial curvature information for 3D facial emotion recognition. In *Image and Graphics (ICIG), 2013 Seventh International Conference on*, pages 529–532. IEEE.

[Wang and Yang, 2017] Wang, Z. and Yang, X. (2017). V-head: Face detection and alignment for facial augmented reality applications. In *International Conference on Multimedia Modeling*, pages 450–454. Springer.

[Xie et al., 2010] Xie, S., Shan, S., Chen, X., and Chen, J. (2010). Fusing local patterns of gabor magnitude and phase for face recognition. *Transactions on Image Processing*, 19(5):1349–1361.

[Xue et al., 2015] Xue, M., Mian, A., Liu, W., and Li, L. (2015). Automatic 4D facial expression recognition using dct features. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 199–206. IEEE.

[Yang et al., 2015] Yang, X., Huang, D., Wang, Y., and Chen, L. (2015). Automatic 3D facial expression recognition using geometric scattering representation. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE.

[Yin et al., 2008] Yin, L., Chen, X., Sun, Y., Worm, T., and Reale, M. (2008). A high-resolution 3D dynamic facial expression database. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE.

[Yin et al., 2006] Yin, L., Wei, X., Sun, Y., Wang, J., and Rosato, M. J. (2006). A 3D facial expression database for facial behavior research. In *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*, pages 211–216. IEEE.

[Yu et al., 2017] Yu, Y., Mora, K. A. F., and Odobez, J.-M. (2017). Robust and accurate 3D head pose estimation through 3dmm and online head model reconstruction. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 711–718. IEEE.

[Yun and Guan, 2010] Yun, T. and Guan, L. (2010). Human emotion recognition using real 3D visual features from gabor library. In *Multimedia Signal Processing, International Workshop on*, pages 505–510. IEEE.

[Zahn and Roskies, 1972] Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281.

[Zeng et al., 2013] Zeng, W., Li, H., Chen, L., Morvan, J.-M., and Gu, X. D. (2013). An automatic 3D expression recognition framework based on sparse representation of conformal images. In *Automatic Face and Gesture Recognition, International Conference and Workshops on*, pages 1–8. IEEE.

[Zeng et al., 2009] Zeng, Z., Pantic, M., Roisman, G. I., and Huang, T. S. (2009). A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58.

[Zhang et al., 2015] Zhang, H., El-Gaaly, T., Elgammal, A., and Jiang, Z. (2015). Factorization of view-object manifolds for joint object recognition and pose estimation. *Computer Vision and Image Understanding*, 139:89–103.

[Zhang et al., 2010] Zhang, H., Van Kaick, O., and Dyer, R. (2010). Spectral mesh processing. In *Computer graphics forum*, volume 29, pages 1865–1894. Wiley Online Library.

[Zhao et al., 2015] Zhao, K., Chu, W.-S., De la Torre, F., Cohn, J. F., and Zhang, H. (2015). Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216.

[Zhen et al., 2016] Zhen, Q., Huang, D., Wang, Y., and Chen, L. (2016). Muscular movement model-based automatic 3D/4d facial expression recognition. *IEEE Transactions on Multimedia*, 18(7):1438–1450.

[Zhu et al., 2014] Zhu, Y., Xue, Z., and Li, C. (2014). Automatic head pose estimation with synchronized sub manifold embedding and random regression forests. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(3):123–134.