Rafael Alves Duarte

Identity Recognition using Facial Appearance and Shape Information

September, 2016

· U C ·

Universidade de Coimbra

Department of Electrical and Computer Engineering,
Faculty of Sciences and Technology, University of Coimbra,
3030-290 COIMBRA, PORTUGAL.

A Dissertation for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

Identity Recognition using Facial Appearance and Shape Information

Rafael Alves Duarte

Supervisor:
Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista

Co-Supervisor:
Doutor Pedro Alexandre Dias Martins

Jury:
Prof. Doutor Paulo José Monteiro Peixoto
Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista
Prof. Doutor João Pedro de Almeida Barreto

September, 2016

In loving memory of my grandparents Olímpia and Ricardo

Acknowledgements

First and foremost, I would like to thank André Ferrerinha, António Simões, João Toscano for the great atmosphere, support and mutual aid in the lab, with a special thanks to Pedro Martins for is counseling, dedication and work ethic.

To my supervisor Prof. Jorge Batista for the orientation and availability demonstrated during the development of this work.

To my friends for the endless care and support, it is a joy to have you by my side and to my girlfriend, thank you for always being by my side and support me in difficult times.

And on a special note I would like to thank my parents. Mãe e Pai, não há palavras para descrever o sentimento que tenho por vocês, sempre me amaram, apoiaram e acreditaram em mim, um muito obrigado.

Abstract

This thesis focus on identity recognition through RGBD images and for that effect two different methods will be used simultaneously: Constrained Local Models (CLM) and Sparse Coding.

The CLMs are popular method designed to locate characteristics points in faces. They combine a set of local detectors (or patch experts), one for each landmark, with a global optimization strategy that maximize the score of all the detectors simultaneous.

In this thesis, it was developed an extension of the Constrained Local Models algorithm that allows the use of multiples sources of information, like for example data from RGBD sensors.

This type of information allow us to combine traditional RGB images with depth images with the intention of mitigating some drawbacks of traditional methods an augment his robustness, since the depth images are insensible to lightning conditions, reflect pure geometry and contours of the shape.

In this context, it was introduced the possibility to use normal vectors, as depth descriptor, in the CLM formulation. In practice are use linear detectors which resort to the Histogram of Oriented Normal Vectors (HONV).

Relatively to facial recognition, it will be used sparse coding, due to is rising popularity with the scientific community of Computers Vision, justified by its capabilities of both representation and classification, minimizing the computational overhead and its ability to deal with occlusion. The sparse coding is based in, using a prior leaned dictionary, finding the sparse linear combination of atoms, or elements, that most accurately recreate the original image. This will allow to validate the existence of an individual in the dictionary (database) thus making the sparse coding a potential candidate to perform the desired facial recognition.

Both methods will be developed and tested in OpenCV/C++ along side with a Kinect camera, since it has the ability to capture both color images and depth images. To achieve the objective a photo database will be created, using nineteen individuals, where each one will be asked to make seven facial expressions and five different poses. This will be used in order to create the deformable face models, and in parallel, will originate a dictionary in order to enable the computer to make a face matching.

This dissertation aims to verify if there is improvements in the alignment process compared with the traditional RGB images; see if it is possible to combine alignment methods with this kind of facial recognition; and finally, testing the feasibility of a developed system in real-time to perform both processes.

Keywords:

Facial recognition, RGB, RGBD, Facial alignment, CLM, Histogram of Oriented Normal Vectors, OpenCV, Sparse Coding, Real-Time, Depth Image, Kinect

Resumo

Esta tese centra-se no reconhecimento de identidade através de imagens RGBD, sendo que para tal vão ser utilizados dois métodos distintos em simultâneo: 'Constrained Local Models' (CLM) e Codificação Esparsa.

Os CLMs são métodos populares desenhados para localizar os pontos característicos de cada face. Estes combinam um conjunto de detectores locais, um para cada ponto, com uma estratégia de optimização global, que maximiza a pontuação de todos os detectores em simultâneo.

Pretende-se desenvolver uma extensão do algoritmo 'Constrained Local Models' que permite a utilização de múltiplas fontes de informação, entre as quais dados provenientes de sensores RGBD. Este tipo de informação permite combinar as tradicionais imagens RGB com imagens de profundidade, tendo como intuito mitigar algumas lacunas dos métodos tradicionais e aumentar a sua robustez, uma vez que as imagens de profundidade são insensíveis às condições de luminosidade, reflectem geometria e contornos da forma. Neste contexto, introduziu-se a possibilidade de utilizar vectores normais, como descritor de profundidade, na formulação CLM. Na pratica utilizar-se-ão detectores lineares que recorrem a Histograma Orientado a Vectores Normais (HONV).

No que recai sobre reconhecimento facial, irá ser utilizada a codificação esparsa, devido à sua crescente popularidade junto da comunidade cientifica de Visão por Computadores, devido tanto às suas capacidades de representação, de classificação, de minimização de sobrecarga computacional, assim como à sua capacidade de lidar com oclusão. A codificação esparsa baseia-se na utilização de um dicionário aprendido à priori, para encontrar uma combinação linear esparsa de átomos, ou elementos, que mais fielmente recriam a imagem original. Será deste modo possível validar a existência de um individuo no dicionário (base de dados) tornando, assim, a codificação esparsa num potencial candidato para efectuar o desejado reconhecimento facial.

Ambos os métodos serão desenvolvidos e testados em OpenCV/C++ juntamente com uma câmara Kinect, uma vez que esta tem a capacidade de capturar tanto imagens a cores como de profundidade. Para concretizar o objectivo criar-se-á uma base de dados de fotos, utilizando uma amostra de dezanove voluntários, em que a cada um será solicitado que faça sete expressões faciais e cinco poses diferentes. Esta será usada com o intuito de criar os modelos faciais deformáveis, e paralelamente, originará um dicionário com o intuito de habilitar o computador a fazer uma correspondência facial.

Esta dissertação tem como objectivo verificar se há melhorias no processo de alinhamento em comparação com as imagens tradicionais de RGB; verificar se é possível combinar métodos de alinhamento com este género de reconhecimento facial; e por último, testar a viabilidade de um sistema desenvolvido em tempo real que efectua ambos os processos.

Palavras-Chave:

Reconhecimento Facial, RGB, RGBD, Alinhamento Facial, CLM, Histograma Orientado a Vectores Normais, OpenCV, Codificação Esparsa, Tempo-Real, Imagem de Profundidade, Kinect

Contents

List of Figures

List of Tables

Acronyms

AAM - Active Appearance Model

ASM - Active Shape Model

CLM - Constrained Local Model

DBASM - Discriminative Bayesian Active Shape Models

HOG - Histogram of Oriented Gradients

HONV - Histogram of Oriented Normal Vectors

LDS - Linear Dynamic System

MAP - Maximum A-*Posteriori*

MDF - Multi-Dimensional Correlation Filter

MOSSE - Minimum Output Sum of Squared Error

OMP - Orthogonal Matching Pursuit

PAW - Piecewise Affine Warp

PCA - Principal Component Analysis

PDM - Point Distribution Model

ROI - Region Of Interest

SR - Sparse Representation

Mathematical Nomenclature

The mathematical notation used in this thesis is:

Matrices are represented as bold capital letters $\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$

Vectors are represented as bold lower-case letters $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$

A two dimensional point is represented as $\mathbf{x} = [x, y]$

A three dimensional point is represented as $\mathbf{x}_{3D} = [x, y, z]$

The Euclidean space of k-dimensions is represented as $\Re^k$

An Patch expert, or local detector is represented as $\mathcal{P}$

The Fourrier transform is represented as $\mathcal{F}$

A Piecewise Affine Warp is represented as $\mathcal{W}$

An over-complete dictionary is represented as $\mathcal{D}$

$\mathbf{I}_n$ - Identity matrix of n dimensions

$\circledast$ - Correlation operator

$\bullet$ - Complex conjugate

$\odot$ - Hadamard product

^{H} - Conjugate transpose

Φ – is a matrix whose columns are eigenvectors

Λ – diagonal matrix of eigenvalues

λ - eigenvalue

Chapter 1

Introduction

Nowadays a dramatical increase in the interaction between man and machine due to the use of new technologies such as the computer and phone, utensils essential to daily life, create a new needs that must be addressed. In a world where crime increases it is necessary to improve the security methods. The evolution of facial recognition comes as a promising response, being one of the most successful applications in biometrics *e.g.* smart houses, smart car and security. Although facial recognition may not be the most reliable and efficient biometric technique it has one key advantage: it doesn't require the cooperation of the test subject.

Well positioned systems in public places such as airports, train stations, borders control and ATM machines for example, may allow us to identify individuals among the crowds without their awareness. Due to its easy integration in existing system it can be easily explored in several places, such as: Criminal justice systems, where it can be use in Mug-shot/booking systems by searching an image in a database; and the Citizen Card, where a database of facial images is created by recurring to the photos taken when the card is created.

Due to the recent emerge of low-cost depth sensors such as Kinect, researchers are now allowed to revisit unsolved problems such as object detection or recognition, since that when compared to traditional RGB method depth maps are insensitive to change in conditions.

This thesis consists in taking advantage of both RGB and Depth data for face alignment and using the model parameters to perform face recognition with sparse representation using images since this kind of mass identification is only achieved by facial recognition.

1.1 Main Contributions

A variety of non-rigid face models have been proposed in the past years, boosting the performance of face alignment to a new level. The Constrained Local Model (CLM) [8][29][25][18][15][21] is one of the state-of-the-art alignment methods, using a based appearance model, which generates a likelihood map for each landmark. Once the face model is obtained, the first step is to find the face model parameters that maximize the match between the model and the input image. Fitting a CLM can be seen as a two step method: 1) produce likelihood maps (response maps) using detectors, for each landmark; 2) global optimization strategy that maximize all the scores from detection while imposing a pertinent shape.

Despite the fact that most CLM strategies aim to approximate the responses maps by simple parametric form or non-parametric it has proposed a discriminative CLM framework, the Bayesian Constrained Local Model (BCLM) [18][16][21]. But like any other CLM method, these struggle in the existence of extreme lightening conditions, since the current methodologies focus mainly on color or gray scale image. Having this in mind Martins [17] proposed a extension of the MOSSE filter where depth can used as an additional likelihood source, since depth images reflect pure geometry, contours of the shape and are insensitive to changes in lightning conditions which make them a excellent candidate to mitigate those weakness from conventional CLM methods. To capture the 3D characteristics it was adopted the feature proposed by Tang *et al.* [26], *Histogram of Oriented Normal Vectors* (HONV), designed specifically to capture local 3D geometric characteristics as the depth images include local information such as curvature and normals.

At the same time Sparse representations have received a great deal of attention. by searching for the most compact representation of a signal in terms of linear combination of atoms in an over-completed dictionary. It can be boiled down to three aspects: the applications for different tasks, such as signal separation, denoising, coding, image inpainting or even face recognition [30][32][13]; design of the dictionary, such as the K-SVD method [1]; pursuit methods for solving the optimization problem, such as Orthogonal Matching Pursuit (OMP). The goal in using sparse representation is to verify the existence of a captured face in the dictionary with the purpose of identity recognition.

1.2 Structure of the thesis

This thesis is divided into three main parts where all the techniques implemented for facial recognition will be presented and detailed, followed by the experiments and results and then conclusion. This following Chapter briefly presents information about the subject. Chapter 3 presents the theory behind Constrained Local Model giving a short explanation, serving as background for the understanding of the rest the thesis. In Chapter 4 depth maps will be analyzed and how to use them in the fitting process. As for Chapter 5 a classification technique will be approached providing an insight on how it works. The experiments and results will be discuss in Chapter 6 and how the implementations influence the performance. Finally, the conclusions will be presented in Chapter 7.

Chapter 2

State of the Art

Since two separate topics will be use to implement Face Recognition, Face Alignment and Sparse Coding, the state of the art will distinctively approach each of the subjects

2.1 Deformable Face Models

In the current literature a large number of face models methods have been proposed, which can be roughly grouped in two main classes: generative methods and discriminative methods.

In the field of face alignment (localizing facial features), where faces are seen as deformable objects which can vary in terms of appearance and shape, where the Active Appearance Models (AAMs) are arguably the most well-known generative method. The AAMs, proposed in 2001 by Cootes *et al.*[5], are statical parametric models that combine shape and appearance in a single unified model. Matching an AAM involves finding the model parameters that minimize the difference between the target image and the synthesized model. Posteriorly, Baker and Matthews [19] presented an efficient fitting algorithm based on the Inverse Compositional Image Alignment (ICIA). Xiao *et al.*[31] also proposed a real time hybrid solution, describing how a non-rigid structure-from-motion algorithm originates a 3D shape that constrains a 2D AAM.

Lastly, in 1999, Blanz and Vetter [2] suggested a dense 3D model, the morphable face model (3DMM), which later was extended by Romdhani and Vetter [27] to also use the ICIA algorithm.

The discriminative methods learn a local appearance model (detector) for each facial landmark and a shape model that impose a global constrain on the model. In 1995, Cootes *et al.*[6] proposed the Active Shape Model (ASM) that use the shape constraints to find the best location of the feature. Later the Constrained Local Model (CLM) [7] originated a unified model that uses a patch based texture model constrained by a shape model similar to the ASM. The main difference between the CLMs and the AAMs is that the appearance is obtained from patches samples around each landmark.

In general, CLMs methods differ in the optimization algorithm that aims to find the true likelihood maps of each landmark: Convex quadratic fitting (CQF) proposed by Wang [29], which enforces the convexity of the patch response surfaces; Gu and Kanade [12] also presented a model using Gaussian Mixture (GMM); or nonparametric models proposed by Saragih *et al.*[24] that explores the subspace constrained mean-shifts (SCMS). Recently, the objective function was formulated in terms of Bayesian inference, Paquet [21] extended the CQF to a maximum a posterior and more presently, Martins *et al.*[18] proposed a efficient Bayesian fitting model considering a Linear Dynamic System (LDS) to include the second order estimates of the CLM parameters.

2.2 Sparse Coding

A robust face recognition is one of the most challenging tasks for researchers from several communities, *e.g.* computer vision and artificial intelligence, making the sparse representation one of the most promising methods to approach the problem. The idea behind sparse coding [20] is that using a prior learned dictionary it is possible to find a sparse linear combination of atoms, or elements, that most faithfully recreate the original image.

Aharon *et al.*[1] presented a dictionary learning algorithm, K-SVD, that ensured discriminative criteria. Later, Pham *et al.*[22] proposed a joint framework of dictionary construction and classification where the class label and the classification error are considered *a posteriori*. Inspired by the success of sparse representation in face recognition achieved by Wright *et al.*[30], Zhang *et al.*[33] presented discriminative K-SVD (D-KSVD) method, incorporating the labels directly into the sparse coding. In 2013, Jiang *et al.*[13] extended D-KSVD by integrating both labels and classification error achieving impressive performances in face recognition making it a state of the art algorithm.

Chapter 3

Constrained Local Model

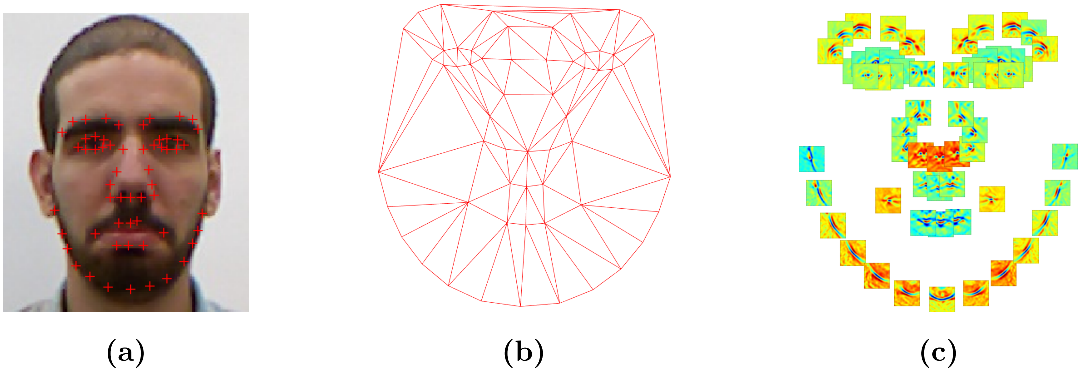
The Constrained Local Model [18][15][29][8][25] (CLM) consists of a collection of v patch experts denoted as $\mathbf{h}_i$, $i = 1 \dots v$, one for each landmark, which are afterwards regularized by a linear shape model. This method combines both shape and appearance constraints to find the best location of each landmark. Fitting a CLM is usually iterative process: 1) convolving the local detectors with the image to generate likelihood (or response) maps for each landmark; 2) regularization step (global optimization that ensures a valid face).

This chapter serves as background information for a better understanding of the rest of the thesis.

3.1 The Shape Model

According to Cootes *et al.*[6] the facial landmarks are predominantly located around facial components such as eyes, mouth, nose and chin, and they are label. Regardless the number of landmarks, they must cover those mentioned areas since they transmit information for both discriminative and generative methods.

Concatenating the points of Figure 3.1a, a shape $\mathbf{s} = (x_1, y_1, \dots, x_v, y_v)^T$ can be created where v is the total number of landmarks. To create a shape model, Figure 3.1b, one must apply Procrustes Analysis [4] to a sufficient number of images with the corresponding landmarks manually labeled, in order to obtain the following linear model:



(a) (b) (c)

Fig. 3.1: Figure 3.1a represents the locations of all landmarks. Figure 3.1b is shape model example. Figure 3.1c illustrate the local landmarks detectors (Image courtesy of Martins, Pedro.)

$$\mathbf{s}' = \mathbf{s}_0 + \sum_{i=1}^{n} \phi_i \mathbf{b}_i \tag{3.1}$$

where the vector $\mathbf{s}_0$ represents the mean shape, ϕ_i and $\mathbf{b}_i$ represent the i^{th} shape parameter and shape basis of the PDM respectively.

If $\mathcal{S}(., \mathbf{q})$ represents the similarity transformation, the previous equation can be extended as:

$$\mathbf{s} = \mathcal{S}(\mathbf{s}_0 + \Phi\mathbf{b}, \mathbf{q}) \tag{3.2}$$

considering that Φ is the shape subspace matrix holding n eigenvectors and that $\mathbf{q}$ are the pose parameters, that is, scale, rotation and translation ($\mathbf{q} = \begin{bmatrix} s\ \cos\theta - 1 & s\ \sin\theta & t_x & t_y \end{bmatrix}^T$).

3.2 Patch Experts

As previously mentioned, the CLM model consists in v patch experts (Figure 3.1c), or local detectors, $\mathcal{P}$ trained for each facial landmark. Most works use support vector machines (SVMs) to train the local detectors, however some recent patch experts have gain emphasis due to improved performance, such as the Minimum Output Sum of Squared Errors (MOSSE) Filters [15]

The correlation of each MOSSE Filter [15], $\mathbf{h}_i$, can be computed by solving the following linear regression problem:

$$\arg\min_{\mathbf{h}_i} \sum_{j=1}^{N} (\mathbf{h}_i \circledast \mathbf{I}_j - \mathbf{g}_j)^2 + \lambda \left\| \mathbf{h}_i \right\|^2 \tag{3.3}$$

where $\circledast$ is to the correlation operator, $\mathbf{I}_j$ to the j^{th} training patch of N total, $\mathbf{g}_j$ the desired target correlation and λ is a regularization parameter.

Due to the 2D Fourier transform properties, the solution can be rewritten through element-wise multiplication:

$$\mathbf{h}_i = \mathcal{F}^{-1} \left\{ \left(\sum_{j=1}^{N} \mathcal{F}\left\{\mathbf{I}_j\right\} \odot \mathcal{F}\left\{\mathbf{I}_j\right\}^{\bullet} + \lambda \right)^{-1} \cdot \left(\sum_{j=1}^{N} \mathcal{F}\left\{\mathbf{g}_j\right\} \odot \mathcal{F}\left\{\mathbf{I}_j\right\}^{\bullet} \right) \right\}^{\bullet} \tag{3.4}$$

where $\mathcal{F}$ the 2D Fourier transform; $\odot$ the Hadamard product; and $\bullet$ the complex conjugate. Each correlation filter is highly stable since the filter maps the aligned detector to an output, $\mathbf{g}$, centered at the facial landmark.

3.3 Model alignment

Given an input image, the objective is to minimize the difference between the model and the corresponding landmarks of the image. Although under a Bayesian approach the model fitting can be seen as probabilistic method. In a maximum *a-posteriori* (MAP) formulation the optimal shape parameters ($\mathbf{b}_s^*$) can be obtain through:

$$\mathbf{b}_s^* = \arg\max_{\mathbf{b}_s} p(\mathbf{b}_s|\mathbf{y}) \tag{3.5}$$

being

$$p(\mathbf{b}_s|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{b}_s)p(\mathbf{b}_s) \tag{3.6}$$

where $\mathbf{y} \in \Re^{2v}$ is the shape measurement, $p(\mathbf{y}|\mathbf{b}_s)$ is the likelihood term (extracted from the response maps) and $p(\mathbf{b}_s)$ is the prior distribution term that detains all possible shape configurations.

If one assumes that each landmark is conditional independent and that exists an approximate solution to the real parameters ($\mathbf{b} \approx \mathbf{b}_s^*$) Equation 3.6 can be rewritten as:

$$p(\mathbf{b}|\mathbf{y}) \propto \left(\prod_{i=1}^{v} p(\mathbf{y}_i|\mathbf{b}) \right) p(\mathbf{b}|\mathbf{b}_{k-1}^*) \tag{3.7}$$

being $\mathbf{y}_i$ is the i^{th} landmark coordinates and $\mathbf{b}^*_{k-1}$ is the previous optimal estimate of $\mathbf{b}$.

3.3.1 The Likelihood Term

Martins *et al.*[16] formulated the objective function as maximum *a-posteriori* of shape parameters, so Equation 3.1 follows a Gaussian distribution given by:

$$p(\mathbf{y}|\mathbf{b}) \propto \exp\left(-\frac{1}{2}(\underbrace{\mathbf{y}-(\mathbf{s}_0}_{\Delta\mathbf{y}} + \Phi\mathbf{b}))^T\, \Sigma_{\mathbf{y}}^{-1}(\mathbf{y}-(\mathbf{s}_0+\Phi\mathbf{b}))\right) \tag{3.8}$$

where $\Delta\mathbf{y}$ is the difference between the observed and the mean shape and $\Sigma_{\mathbf{y}}$ can be estimated from response maps since it represent the uncertainty of the spatial localization of all landmarks.

So, in probabilistic framework, the likelihood term can be seen as:

$$p(\mathbf{y}|\mathbf{b}) \propto \mathcal{N}(\Delta\mathbf{y}|\Phi\mathbf{b}, \Sigma_{\mathbf{y}}) \tag{3.9}$$

3.3.2 The Prior Term

Like the likelihood term, the prior term also follows a Gaussian distribution. In this optimization the prior can be obtained by:

$$p(\mathbf{b}_k|\mathbf{b}_{k-1}) \propto \mathcal{N}(\mathbf{b}_k|\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) \tag{3.10}$$

where $\mu_{\mathbf{b}} = \mathbf{b}_{k-1}$ along with $\Sigma_{\mathbf{b}} = \Lambda + \Xi$. The Λ is the diagonal covariance matrix ($\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, where λ_i denotes the i^{th} PCA eigenvalue) and Ξ, which can be computed off-line, is the additive dynamic noise covariance.

3.3.3 The Posterior Term

An important property of the Bayes theorem is that, when the likelihood and prior are Gaussian distributions the posterior is also Gaussian [21].

Considering $p(\mathbf{b}_k|\mathbf{b}_{k-1})$ a prior Gaussian distribution over $\mathbf{b}_k$ and $p(\mathbf{y}|\mathbf{b}_k)$ a likelihood Gaussian distribution, the posterior distribution can be written as:

$$p(\mathbf{b}_k|\mathbf{y}) \propto \mathcal{N}(\mathbf{b}_k|\mu, \mathbf{\Sigma}) \tag{3.11}$$

with

$$\mathbf{\Sigma} = (\Sigma_{\mathbf{b}}^{-1} + \Phi^T \Sigma_{\mathbf{y}}^{-1} \Sigma)^{-1} \tag{3.12}$$

$$\mu = \mathbf{\Sigma}(\Phi^T \Sigma_{\mathbf{y}}^{-1} \mathbf{y} + \Sigma_{\mathbf{b}}^{-1} \mu_{\mathbf{b}}) \tag{3.13}$$

2$^{\text{nd}}$ Order Estimate

The previous approach can be improved by taking into account the confidence in the current parameters, *i.e.* to model the covariance of the latent variables $\mathbf{b}_k$. Having regard to the above, this approach was extended [16][18] to include second order estimates of the shape parameters. It was formulated in terms of a Linear Dynamic System (LDS) where it estimates posterior Gaussian distribution using Gaussian shape measurements and a linear process recursively.

The state and measurement equations of the LDS can be written as:

$$\mathbf{b}_k = \mathbf{I}_n \mathbf{b}_{k-1} + q \tag{3.14}$$

$$\Delta \mathbf{y} = \Phi \mathbf{b}_k + r \tag{3.15}$$

where $q \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{b}})$ is the additive dynamic noise, $\Delta\mathbf{y} = \mathbf{y} - \mathbf{s}_0$ is the observed shape deviation from the mean, $r \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{y}})$ is the additive measurement noise. The previous estimated shape $\mathbf{b}_{k-1}$ relates to the current parameters $\mathbf{b}_k$ by an identity relation $\mathbf{I}_n$ plus noise [18].

The LDS denote the posterior distribution of the form:

$$p(\mathbf{b}_k | \mathbf{y}_k, \ldots, \mathbf{y}_0) \propto \mathcal{N}(\mathbf{b}_k | \mu_k^{\mathbf{F}}, \mathbf{\Sigma}_k^{\mathbf{F}}) \tag{3.16}$$

with the posterior mean $\mu_k^{\mathbf{F}}$ and covariance $\mathbf{\Sigma}_k^{\mathbf{F}}$ given by the Kalman Filter [18][14]:

$$\mathbf{K} = \mathbf{P}_{k-1} \Phi^T (\Phi \mathbf{P}_{k-1} \Phi^T + \Sigma_{\mathbf{y}})^{-1} \tag{3.17}$$

$$\mu_k^{\mathbf{F}} = \mathbf{I}_n \mu_{k-1}^{\mathbf{F}} + \mathbf{K}(\mathbf{y} - \Phi \mathbf{I}_n \mu_{k-1}^{\mathbf{F}}) \tag{3.18}$$

$$\Sigma_{\mathbf{y}}^{\mathbf{F}} = (\mathbf{I}_n - \mathbf{K}\Phi)\mathbf{P}_{k-1} \tag{3.19}$$

Finally, the optimal shape parameters that maximize the objective, in Equation 3.5, are given by posterior distribution $\mathbf{b}^* = \mu_k^{\mathbf{F}}$.

In conclusion, the model alignment can be seen as an iterative process: 1) requires a shape using the Equation 3.1; 2) warp the image using the pose parameters; 3) evaluate the response maps in each landmark; 4) extract the likelihood parameters $(\mathbf{y}, \Sigma_{\mathbf{y}})$ and obtain a new shape using LDS.

Chapter 4

CLM and Depth Data

This chapter falls on improving the robustness of the patch experts although the impressive success achieved by leading methods [18][15][29][25], but even them have the task hampered due to the extreme lightening conditions. However, depth data might mitigate some of the drawbacks and even improve the performance since depth image contains rich surface information and it can be explored with a powerful image descriptor such as the *Histogram of Oriented Normal Vectors* (HONV) [26]. Therefore, an extended CLM [17] formulation taking advantage of depth image is addressed.

4.1 Histogram of Oriented Normal Vectors

According to Tang *et al.*[26], the HONV was designed specifically to capture local 3D geometric characteristics for object recognition with depth sensor where it is possible to recognize the object category by capturing the orientation of its normal vector, or tangent plane, at every surface point.

The local 3D geometry characteristics can be represented by the local distribution of the normal vector orientation and from the 3D coordinate of a surface point $\mathbf{x}_{3D} = (x, y, \mathbf{d}(x, y))$, being $\mathbf{d}(x, y)$ the depth value acquired by the depth sensor, the normal vector orientation can be represented as a tuple of azimuthal angle and zenith angle.

4.1.1 Normal Vector

Supposing a point $\mathbf{x} = (x, y)$ in the domain of surface function $w = \mathbf{d}(x, y)$, the normal vector can be obtained by using the cross product of two tangent vectors on the tangent plane:

$$\mathbf{n} = \mathbf{g}_x \times \mathbf{g}_y \tag{4.1}$$

where $\mathbf{g}_x = \frac{\partial}{\partial x}\begin{bmatrix} x & y & \mathbf{d}(x,y) \end{bmatrix}^T$ is the tangent vector in x and $\mathbf{g}_y = \frac{\partial}{\partial y}\begin{bmatrix} x & y & \mathbf{d}(x,y) \end{bmatrix}^T$ in the y direction.

Thus the normal vector $\mathbf{n}$ at $\mathbf{x}$ can be computed using the cross product in Equation 4.1:

$$\begin{aligned}
\mathbf{g}_x \times \mathbf{g}_y &= \frac{\partial}{\partial x}\begin{bmatrix} x \\ y \\ \mathbf{d}(x,y) \end{bmatrix} \times \frac{\partial}{\partial y}\begin{bmatrix} x \\ y \\ \mathbf{d}(x,y) \end{bmatrix} \\
&= \begin{bmatrix} 0 & -\frac{\partial \mathbf{d}(x,y)}{\partial x} & 0 \\ \frac{\partial \mathbf{d}(x,y)}{\partial x} & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \frac{\partial \mathbf{d}(x,y)}{\partial y} \end{bmatrix} \\
&= \begin{bmatrix} -\frac{\partial \mathbf{d}(x,y)}{\partial x} \\ -\frac{\partial \mathbf{d}(x,y)}{\partial y} \\ 1 \end{bmatrix}
\end{aligned} \tag{4.2}$$

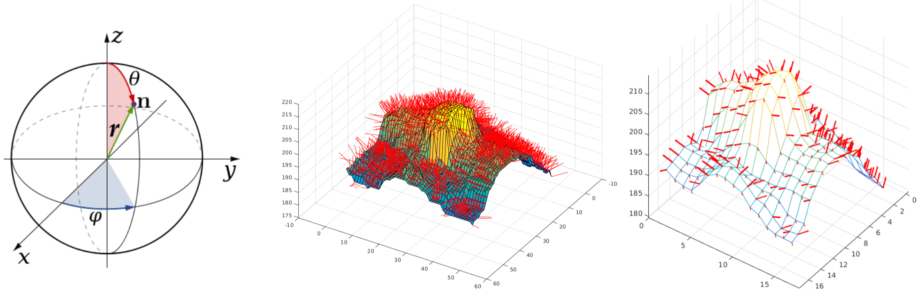
using the finite difference approximation $\frac{\partial \mathbf{d}(x,y)}{\partial x}$ and $\frac{\partial \mathbf{d}(x,y)}{\partial y}$ are given by:

$$\begin{aligned}
\frac{\partial \mathbf{d}(x,y)}{\partial x} &\approx \frac{1}{2}(\mathbf{d}(x+1,y) - \mathbf{d}(x-1,y)) \\
\frac{\partial \mathbf{d}(x,y)}{\partial y} &\approx \frac{1}{2}(\mathbf{d}(x,y+1) - \mathbf{d}(x,y-1))
\end{aligned} \tag{4.3}$$

Since the depth image can be processed like a image in gray level, the terms $\frac{\partial \mathbf{d}(x,y)}{\partial x}$ and $\frac{\partial \mathbf{d}(x,y)}{\partial y}$ can be obtain with a gradient computation.

4.1.2 HONV Feature

As mentioned, the normal vector can be represented by a tuple of zenith and azimuth angles since the spherical coordinates (θ, φ, r), Figure 4.1a, encode the orientation information better than Cartesian coordinates. According to [26] it is redundant to consider the radius once θ and φ are sufficient to describe the normal vector.



(a) **(b)** **(c)**

Fig. 4.1: Figure 4.1a represents a normal vector $\mathbf{n}$ with zenith angle θ and azimuthal angle φ. Figure 4.1b represents all the surface normals. The HONV Feature of a nose is illustrated in Figure 4.1c, where only the most voted normal in each *cells* is considered

Using the following equation and the depth gradient's components it is possible to calculate θ and φ orientation at each pixel's:

$$\begin{aligned} \varphi &= \tan^{-1}\left(\frac{\partial \mathbf{d}(x,y)}{\partial y} \Big/ \frac{\partial \mathbf{d}(x,y)}{\partial x}\right) \\ \theta &= \tan^{-1}\left(\left(\frac{\partial \mathbf{d}(x,y)}{\partial y}\right)^2 + \left(\frac{\partial \mathbf{d}(x,y)}{\partial y}\right)^2\right)^{\frac{1}{2}} \end{aligned} \tag{4.4}$$

The HONV is in very ways similar to the well engineered *Histogram of Oriented Gradients* (HOG) [9] since it requires an image decomposition into sections of $c \times c$ pixels named *cells*. Each *cells* will be described by two histograms (θ and φ). An histogram can have as many orientation bins as desired between 0° and 180° or 360°. Each pixel of a *cells* will cast a weighted vote since they are interpolated bilinearly between the neighboring bins. Furthermore, a 2D Gaussian smoothing is applied to adjacent cells (called *blocks*) to avoid boundary effects. It is important to denote that Block Decomposition is done with non-overlapping *blocks*[26].

The final feature vector is obtained by concatenation of all the normalized histograms

4.2 Multiple Channel MOSSE Filter

An extended formulation of the MOSSE Filter is presented by Martins *et al.*[17], using multi-dimensional correlation filter which is tailored to take advantage of depth data since it can deal with more than two sources, *e.g.* RGBD.

First and foremost, lets assume that (k) multiple patch experts $\mathbf{h}_i^{(k)}$ exists for the i^{th} landmark and these patch experts can be learned from all image channels. The method consists to simultaneously learn a multi-dimensional correlation filter (MDF) that uses all data from all channels at once. So Equation 3.4, is extended to be minimized across all D channels and can be rewritten as:

$$\min_{\mathbf{h}_i^{(1)},\ldots,\mathbf{h}_i^{(D)}} \sum_{j=1}^{N}\sum_{k=1}^{D} \Big(\mathbf{h}_i \circledast \mathbf{I}_j - \mathbf{g}_j\Big)^2 + \lambda \sum_{k=1}^{D} ||\mathbf{h}_i||^2 \tag{4.5}$$

where N is the number of training images and λ a regularization parameter. The previous equation deduce the MDF $\{\mathbf{h}_i^{(k)}\}_{k=1}^{D}$ that lessen the correlation between current output $(\mathbf{h}_i^{(k)} \circledast \mathbf{I}_j^{(k)})$ and the aimed correction $(\mathbf{g}_j)$ simultaneously across all multi-dimensional samples $(\mathbf{I}_j^k)$ thus yielding the solution:

$$\{\mathbf{h}_{i(l)}^{(k)}\} = \mathcal{F}^{-1}\left\{\left(\sum_{j=1}^{N} \nu\left(\mathcal{F}\{\mathbf{I}_j\}_l\right)^H \nu\left(\mathcal{F}\{\mathbf{I}_j\}_l\right) + \lambda\mathbf{I}\right)^{-1} \sum_{j=1}^{N} \nu\left(\mathcal{F}\{\mathbf{I}_j\}_l\right)^H \nu\Big(\mathcal{F}\{\mathbf{g}_j\}_l\Big)\right\}^{\bullet} \tag{4.6}$$

with $(^H)$ the conjugate transpose and $\nu(\mathbf{A}_l) = \begin{bmatrix} \mathbf{A}_l^{(1)} & \cdots & \mathbf{A}_l^{(D)} \end{bmatrix}^T$ the concatenation operator, in a given frequency l.

The aggregate of the scores of each feature channel originates the overall correlation:

$$\mathcal{P}_i^{MDF} = \sum_{k=1}^{D} \mathbf{h}_i^{(k)} \mathbf{I}(\mathbf{x}_i)^{(k)} \tag{4.7}$$

In conclusion, this extended CLM fitting approach is able to integrate multiple features simultaneously *e.g.* HOG and/or HONV.

Chapter 5

Identity Recognition

This last theoretical chapter, will focus on *sparse coding* since it has recently generated interest in pattern recognition and computer vision. It can be used for both representation and classification, as it minimizes computational overhead and can deal with occlusions. Advantages such as these, make sparse representation (SR) very attractive for identity recognition [30][13][32]. At the end, for a better understanding it will be present a way to unify the CLM with *sparse coding.*

5.1 Sparse Coding

The allegedly *sparse coding* method is a powerful tool to provide good representation of input images, which represents a given set of data by the linear combination of few elements, or atoms, of certain dictionary.

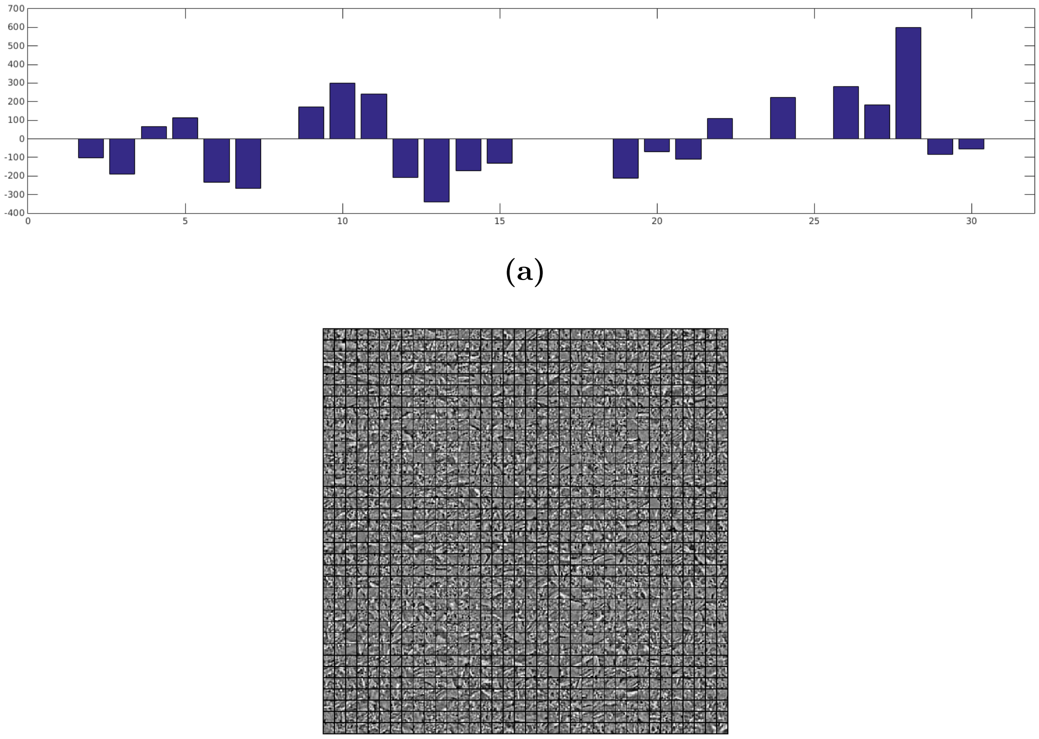
More specifically, using a prior learned over-complete, $n > d$, dictionary $\mathcal{D} \in \Re^{d \times n}$ it is possible to find a sparse linear combination of atoms that most accurately recreate the original image.

Although the existance multiple methods to construct a dictionary [1][30][32][13], they consiss essentially in reproducing the input image with a combination of training samples.

Mathematically, the sparse representation can be formulated as the following minimization problem:

$$\min ||\mathbf{u}||_0 \quad \text{s. t.} \quad ||\mathbf{y} - \mathcal{D}\mathbf{u}||_2 < \varepsilon \tag{5.1}$$

where $\mathbf{u} \in \Re^n$ is the sparse code where each atom weights a dictionary element, 5.1a; $\mathbf{y}$ is test vector; ε is the stopping condition or reconstruction error; l_0 norm $||\mathbf{u}||_0$ the number of positive atoms.



(a)

(b)

Fig. 5.1: Figure 5.1a represents a sparse code $\mathbf{u}$. Figure 5.1b shows an illustratively dictionary.

Ultimately, a pursuit algorithm attempts to find in the dictionary $\mathcal{D} \in \Re^{d \times n}$, a linear combination of elements or atoms, that have maximal projection onto an input vector $\mathbf{y} \in \Re^d$.

5.1.1 Dictionary Learning

Since the final performance of any sparse algorithm rely on a good dictionary, is necessary to have in account size of the bases, d, due to the computational cost to find a sparse code that most accurately represent an input image.

Briefly, considering $\mathbf{Y}$ a set of d-dimensional k input signals, *i.e.* $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_k \end{bmatrix} \in \Re^{d \times k}$ learning a reconstructive dictionary, $\mathcal{D}$, with n items can be performed by solving the following minimization for a sparse representation of $\mathbf{Y}$:

$$< \mathcal{D}, \mathbf{U} > \arg \min_{\mathcal{D},\mathbf{X}} ||\mathbf{Y} - \mathcal{D}\mathbf{U}||_2^2 \quad \text{s. t.} \quad \forall_i, ||\mathbf{x}_i||_0 \leqslant T \tag{5.2}$$

where $\mathcal{D} = \begin{bmatrix}\mathbf{d}_1 & \dots & \mathbf{d}_n\end{bmatrix} \in \Re^{d\times n}$ and considering an over-complete dictionary, $n > d$; the sparse codes of the input signal, $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1 & \dots & \mathbf{u}_k\end{bmatrix} \in \Re^{n\times k}$; the sparsity constrain, T, where each column of $\mathbf{U}$ as T or less non-zero atoms.

Two simple methods for learning discriminative dictionaries were established by Jiang *et al.*[13] untitled *LC-KSVD1*, Equation 5.3, and *LC-KSVD2*, Equation 5.4. Assuming that the stopping condition on both is $||\mathbf{u}_i||_0 \leqslant T$ and adapting Equation 5.2 for reconstructive and discriminative power the objective function can be rewritten as:

$$< \mathcal{D}, \mathbf{A}, \mathbf{U} > = \arg \min_{\mathcal{D},\mathbf{A},\mathbf{U}} ||\mathbf{Y} - \mathcal{D}\mathbf{U}||_2^2 + \alpha||\mathbf{Q} - \mathbf{A}\mathbf{U}||_2^2 \tag{5.3}$$

$$< \mathcal{D}, \mathbf{A}, \mathbf{W}, \mathbf{U} > = \arg \min_{\mathcal{D},\mathbf{A},\mathbf{W},\mathbf{U}} ||\mathbf{Y} - \mathcal{D}\mathbf{U}||_2^2 + \alpha||\mathbf{Q} - \mathbf{A}\mathbf{U}||_2^2 + \beta||\mathbf{H} - \mathbf{W}\mathbf{U}||_2^2 \tag{5.4}$$

The first two terms are common in both algorithms, $||\mathbf{Y} - \mathcal{D}\mathbf{U}||_2^2$ as mentioned previously is the reconstruction error; the second is the discriminative sparse code error being $\mathbf{Q} \in \Re^{n\times k}$ a binary matrix where if a dictionary element $\mathbf{d}_i$ and input signal $\mathbf{y}_j$ share the same label the entry q_{ij} is positive.

The distinction is made by the third term, called classification error, $||\mathbf{H} - \mathbf{W}\mathbf{U}||_2^2$ since there is one column per input signal, and in each column the non-zero index indicates the class of $\mathbf{y}_i$. $\mathbf{H} \in \Re^{c\times k}$. is also a binary matrix with c being the number of classes or labels.

The linear transformation matrices $\mathbf{A} \in \Re^{n\times n}$ coerce the sparse codes to be as discriminant as possible according to $\mathbf{Q}$ and $\mathbf{W} \in \Re^{c\times n}$ is predictive classifier, both must be initialized before being used. The scalars α and β weight the discriminative and classification error's respectively.

According to [13], one initial dictionary is learned for each class using the regular K-SVD [1]. After that, a pursuit algorithm is used to compute a sparse code $\mathbf{U}$ that best suits the training input signals.

The parameters $\mathbf{A}$ and $\mathbf{W}$ can be initialized through the minimizations:

$$\mathbf{A} = \arg \min_{\mathbf{A}} ||\mathbf{Q} - \mathbf{A}\mathbf{U}||^2 + \lambda_1||\mathbf{A}||_2^2$$

$$\mathbf{W} = \arg \min_{\mathbf{W}} ||\mathbf{H} - \mathbf{W}\mathbf{U}||^2 + \lambda_2||\mathbf{W}||_2^2$$

which yield the following solutions using the multivariate ridge regression model [11]:

$$\mathbf{A} = (\mathbf{U}\mathbf{U}^T + \lambda_1 \mathbf{I})^{-1}\mathbf{U}\mathbf{Q}^T \tag{5.5}$$

$$\mathbf{W} = (\mathbf{U}\mathbf{U}^T + \lambda_2 \mathbf{I})^{-1}\mathbf{U}\mathbf{H}^T \tag{5.6}$$

With the initial dictionary, **A** and **W** the optimal dictionary according to Equation 5.3 and Equation 5.4 can be obtained. Using the efficient K-SVD algorithm [1] to find the optimal solution for all parameters Equation 5.4 can be rewritten as:

$$< \mathcal{D}, \mathbf{A}, \mathbf{W}, \mathbf{U} >= \arg \min_{\mathcal{D},\mathbf{A},\mathbf{W},\mathbf{U}} \left\| \begin{pmatrix} \mathbf{Y} \\ \sqrt{\alpha}\mathbf{Q} \\ \sqrt{\beta}\mathbf{H} \end{pmatrix} - \begin{pmatrix} \mathcal{D} \\ \sqrt{\alpha}\mathbf{A} \\ \sqrt{\beta}\mathbf{W} \end{pmatrix} \mathbf{U} \right\|_2^2 \quad \text{s. t.} \quad \forall_i, ||\mathbf{u}_i||_0 \leqslant T \tag{5.7}$$

Assuming that $\mathbf{Y}_{new} = \begin{pmatrix} \mathbf{Y}^T & \sqrt{\alpha}\mathbf{Q}^T & \sqrt{\beta}\mathbf{H}^T \end{pmatrix}^T$ and $\mathcal{D}_{new} = \begin{pmatrix} \mathcal{D}^T & \sqrt{\alpha}\mathbf{A}^T & \sqrt{\beta}\mathbf{W}^T \end{pmatrix}^T$ the previous equation can be rewritten as:

$$< \mathcal{D}_{new}, \mathbf{U} >= \arg \min_{\mathcal{D}_{new},\mathbf{U}} ||\mathbf{Y}_{new} - \mathcal{D}_{new}\mathbf{U}||_2^2 \quad \text{s. t.} \quad \forall_i, ||\mathbf{u}_i||_0 \leqslant T \tag{5.8}$$

Since a L_2 norm is made on $\mathcal{D}_{new}$ before using the K-SVD it is impossible to use $\mathcal{D}$, **A** and **W**. So after the dictionary learning it is mandatory to make the following computations:

$$\hat{\mathcal{D}} = \left[\frac{\mathbf{d}_1}{||\mathbf{d}_1||_2} \quad \cdots \quad \frac{\mathbf{d}_n}{||\mathbf{d}_n||_2} \right] \tag{5.9}$$

$$\hat{\mathbf{A}} = \left[\frac{\mathbf{a}_1}{||\mathbf{a}_1||_2} \quad \cdots \quad \frac{\mathbf{a}_n}{||\mathbf{a}_n||_2} \right] \tag{5.10}$$

$$\hat{\mathbf{W}} = \left[\frac{\mathbf{w}_1}{||\mathbf{w}_1||_2} \quad \cdots \quad \frac{\mathbf{w}_n}{||\mathbf{w}_n||_2} \right] \tag{5.11}$$

The matrix $\hat{\mathbf{A}}$ can be neglected since just $\hat{\mathcal{D}}$ and $\hat{\mathbf{W}}$ are necessary. The optimal dictionary learned $\hat{\mathcal{D}}$ will obtain the sparse codes according to:

$$\mathbf{u}_i = \arg \min_{\mathbf{u}_i} ||\mathbf{y}_i - \hat{\mathcal{D}}\mathbf{u}_i||_2^2 \quad \text{s. t.} \quad \forall_i, ||\mathbf{u}_i||_0 \leqslant T \tag{5.12}$$

Ultimately, using the classifier $\hat{\mathbf{W}}$ to weight the codes and estimate the label j of the image $\mathbf{y}_i$:

$$j = \arg \max_j (\hat{\mathbf{W}}\mathbf{u}_i) \tag{5.13}$$

In conclusion, the methods presented by Jiang *et al.*[13] are similar in multiples ways, the arithmetic can be used reciprocally by them, although no classification error term is used in *LC-KSVD1*, implying that $\hat{\mathbf{W}}$ is despised in Equation 5.13.

5.1.2 Orthogonal Matching Pursuit

Once the dictionary is learned a pursuit algorithm is needed, such as the Orthogonal matching pursuit (OMP), a step wise sparse representation solver, which in each iteration searches for the dictionary elements that have the biggest inner product with respect to the residual vector. First, the algorithm finds and store the index of the element, in the dictionary, that meets:

$$i_k = \arg\ \max_{i} \langle \mathbf{r}_{k-1}, \mathbf{d}_i \rangle \tag{5.14}$$

where $\mathbf{d}_i$ is the i^{th} atom of $\mathcal{D}$ and in the first iteration, $k = 1$, considering the residual vector $\mathbf{r}_0 = \mathbf{y}$ and the zeroed vector $\mathbf{x}$.

The objective in mind is to retrieve the sparse code $\mathbf{u}$ that better represents $\mathbf{y}$, the input vector:

$$\mathbf{u} = \arg\ \min_{\mathbf{u}} ||\mathbf{y} - \sum_{j=1}^{k} \mathbf{d}_{i_j} \mathbf{u}_{i_j}||_2^2 \tag{5.15}$$

being $\mathbf{u}_{i_j}$ the sparse code coefficient that weights $\mathbf{d}_{i_j}$, while the remaining ones are zero. The next step after obtaining $\mathbf{u}$ is required to update the residual vector:

$$\mathbf{r}_k = \mathbf{y} - \sum_{j=1}^{k} \mathbf{d}_{i_j} \mathbf{u}_{i_j} \tag{5.16}$$

Last but not least, the iteration is incremented and the process repeated from Equation 5.14 if the stopping criterion hasn't verified. Important to denote that the stopping condition must allow the method to converge ensuring that $\mathbf{u}$ retains as much sparsity as possible and that the non-zero atoms of $\mathbf{u}$ are equal or less than the number of iteration k.

5.2 Piecewise Affine Warp

As previously shown, the images in the dictionary training and the input signal must always have the same size, due to such Piecewise Affine Warp (PAW) is presented.

Briefly, the Piecewise Affine Warp is a texture mapping procedure where each pixel from the fitting image, $\mathbf{s}$, belonging to a specific triangle, is mapped into the respective destination triangle in the mean shape frame, $\mathbf{s}_0$ using barycentric coordinates (α, β) with bilinear interpolation correction as illustrated in Figure 5.2.

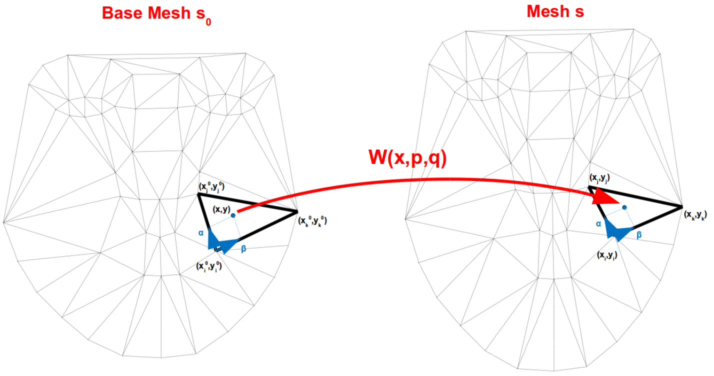


Fig. 5.2: Illustration of the texture mapping procedure for a specific triangle. (Image courtesy of Martins, Pedro.)

The texture mapping is executed by partitioning the convex hull of the mean shape by a set of triangles using Delaunay Triangulation. All pixels enclosed in each triangle are mapped into the correspondent triangle in the mean shape using barycentric coordinates.

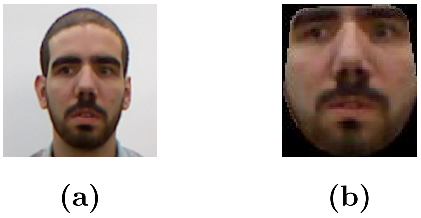

(a) **(b)**

Fig. 5.3: Piecewise affine warping example. Figure 5.3a is the input image while Figure 5.3b is the warped image.

Due to the convex nature of human face shape, the texture mapping should be a direct process, but across several pose variation the triangulation result goes outside the expected face. This can be conquered with restrict Delaunay triangulation. This problem is illustrated in the following figure:

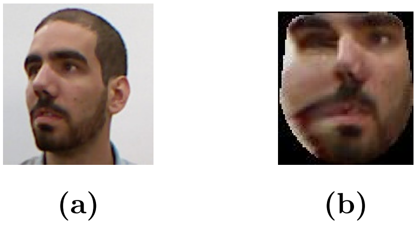

(a) **(b)**

Fig. 5.4: Failed piecewise affine warping. Figure 5.4a is the input image with extreme pose variation while Figure 5.4b is the failed warped image due to the variation.

The warping procedure for a projected pixel $\mathbf{x}$ with shape parameters $\mathbf{p}$ and pose $\mathbf{q}$ is given by:

$$\mathcal{W}(\mathbf{x},\mathbf{p},\mathbf{q}) = \mathbf{x}_i + \alpha(\mathbf{x}_j - \mathbf{x}_i) + \beta(\mathbf{x}_k - \mathbf{x}_i) \quad , \forall \text{ triangles} \in \mathbf{s}_0 \tag{5.17}$$

where the barycentric coordinates α and β yield:

$$\alpha = \frac{(x - x_i^0)(y_k^0 - y_i^0) - (y - y_i^0)(x_k^0 - x_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (x_j^0 - x_i^0)(x_k^0 - x_i^0)} \tag{5.18}$$

$$\beta = \frac{(y - y_i^0)(x_j^0 - x_i^0) - (x - x_i^0)(y_k^0 - y_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (x_j^0 - x_i^0)(x_k^0 - x_i^0)} \tag{5.19}$$

where $\left\langle\ (x_i^0, y_i^0)^T, (x_j^0, y_j^0)^T, (x_k^0, y_k^0)^T \right\rangle$ are the triangle vertexes of the projected base mesh $\mathbf{s}_{0p}$ and $\left\langle\ (x_i, y_i)^T, (x_j, y_j)^T, (x_k, y_k)^T \right\rangle$ are the current mesh $\mathbf{s}$ triangles vertexes coordinates.

As a, result Equation 5.17 can be united into a single per-triangle warp given by:

$$\mathcal{W}(\mathbf{x},\mathbf{p},\mathbf{q}) = (a_1 + a_2 x + a_3 y\ ,\ a_4 + a_5 x + a_6 y)^T \tag{5.20}$$

being a_1, a_2, a_3, a_4, a_5 and a_6 the affine parameters:

$$\begin{aligned}
a_1 &= (x_i(x_j^0 y_k^0 - y_j^0 x_k^0) + x_i^0(x_k y_j^0 - y_k^0 x_j) + y_i^0(x_k^0 x_j - x_j^0 x_k))/\mathit{\Delta} \\
a_2 &= (y_k^0(x_j - x_i) + y_i^0(x_k - x_j) + y_j^0(x_i - x_k))/\mathit{\Delta} \\
a_3 &= (x_k^0(x_i - x_j) + x_j^0(x_k - x_i) + x_i^0(x_j - x_k))/\mathit{\Delta} \\
a_4 &= (y_i(x_j^0 y_k^0 - y_j^0 x_k^0) + x_i^0(y_k y_j^0 - y_k^0 y_j) + y_i^0(x_k^0 y_j - x_j^0 y_k))/\mathit{\Delta} \\
a_5 &= (y_k^0(y_j - y_i) + y_i^0(y_k - y_j) + y_j^0(y_i - y_k))/\mathit{\Delta} \\
a_6 &= (x_k^0(y_i - y_j) + x_j^0(y_k - y_i) + x_i^0(y_j - y_k))/\mathit{\Delta}
\end{aligned} \tag{5.21}$$

with

$$\mathit{\Delta} = (x_j^0 - x_i^0)(y_k^0 - y_i^0) - (y_j^0 - y_i^0)(x_k^0 - x_i^0)$$

It is important to denote that affine parameters only need to be obtained once per triangle. Likewise, a lookup table which encodes the triangle identity is constructed since the projected base mesh is fixed.

With this in mind, it is now possible to jointly use the CLM and the sparse coding for identity recognition since $\mathbf{s}$ is mapped to a fixed frame, originating images with the same size.

Chapter 6

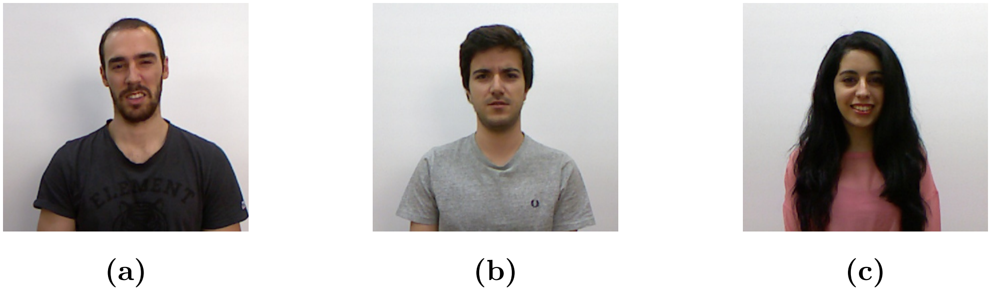
Experiments and Results

Throughout this last chapter a series of tests will be perform to test the viability of face recognition using the combination of CLM with Sparse Coding. Since an extension of the fitting algorithm was address one must confer if there is a improvement in performance in using the HONV descriptor. Regarding the sparse code robustness several tests will be performed with and without the depth image to comprehend is possibility in an real life scenario. Important to mention that a OpenCV[10]/C++ implementation has develop to utilize the Kinect in order to capture depth image and to test the possibility of a real-time implementation.

6.1 Dataset

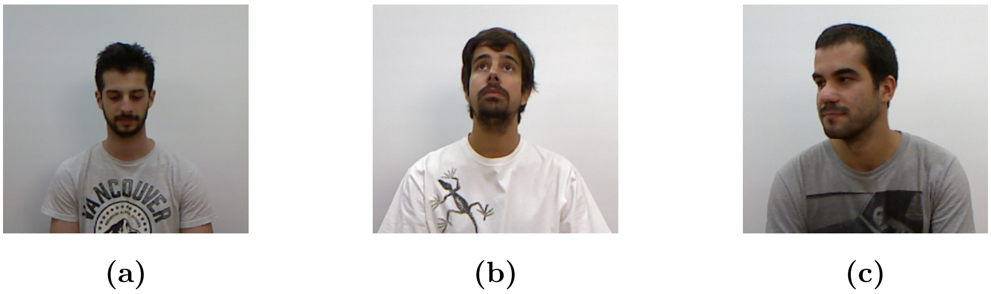
Since both performance evaluations are dependent of a Database with RGB-Depth, but mostly the CLM, one was created since is fundamentally necessary a dataset with 58 points manually label, and currently there is none.

The presented database, entitled *ISRZ*, consists of 242 images of well-aligned 2D, 2.5D and 3D from nineteen individuals taken by the Kinect. It contains at least seven different facial expressions (Neutral, Happiness, Sadness, Anger, Fear, Disgust, Surprised) and five poses for each individual.



(a) (b) (c)

Fig. 6.1: Example of several facial expressions. Figure 6.1a represents Disgust, Figure 6.1b Anger and Figure 6.1c Happiness.

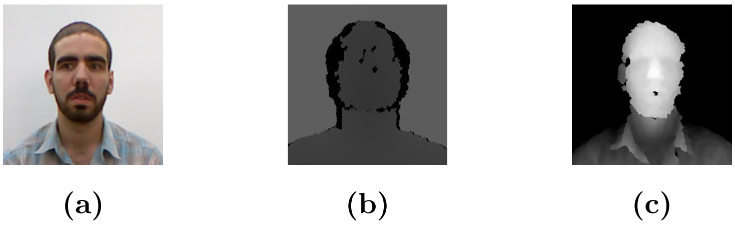


(a) (b) (c)

Fig. 6.2: Example of several poses.

Database Structure

From the nineteen participants two are females and the rest are males. The age varies between 23 to 31 and are from the same country and ethnicity. Four types of data were capture for each shot: 1) the 2D RGB image; 2) the 2.5D depth map; 3) the preprocessed 2.5D depth map; 4) the 3D point cloud;



(a) (b) (c)

Fig. 6.3: Example of three types of image acquired. Figure 6.3a is the RGB frame, Figure 6.3b is the depth image and Figure 6.3c is the preprocessed depth image.

Acquisition

A software implementation was developed for the database recording using OpenCV+OpenNI [23] since it is possible to apply an automatic calibration between the RGB and Depth image, allowing to automatically capture and save all the images, properly organized by type and person. Each volunteer sat at a distance between 0.7 m to 0.9 m with a white board behind at a distance of 1.30 m in order to filter background noise and the Kinect placed at a distance of 1.10 m from the ground.

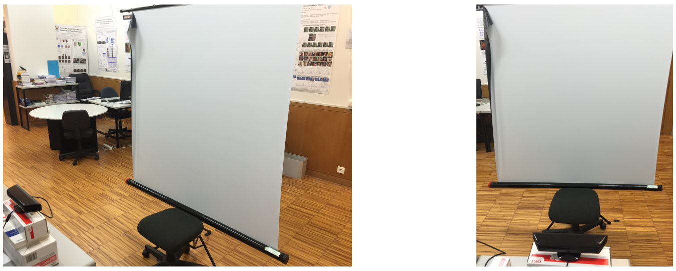

(a) **(b)**

Fig. 6.4: Environment used in the creation of the Database

Facial Landmarks

In order to train a new deformable face model or perform facial recognition, 58 landmarks were manually marked (predominantly around the chin, nose, eyes and eyebrows) in each RGB image since there is a correspondence between the color image and the depth images.

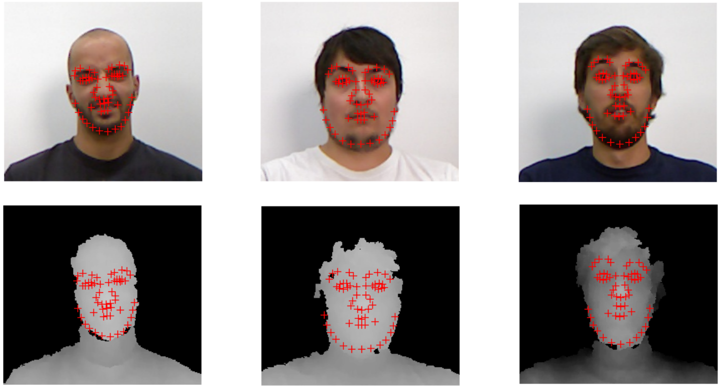


Fig. 6.5: Some marked faces from the created Database with the corresponding depth image

6.2 Fitting Performance

To test the performance of the extended CLM formulation presented in Chapter 4 the usual method [8][29][25][18] is used, where the error metric is given by $e_m(\mathbf{s}) = \frac{1}{v\ d_{eyes}} \sum_i^v ||\mathbf{s}_i - \mathbf{s}_i^{gt}||^2$ being d_{eyes} inter-ocular distance and $\mathbf{s}_i^{gt}$ the location of the i^{th} landmark in the ground-truth. The Figure 6.6 shows the fitting performance for all evaluated methods, which are defined as the ratio between the area below the curve and the total area of the ground truth.

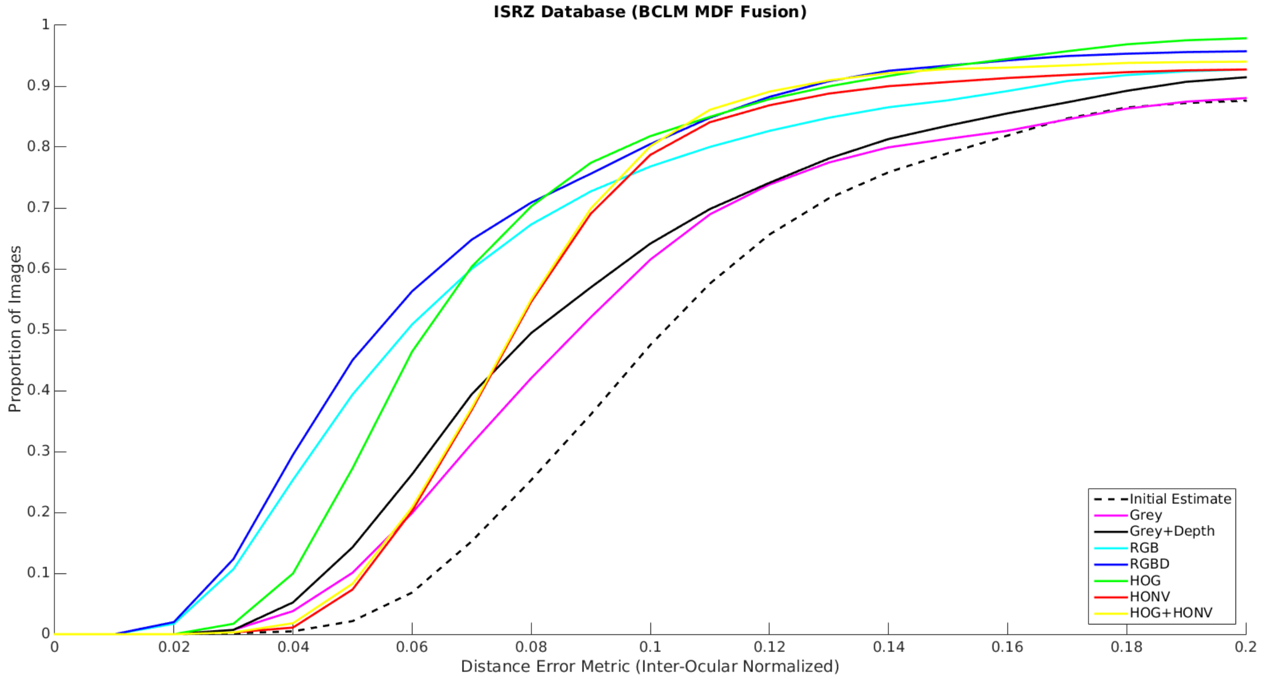
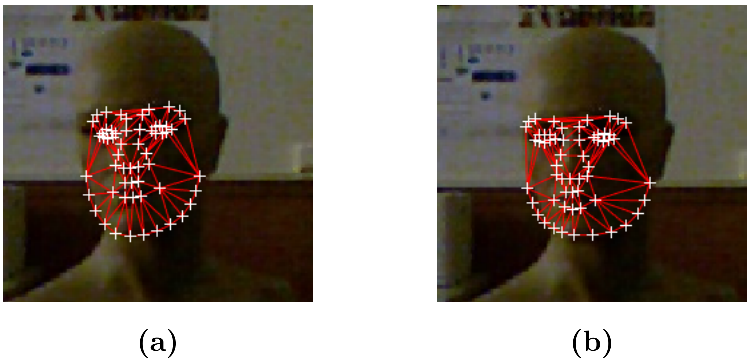


Fig. 6.6: Fitting performance on *ISRZ* database

MDF Fusion	Area Ratio (%)
Gray	49.4
Gray+Depth	52.1
RGB	61.9
RGBD	65.7
HOG	62.8
HONV	56.1
HOG+HONV	57.3

Table 6.1: Quantitative measurement of the ration between area below each curve and total area

As showed in Table 6.1, even thou the results are not the expected it is promising using depth image to improve the face alignment. Some other tests were conducted to verify the performance of the proposed implementation, in specific using images with poor lightning conditions and it has verified that it can mitigate these drawbacks. The Figure 6.7 demonstrate the comparison using HOG versus HOG+HONV.



(a)

(b)

Fig. 6.7: Fitting Process in poor lightning conditions. Figure 6.7a is the fitting with HOG while Figure 6.7b represents the fitting with HOG+HONV

6.3 Face Recognition

First and foremost, recalling Chapter 5 for the Dictionary to be over-complete it needs to have more columns than rows. Since the Database is reduced it is necessary to create synthetic images, so each image suffers some transformations, *e.g.* rotation, translation, motion, increasing/decreasing contrast.

With this transformations the Database was extended from 242 images to 5082 images allowing to project various dictionaries sizes in order to obtain the best accuracy for face recognition.

The accuracy on face recognition can be seen in Table 6.2.

Method	**Accuracy (%)**
Gray	55.6
Depth	22
Gray+Depth	44
RGB	44
RGB+Depth	44

Table 6.2: Recognition results using different methods

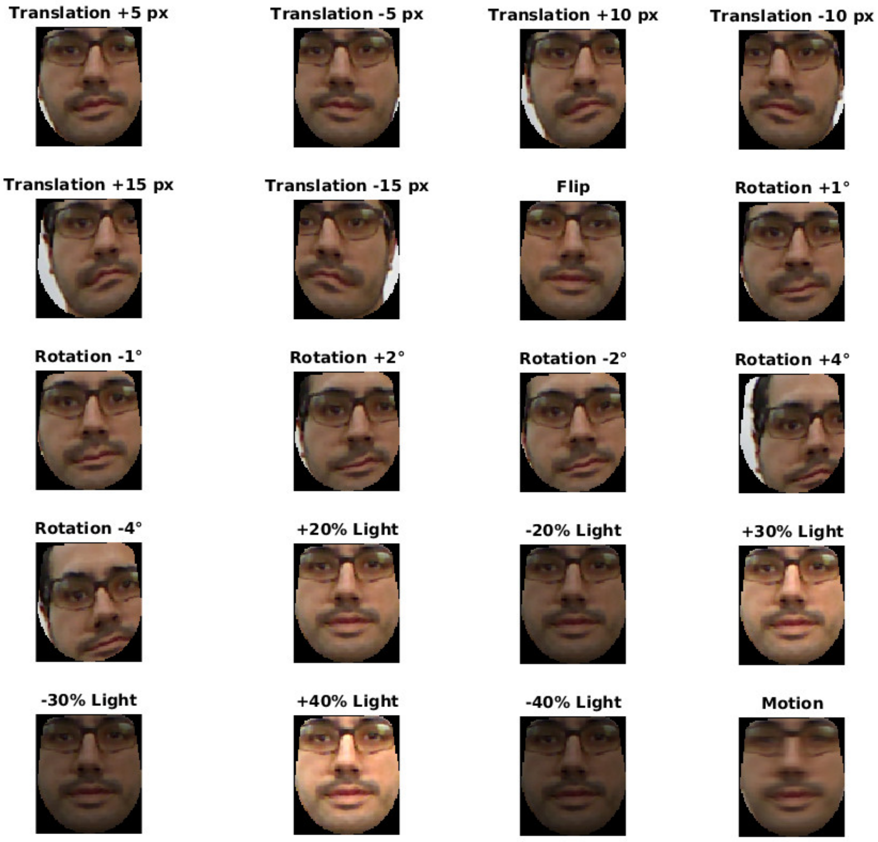


Fig. 6.8: Transformations applied to create synthetic images

6.4 C++ Implementation

This section refers to the C++ implementation and its computational times.

The evaluation was performed with a Intel® Core™ 2 Duo CPU E8400 3.00GHz and 8GB RAM. The following table shows the time consume in each step:

ms	HONV	Gray	Gray+Depth	RGB	RGBD	HOG	HOG+HONV
Face Detect.[28]	354						
Init. Shape + Pose Param.	0.176						
Warp	9	9	10	10	10.5	10.5	10.5
Response Map (each landmark)	2.5	1.4	2.60	3.70	4.90	19.8	21.5
Optimization (each landmark)	0.10						
Kalman Pose	2.55						
Kalman Shape	3.25						

Table 6.3: Computational time of each step of the CLM algorithm (average times in *ms*)

As it can be observed, the real-time implementation it's not possible using only the CPU due to bottlenecks for example, the Response Maps which consume in average $20ms \times 58 = 1.16s$ for all landmarks in the case of the HOG method. Although it is plausible to improve the implementation using and Hybrid implementation (CPU+GPU) since each landmark is independent allowing to compute all landmarks in parallel.

Chapter 7

Conclusions and Future Work

In this thesis the basic tools for facial alignment and facial recognition are covered. This work started with an explanation of one of the most successful methods of deformable models. It moved on to present a solution to capture depth richness and how to include it in a fitting process. Finally it was proposed way to perform facial recognition using sparse coding and way to combine the fitting process to do such a task. It as also presented a Database which was created to meet the needs for both model construction and sparse representation.

In summary, it has possible to achieve promising results relatively to model fitting and face recognition. The tests performed with the available data allowed to better understand the behavior of the algorithms and test its robustness. Error measurements show that proposed extension produces good results compared to the previous achieved results.

As future work, several improvements can be made to improve the accuracy of both face recognition and model fitting such as: improve the Database, since it was created with a small number of people with low diversity, which most likely will improve the face recognition results; real-time implementation, most likely with CUDA this can be achieved [3] due to the fact that each landmark is independent, allowing to exploit the potential parallelism in order to reduce the bottleneck's.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pages 187–194, 1999.

[3] Shiyang Cheng, Akshay Asthana, Stefanos Zafeiriou, Jie Shen, and Maja Pantic. Real-time generic face tracking in the wild with CUDA. *Multimedia Systems*, (1):148–151, 2014.

[4] Colin Goodall. Procrustes Methods in the Statistical Analysis of Shape, 1991.

[5] TF Cootes. Active Appearance Models. *Pattern Analysis and . . .* , 23(6):681–685, 2001.

[6] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models-Their Training and Application, 1995.

[7] David Cristinacce and T. F. Cootes. Feature Detection and Tracking with Constrained Local Models. *Procedings of the British Machine Vision Conference 2006*, pages 95.1–95.10, 2006.

[8] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, volume I, pages 886–893, 2005.

[10] Willow Garage, Itseez, and Intel Corporation. Open Source Computer Vision Library, 2010.

[11] Gene H. Golub, Per Christian Hansen, and Dianne P. O'Leary. Tikhonov Regularization and Total Least Squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[12] Leon Gu and Takeo Kanade. A generative shape regularization model for robust face alignment. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5302 LNCS, pages 413–426, 2008.

[13] Zhuolin Jiang, Zhe Lin, and Larry S. Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, 2013.

[14] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, 1960.

[15] Pedro Martins, Rui Caseiro, and Jorge Batista. Non-parametric bayesian constrained local models. In *CVPR*, 2014.

[16] Pedro Martins, Rui Caseiro, João F. Henriques, and Jorge Batista. Discriminative Bayesian active shape models. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7574 LNCS, pages 57–70, 2012.

[17] Pedro Martins, João F. Henriques, Patrick Brandão, and Jorge Batista. Bayesian constrained local models with depth data. *2016 International Conference on Image Processing, 2016. ICIP '16.*, 2016.

[18] Pedro Martins, João F. Henriques, Rui Caseiro, and Jorge Batista. Bayesian constrained local models revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):704–716, 2016.

[19] Iain Matthews and Simon Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[20] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images, 1996.

[21] Ulrich Paquet. Convexity and bayesian constrained local models. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 1193–1199, 2009.

[22] Duc Son Pham and Svetha Venkatesh. Joint learning and dictionary construction for pattern recognition. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[23] PrimeSense. OpenNI User Guide. *OpenNI User Guide*, page 44, 2011.

[24] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Face alignment through subspace constrained mean-shifts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1034–1041, 2009.

[25] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[26] Shuai Tang, Xiaoyu Wang, Xutao Lv, Tony X. Han, James Keller, Zhihai He, Marjorie Skubic, and Shihong Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7725 LNCS, pages 525–538, 2013.

[27] Thomas Vetter. Efficient, robust and accurate fitting of a 3D morphable model. *Proceedings Ninth IEEE International Conference on Computer Vision*, (Iccv):59–66 vol.1, 2003.

[28] P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1:I—-511—-I—-518, 2001.

[29] Yang Wang, Simon Lucey, and Jeffrey F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[30] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[31] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2D+3D active appearance models. *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 535–542, 2004.

[32] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 1794–1801, 2009.

[33] Qiang Zhang and Baoxin Li. Discriminative K-SVD for dictionary learning in face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698, 2010.