Renato Baptista

Face Recognition in Low Quality Video Images via Sparse Encoding

September 2013

UNIVERSIDADE DE COIMBRA

FCTUC

University of Coimbra
Faculty of Sciences and Technology
Department of Electrical and Computer Engineering of University of Coimbra

Face Recognition in Low Quality Video Images via Sparse Encoding

Renato Manuel Lemos Baptista

Juri:

Prof. Doutor Paulo José Monteiro Peixoto

Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista

Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves

Thesis submitted in partial fulfillment of the requirements for the Master's Degree in the Department of Electrical and Computer Engineering.

September, 2013

This thesis was done under the supervision of Dr. Jorge Manuel Pereira Batista

Department of Electrical and Computer Engineering, University of Coimbra

Acknowledgements

Firstly, I would like to thank to Dr. Jorge Batista for accepting me as his student and for his confidence in my capabilities to do this dissertation and for all the support given. I would also like to thank my laboratory colleagues. A special thanks to my parents, brother and girlfriend for the unconditional support, love, care and motivation along these past months. I would also like to thank to my housemates during this journey, thank you Fábio Nery, João Faro and Tiago Ramalho for all the moments and memories. Last but not least, I would like to thanks to my friends João Santos, Emanuel Marques, João Martins and João Amaro for all the support, thank you guys. Many thanks to my friends who were willing to help me.

Abstract

Currently, the issues concerning security have greater impact in society. During the last decades, face recognition is an area of computer vision that has received a lot of attention. Face recognition is a natural ability of the human being, however, developing algorithms able to recognize faces is a complex process. These algorithms can be used on systems with various applications, more precisely in video surveillance in open areas – areas in which the video surveillance cameras are placed far away from the scene. In these cases, the quality of the image received is not always the best possible, normally compromising the efficiency of a face recognition system. This dissertation addresses the problem of image Super-Resolution (SR) and image quality enhancement, normally named *face hallucination*, providing superior data to the process of video face recognition on open-spaces. Regarding face recognition, this dissertation focus on the concept of video based "dictionary learning" methods using sparse coding. In an attempt to validate this concept a Region Covariance Matrices (RCM's) based approach is explored. These matrices are known to describe an image with more information than just by analysing the information provided by the gray level. The information contained on a RCM goes through simple operations on the image, like image derivatives, but also through Gabor filters. The Gabor filters have been very useful on the field of facial recognition presenting a high success rate.

Keywords: Face Hallucination; Face Recognition; Sparse Coding; RCM; Gabor

Resumo

Nos dias de hoje as questões relacionadas com a segurança têm cada vez mais impacto na sociedade. Durante as últimas décadas o reconhecimento facial é uma área da visão por computador que tem vindo a merecer grandes atenções. O reconhecimento facial é uma habilidade natural do ser humano, contudo, desenvolver algoritmos capazes de reconhecer faces é um processo complexo. Estes algoritmos podem ser utilizados em sistemas com vários tipos de aplicações, mais especificamente em aplicações de vídeo vigilância em espaços abertos, isto é, em espaços em que a câmara de vigilância está situada longe da zona de acção, como por exemplo a monitorização de locais públicos ou privados. Geralmente, nestes casos, a qualidade da imagem adquirida nem sempre é a melhor, podendo comprometer a eficácia de um sistema de reconhecimento facial. Nesta dissertação é estudado um método para melhorar a qualidade e a resolução de uma imagem de modo a que nos sistemas de reconhecimento, as características extraídas da imagem contenham mais detalhe quando comparadas com as imagens adquiridas directamente da câmara de vídeo vigilância. O método é conhecido como *face hallucination*. No que diz respeito ao sistema de reconhecimento facial, nesta dissertação é abordado o conceito de "dicionários aprendidos" em sequências de vídeo utilizando codificação esparsa. De modo a validar este conceito, é explorado uma abordagem baseada em matrizes de regiões de covariância (RCM). Estas matrizes são utilizadas como descritores de uma imagem, contendo muito mais informação do que apenas utilizando a informação proveniente dos níveis de cinzento (*gray level*) de uma imagem. A informação utilizada numa matriz de regiões de covariância (RCM) passa por operações simples sobre a imagem, como derivadas, mas também pela utilização de filtros de *Gabor*. Os filtros de *Gabor* têm vindo a ser bastantes usados na área de reconhecimento facial apresentando uma elevada taxa de sucesso.

Palavras-Chave: Face Hallucination; Reconhecimento Facial; Codificação Esparsa; RCM; Gabor.

Contents

List of Figures

List of Tables

List of Acronyms

PCA Principal Component Analysis

PPCA Probabilistic PCA

SANP Sparse Approximate Nearest Point

SR Super-Resolution

ANN Approximate Nearest Neighbors

RCM Region Covariance Matrix

ROC Receiver Operating Characteristic

FAR False Acceptance Rate

TAR True Acceptance Rate

GRCM Gabor-based Region Covariance Matrix

Chapter 1

Introduction

Biometric identification is a technique of automatically identifying or verifying an individual by physical characteristic. Biometric measures are divided in two categories: behavioral and physical. Behavioral biometrics is related with the person's behavior like typing rhythm, gait, and voice. Physical biometrics uses the eye, iris recognition, fingerprints, hand geometry, palm print, face recognition, DNA, and others. Nowadays, questions related to security are gaining more and more importance in the society. Face recognition brings several advantages over other biometric methods. It can be done passively without any explicit action by the user, such as fingerprint or hand geometry detection (the user needs to place his hand on a hand-rest). This is very beneficial for surveillance purposes because the face images can be acquired from a distance by a camera. Face recognition has always been a cause for concern in the scientific communities because of its non invasive nature and it is people's primary characteristic of person identification. Traditional algorithms of face recognition recognize face from static images but it can bring several issues like illumination, pose and expression variation over time. To avoid this problems related to variations over time, it is used video sequence in order to take advantage of the motion and temporal information. The advantages over static-images algorithm are: the huge affluence of data allows the system to choose the frame with the best possible image and ignores the worst frames. Video-based sequence provides temporal continuity, so classification information from several frames can be combined to increase the success rate of a recognition system. Furthermore, video allows the tracking of face images such that variations in facial expressions and pose can be compensated for. Face recognition methods are constantly being developed in order to increase effectiveness of security systems. Those techniques can be used to validation of control access for PCs, for private areas in buildings, for ATM transactions and many other,

but it can also be implemented in open areas like places where the camera is situated far from the scene such as car parking, hallways of buildings, public and private surveillance, criminal identification and many others. Generally, in video surveillance systems are used low resolution cameras. Thus, the resolution of face image is low, but the details of facial features which can be found in a potential high resolution face image may be crucial for recognition. This potential high resolution face image can be obtained by face hallucination. Face hallucination is super-resolution of a face image, in other words, a method to clarify the details of face from a low resolution image. The main focus of this dissertation is face recognition from video surveillance with low resolution face images based on sparse coding.

1.1 Related Work

Face recognition has been a research problem in crescent development during the past several years. Traditional face recognition methods use faces from static images [2, 3, 4]. These methods are classified by face recognition from intensity images, which are divided in two main categories: feature-based and holistic [5, 6, 7]. Feature-based techniques consist to identify and extract distinctive facial features like the eyes, mouth, nose, etc., and then compute the geometric relationships between those facial features. Thus, the input image is reduced to a vector of geometric features. One of the earliest work was developed by Kanade [5] and a simple process to extract a vector of 16 facial parameters is created by using: ratios of distances, areas and angles. The measurement used to achieve the recognition is the Euclidean distance. Other methods were later developed by following Kanade's approach. These feature-based methods are relatively robust to position variations in the input image [8], but a drawback of these methods are the difficulty of automatic feature detection. The other category of face recognition from intensity images is related to the holistic techniques. Theses techniques seek to identify faces using global representations (entire image). The image is represented as a 2D array of intensity values and the recognition is performed by comparing the input face image and a database with faces. This approach is computationally very expensive and it is limited due to sensitivity of face orientation, size, illumination variation, background clutter, and noise [9]. To economically represent face images, Sirovich and Kirby [10] were the first to use Principal Component Analysis (PCA) [11, 12]. Given a face image, it can be efficiently represented along the eigenpictures coordinate space. This face can be approximately reconstructed by using a small set of eigenpictures and the corresponding projections along each eigenpicture. PCA based methods appears to be robust

to illumination variations but its performance degrades with scale changes. When a single image of each person is available PCA appears to work well, but when multiples image by person are available PCA retains undesired variations due to lighting and facial expressions [13]. Moses *et al.* [14] said that these variations are almost always larger than image variations due to a change in face identity. Belhumeur *et al.* [13] proposed a Fisher's Linear Discriminant Analysis (Fisherfaces) [15], which maximizes the ratio of the between-class scatter and the within-class scatter. Many others algorithms were developed by following PCA and Fisherfaces approaches, as related in [16].

A video-based face recognition system consists of three steps: face detection, face tracking and face recognition. Video-based face recognition appears to be at a disadvantage relative to static-image recognition due to low quality images, cluttered background, the presence of more than one face, and a huge amount of data. However, the enormous abundance of data and the temporal continuity provided by a video are major advantages of using video-based face recognition. Furthermore, video allows the tracking of face images with variations in facial expressions and poses, resulting in improved recognition [17]. Given a video sequence, the first step is to detect a face and then it is used a tracking method to store all face images together. Zhou *et al.* [18] proposed an algorithm to exploit the temporal information in manner that tracking and recognition of faces become sequential tasks. Zhou's tracking-then-recognition method resolves uncertainties in tracking and recognition simultaneously in a unified probabilistic framework. Another track-then-recognition method is proposed by Lee *et al.* [19] where an individual is represented by a complex nonlinear appearance manifold. The complex nonlinear appearance manifold of each registered person is divided into a set of submanifolds. Each submanifold consists of nearby poses ant it is obtained by PCA of frames from video sequences.

In a video sequence, aspects such as variations in illumination and pose are very important in face recognition validation. Arandjelović and Cipolla [20] proposed a face recognition method based in video sequences where illumination and pose present variations during the sequence. The proposed method consists of using a weak photometric model of image formation with offline machine learning. It is shown that the combined effects of face pose and illumination can be effectively learned using Probabilistic PCA (PPCA) from a small, unlabelled set of video sequences of faces in randomly varying lighting conditions. Given a new sequence, the learned model is used to decompose the face appearance manifold into albedo and pose-illumination manifolds, producing the classification decision by robust likelihood estimation.

Face recognition based in video sequence also has statistical methods. Turaga *et al.* [21] present methods that use subspace-based models and tools from Riemanian geometry of the Grassmann manifold. Techniques like intrinsic and extrinsic statistics are used to enable maximum-likelihood classification. Hu *et al.* [22] proposed an algorithm for images set classification and introduced a novel between-set distance called Sparse Approximate Nearest Point (SANP) distance. The dissimilarity is measured as the distance between SANP of two image sets and it uses a scalable accelerated proximal gradient method for optimization.

Chen *et al.* [23] introduced the concept of video-dictionaries for face recognition. A generative approach based on dictionary learning methods is proposed to minimize the challenges of face recognition from unconstrained videos. The principal advantage is the robustness to some variations such as illumination and pose due to video sequence partition algorithm and to sequence-level dictionaries learning method.

Increase quality in an image, also known as Super-Resolution (SR), may be important in many scenarios where objects of interest are not clear to users perspective due to far distance or blurriness. One particular interest of SR techniques is to compute high-resolution images from low-resolution ones. This technique was introduced by Baker and Kanade [24, 25] as face hallucination. This technique consists of learn a prior on the spatial distribution of the image gradient for frontal face images. This technology has many applications in areas like image enhancement, image compression and object recognition.

Liu *et al.* [26] approach consists to combine a global parametric model with a local nonparametric model. The global model assumes a Gaussian distribution learned by PCA. The local model uses a patch-based nonparametric Markov network to learn the statistical relationship between the global image and local features. Face alignment is an important issue for successful face hallucination, so a robust low-resolution face alignment algorithm is designed to increase the success rate of face hallucination technique.

Yang *et al.* [27] presented a new approach to conduct single-image SR base on sparse representation signal. This approach seeks a sparse representation for each patch of the low resolution input, and then use the coefficients of this representation to generate the high-resolution output.

In [28], Jia proposed a novel face hallucination algorithm that uses information from previous low-resolution face images extracted from the same video sequence to produce high-resolution face image. This method does not use a statistical relationship between global and local features, so it has a computational time lower than Liu's solution. This

method has not such good results as Liu's method, but it provides a reasonable increase of face image quality.

1.2 Thesis Description

Face recognition systems are used to verify the identity of an individual by matching a face against a database of known faces. For applications like video surveillance security is necessary to avoid the temporal redundancy while capturing variations due to changes in pose and illumination. Thus, it is followed an approach introduced by Chen *et al.* [23] based on video-dictionaries. These applications can be used to validation of control access like access control for PCs, ATM transaction and many others, but it can also be used in open areas such as public and private surveillance, criminal identification and so on. In these last cases, generally, the acquired image is of low quality due to far distance between the camera and the scene, so it is important to enhance image quality in order to achieve better results on face recognition systems.

This dissertation is organized as follows: Chapter 1 introduces the topic and the related work. Chapter 2 describes a method to increase image resolution from previous low resolution input face images and how to build an image using sparse representation. Chapter 3 relates a face recognition solution based on video sequences to increase the robustness to some variations like pose or illumination. An input video sequence is divided into different partitions to remove the temporal redundancy due to changes in pose and illumination. Each partition is used to learn a partition specific dictionary via sparse coding. The partition-specific dictionaries (sub-dictionaries) are combined in order to create a sequence-specific dictionary which are used to perform identification and verification processes. Chapter 4 shows the experimental results and all the comparisons obtained from all process developed in past chapters. And at last, chapter 5 concludes this work.

Chapter 2

Face Hallucination

In video surveillance systems sometimes it is important to recognize a face from video stream. Due to far distance between the cameras and the scene it is hard to achieve good recognition rates. So, it is important to develop a method that can be able to increase face image resolution, called face hallucination. Face hallucination is Super-Resolution (SR) of face images, which is the process of combining multiple low resolution images to form a higher resolution image.

Face hallucination was introduced by Baker and Kanade [24] and later Liu *et al.* [26] proposed a new approach to hallucinate low resolution face images. A successful face hallucination algorithm should meet the following three constraints:

1. **Sanity constraint.** The result face image must be very close to the input face image when smoothed and down-sampled.

2. **Global constraint.** The result face image must have some common features of a human face (eyes, mouth, nose, etc.).

3. **Local constraint.** The result face image must have specific characteristics of this face image with photorealistic local features.

Sanity constraint can easily be met, it can be simply formulated as a linear constraint on the result. A global parametric model and a local nonparametric model are more difficult to formulate. The global constraint assumes a Gaussian distribution learned by PCA and the local constraint uses a patch-based nonparametric Markov network to learn the statistical relationship between the global face image and the local features. This approach follows Freeman *et al.* [1] to build a nonparametric patch-based Markov network. A dictionary with several high resolution images is created by dividing all images into a huge number

of high and low resolution patches. These image patches could be from many types of images such as nature landscapes, animals, people, streets, etc. These patches are used to generate a Markov network to probabilistically model the relationships between high and low resolution patches, and between neighboring high resolution patches. This relationship is shown in figure 2.1 where the circles represent network nodes and the lines indicate statistical dependencies between nodes. The probability of any given high resolution choice for each

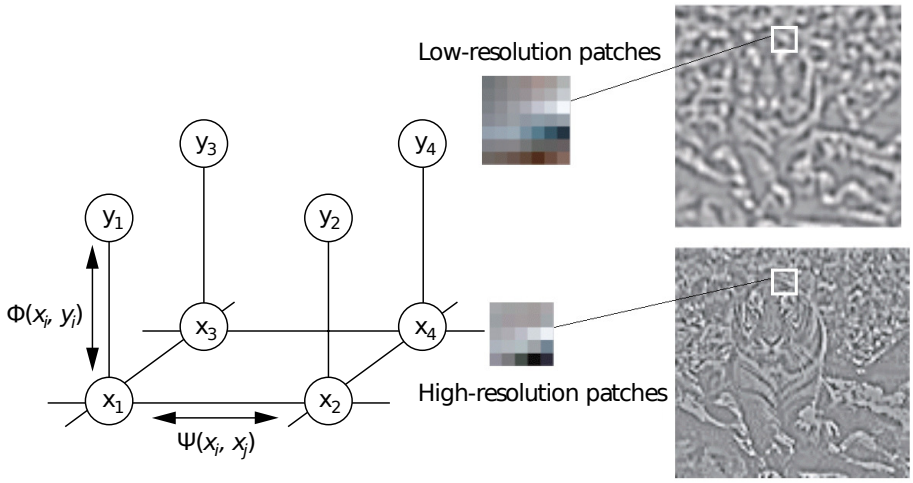


Figure 2.1: Markov network model for the SR problem. Image extracted from [1]

node is proportional to the product of all sets of compatibility matrices ψ relating the possible states of each pair of neighboring hidden nodes, and vectors ϕ relating each observation to the underlying hidden states:

$$P(x|y) = \frac{1}{Z} \prod_{(ij)} \psi_{ij}(x_i, x_j) \prod_i \phi(x_i, y_i), \tag{2.1}$$

where Z is a normalization constant. The first product is over all neighboring pairs of nodes, i and j. The observed low and the estimated high resolution patches at node i are y_i and x_i, respectively.

A dictionary of high resolution patches (figure 2.2) is created by using several high resolution images. It is used to search the relationship between low and high resolution patches.

Patch-by-patch searching is very time consuming, so it is undesirable in video surveillance applications. Jia *et al.* [28] describe a method to avoid this time consuming step. Jia uses an online dictionary approach instead of offline trained dictionaries. Online dictionary consists of combining similar face features from several low resolutions tracked faces to enhance the target face. This algorithm does not require a prior training database, since tracked faces

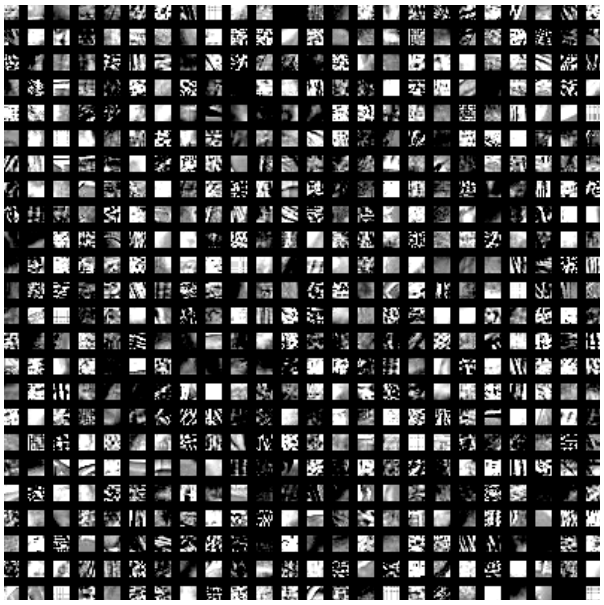


Figure 2.2: Example of a dictionary with high resolution patches

are used to generate an eigenfaces database.

Based on image statistics, an image patch can be well-represented as a sparse linear combination of elements from an appropriately chosen over-complete dictionary. Inspired on this, Jia considers eigenfaces as the global image patches. The eigenfaces are over-complete because one query face can only match a small number of eigenfaces with similar shape. Since during PCA training eigenfaces are generated based on differences between every training face and the mean faces, sparse representation is used for the difference face instead of the original face.

Freeman *et al.* [1] used a tree-based Approximate Nearest Neighbors (ANN) search. ANN[1] searching is a well-known database indexing and searching method which quickly and accurately retrieves nearest neighbors from a database. To avoid high computational times, Jia proposed a new manner to employ ANN search. Patches with high and low resolution are stored into one stack (figure 2.10) and it is used as the input query to search against the trained database. In this way, it is only needed to do one-time searching and get the k nearest neighbors for all the query patches. This approach can greatly reduce the time required to perform a search.

[1] `http://en.wikipedia.org/wiki/Nearest_neighbor_search`

2.1 Algorithm Description

The proposed algorithm by Jia *et al.* [28] can be divided into three different parts. The first part is learning an online database using PCA training on tracked face images. The second part is the step of sparse representation which is used to reconstruct a difference face based on eigenfaces database. The last part is the searching step, which combines information from the tracked low resolution face images to form higher resolution face image. The proposed algorithm is divided in seven steps and are described below:

1. At first, face must be detected and tracked from a video sequence. All tracked faces must be grouped together to apply face hallucination, like figure 2.3.

Figure 2.3: Example of an input low resolution sequence

2. Given a grouped face image sequence (figure 2.3), the first step is to generate a mean face μ and create an eigenface database B by PCA training, following Turk *et al.* [29].

3. The size of face image extracted from video sequence is usually small. To increase an image resolution is necessary to enlarge image size. A low resolution image I_L is a blurred and downslampled version of the high resolution image I_H (equation (2.2)). Based on [30] C is the blurring matrix and H is the decimation matrix. Bicubic interpolation (upgrade factor U) is used to approximately represent the inverse decimation matrix H^T in order to generate the interpolated high resolution image $\bar{I_H}$ (equation (2.3)).

$$I_L = C \cdot H \cdot I_H. \tag{2.2}$$

$$\bar{I_H} \approx C^T \cdot H^T \cdot I_L. \tag{2.3}$$

   Later the interpolated mean face $\bar{\mu}$ is subtracted from the approximated high resolution image $\bar{I_H}$ to generate the difference face $Diff_{face}$ (equation (2.4)). This difference face shows the lost global face shape for the target face.

$$Diff_{face} = \bar{I_H} - \bar{\mu}. \tag{2.4}$$

4. The result of (2.4) is used to estimate the sparse coefficients α (figure 2.4), in order to

combine eigenfaces from the database to obtain the difference face image reconstruction. Thus, a new difference face is created (2.5).

$$New_{Diff_{face}} = \alpha \cdot B. \tag{2.5}$$

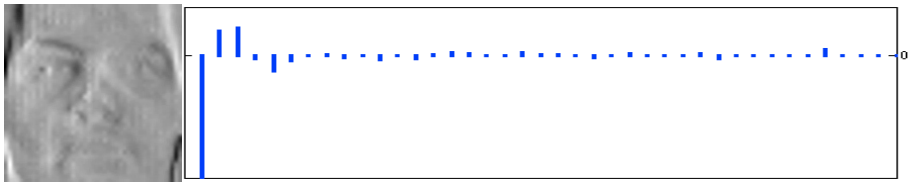


Figure 2.4: Sparse coefficients representation of reconstructed face

5. To increase detail in interpolated high resolution image, $\bar{I_H}$, it is added the new difference face to the interpolated image. Basically the global enhanced image ($I_{H,GlobalEnhanced}$) is the interpolated target image with better details at high-frequency components.

$$I_{H,GlobalEnhanced} = \bar{I_H} + \alpha \cdot B = \bar{I_H} + New_{Diff_{face}}. \tag{2.6}$$

6. The global enhanced image (2.6) is a bit noisy and has some artifacts. In order to reduce that unwanted components it is applied a bilateral filter. Bilateral filter, implemented by Tomasi and Manduchi [31], is a noniterative, local and simple method that smooths image while preserving edges by using a nonlinear combination of nearby image values. Figure 2.5 shows an example of bilateral filter application.

Figure 2.5: Bilateral Filter example. Left image is before and right is after filtering

7. After the bilateral filtering, the globally enhanced face $I_{H,GlobalEnhanced}$ is further enhanced by a method based on ANN search. The returned image is the final hallucinated face. This ANN search method is different from patch-by-patch searching. All patches

are stored into one stack (entire image is considered as the stored patches/stack). This stack is used as a query to search from the training database.

To better understanding, figure 2.6 describes visually all of those steps explained before.

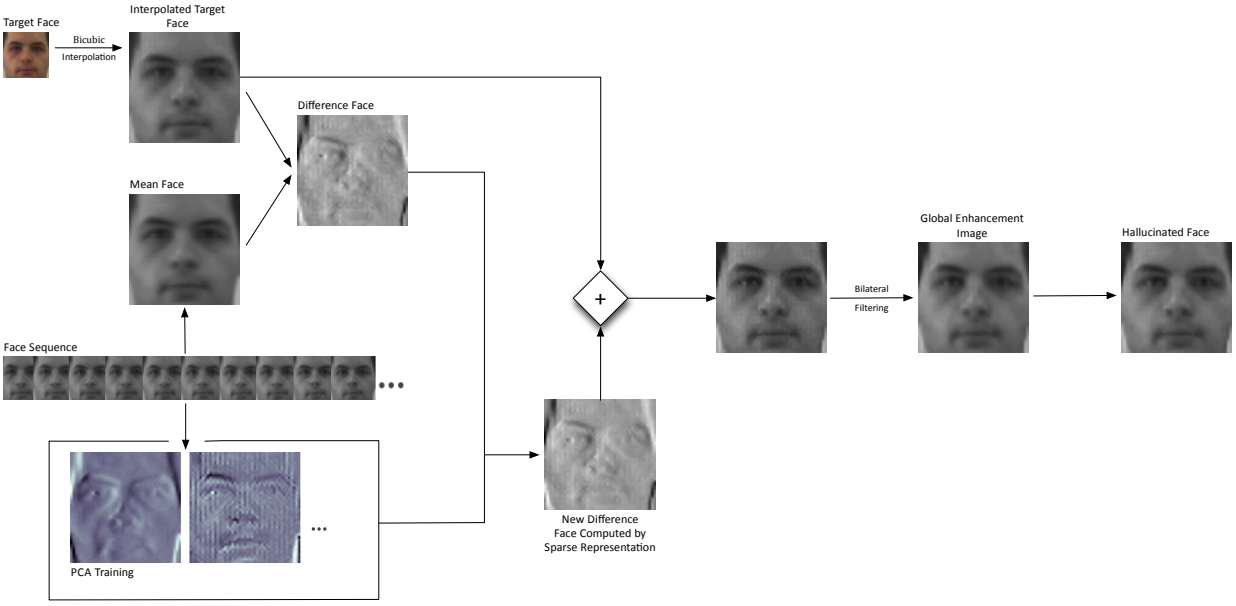


Figure 2.6: Algorithm Overview

2.2 Sparse Representation

Given a video sequence of cropped face images, the first step is to enlarge all images using a bicubic interpolation. After that, PCA training is applied to generate a mean face $\bar{\mu}$ and an eigenface database B. This database is used to estimate the sparse coefficients α (example in figure 2.4). Research on image statistics suggests that image patches can be well-represented as a sparse linear combination of elements from an appropriately chosen over-complete dictionary. Based on sparse signal representation, Yang *et al.* [27] presented a method to seek a sparse representation for each patch of the low resolution input, and then use the coefficients of this representation to generate the high resolution output. Instead of this patch based sparse representation, Jia's approach seek a sparse representation for a full image, using the difference image (equation (2.4)) as a query to compute the sparse representation coefficients (α) from the PCA database (B). To estimate this coefficients it is necessary to solve a ℓ_1-regularized least square problem [32] as follows:

$$\min \lambda \left\| \alpha \right\|_1 + \frac{1}{2} \left\| B \cdot \alpha - Diff_{face} \right\|^2, \tag{2.7}$$

where the parameter λ is a constant and balances sparsity of the solution. This is a linear regression regularized with ℓ_1-norm on the coefficients, also known as LASSO in statistical literature [33]. Sparse coefficients α is a vector with the same size as the number of images used to create an eigenface database. To build an image using sparse representation (equation (2.5)) it is necessary to apply a linear combination of vector α and the eigenface database B. Concretely, each new difference image $\vec{New_{Diff_{Image}}} \in \mathbf{R}^k$ is succinctly represented using basis vectors $\vec{B_1}, \vec{B_2}, \cdots, \vec{B_n} \in \mathbf{R}^k$ and a sparse vector of coefficients $\vec{\alpha} \in \mathbf{R}^n$ expressed by 2.8

$$\vec{New_{Diff_{Image}}} \approx \sum_j \vec{B_j} \alpha_j. \tag{2.8}$$

To explain this combination, figure 2.7 shows how it works.

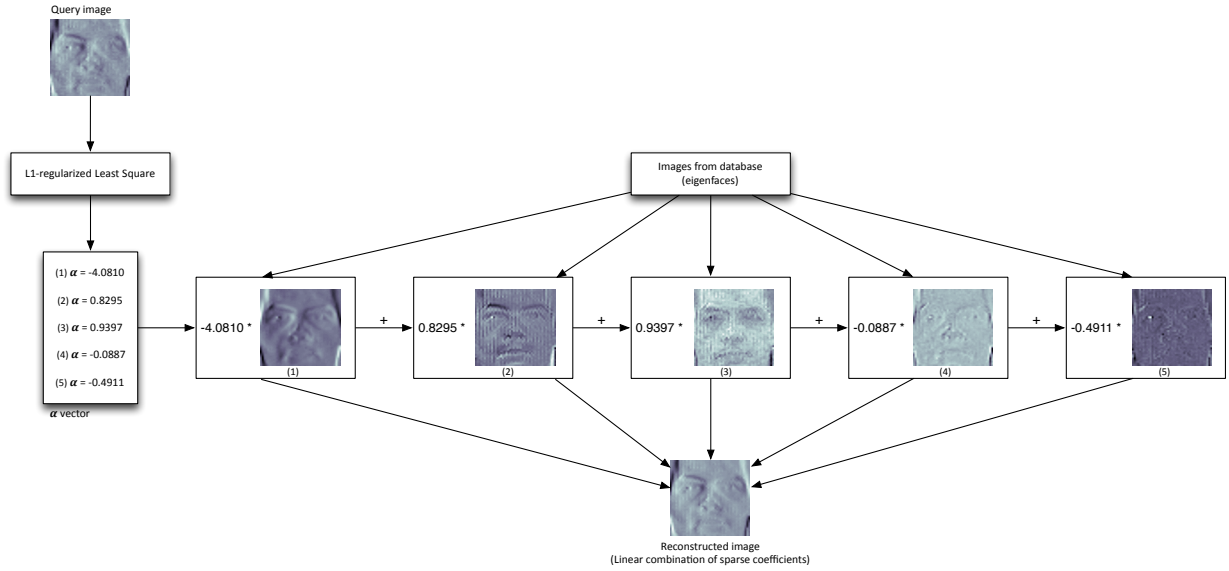


Figure 2.7: Example of linear combination of sparse coefficients α

2.3 Approximate Nearest Neighbor Search

The patch-by-patch searching is very time-consuming. Each patch from the interpolated image is searched from the whole training database. Thus, if the training data is large and the size of interpolated image is large, this iterative searching process will take a very long time. To avoid this time-consuming process, Jia proposed a new approach to employ Approximate Nearest Neighbors (ANN) search. ANN searching is a well-known database indexing and searching method, which quickly and accurately retrieves nearest neighbor from a database. This searching step is very important to face hallucination process. Many algorithms, [24, 26, 1], use dictionaries with high resolution image patches. These dictionaries

are trained offline and it uses all kind of high resolution images like nature landscape, streets, flowers, animals or even people. These images are divided into many thousands of high and low frequency patches in order to learn a dictionary. Each low frequency patch must have a correspondence between its high frequency patch and itself. Figure 2.8 shows the correspondence between the low and high frequency patches.

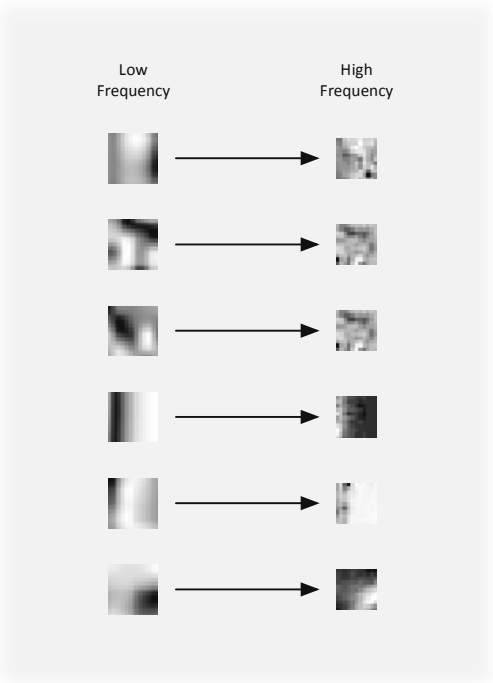


Figure 2.8: Correspondence between low and high frequency patches

Patch-by-patch (or local search) searching from a dictionary with a lot of patches is very time consuming. Moreover, local search alone is not sufficient to estimate plausible looking high-resolution detail. Searching a patch from the low resolution patches dictionary results in a many similar low resolution patches. Their corresponding high resolution patches looks fairly different from each other. Figure 2.9 shows why local search does not work.

Local patch information alone is not sufficient for SR. Therefore, an approach to exploit neighborhood relationships is explored. A Markov network is used to probabilistically model the relationships between high and low frequency patches, and between neighboring high frequency patches, as figure 2.1 shows.

Jia developed a new algorithm to employ ANN to accurately retrieve high frequency information. It consists in creating an online dictionary with low and high frequency components from the image sequence (figure 2.3). Instead of using image patches, a query stack containing the entire image is created. This stack is used to perform a search against the trained dictionary. Figure 2.10 shows the resulting stack used as an input to perform a

Figure 2.9: Input patch and corresponding low and high resolution patches.Image extracted from [1]

search. The green part is the low frequency information and the red part is the high frequency information. The left side relates to the search method using dictionaries with image patches, and the right side shows the Jia's method.

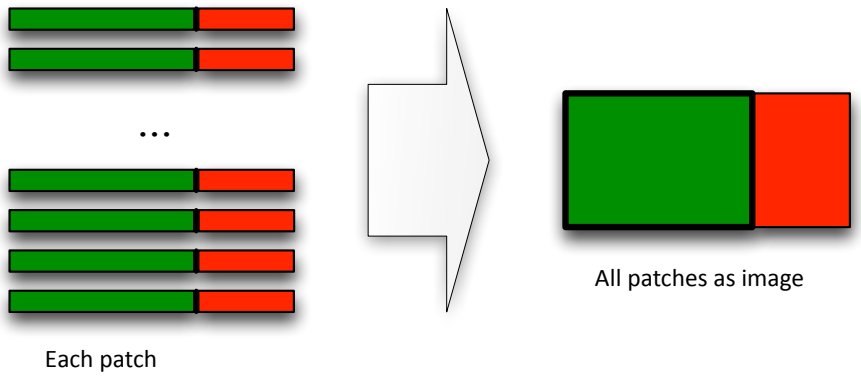


Figure 2.10: Each patch to a stack

The low and high frequency information used to build this new stack are the low and high frequency of the image. Figure 2.11 shows an example of the corresponding low and high frequency images used to build the stack.

There are two major advantages of using the method proposed by Jia, which are: The method does not use offline trained dictionaries with image patches which can be from all kind of images. Another advantage is that the dictionary based on Jia's approach has a considerably less elements than others approaches which increases the speed to perform a search. However, there is one important trade-off to keep in mind. As the search is done using the full image patches (stack), the returned nearest neighbors for the whole stack are not necessarily accurate.

The new stack is build with the low frequency information of the input interpolated face

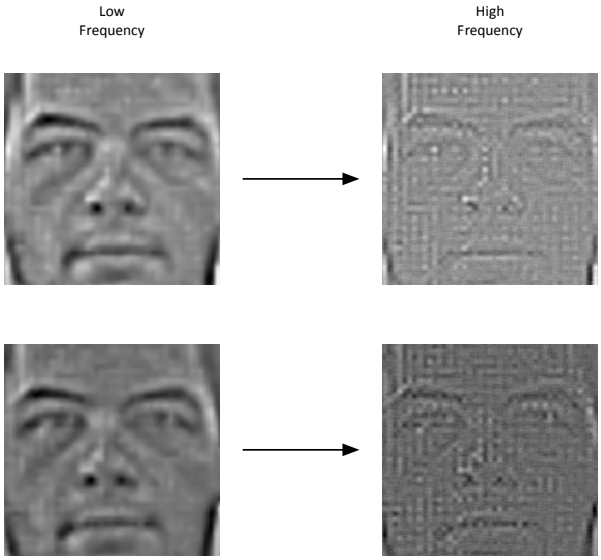


Figure 2.11: Correspondence between low and high frequency images

which is used to search from a database of low frequency images. According to ANN's theory, the time to find one exact nearest neighbor of a query is similar to the time for finding a number k of nearest neighbors of a query. After searching for the k nearest neighbors (k closest low frequency images of an input) takes place a matching process in order to get the correspondence between the low and high frequency k nearest neighbors. The resulting k high frequency images are subjected to a median filter, in order to improve the accuracy and filter out noise and artifacts. Figure 2.12 shows this process of combining all images using a median filter.

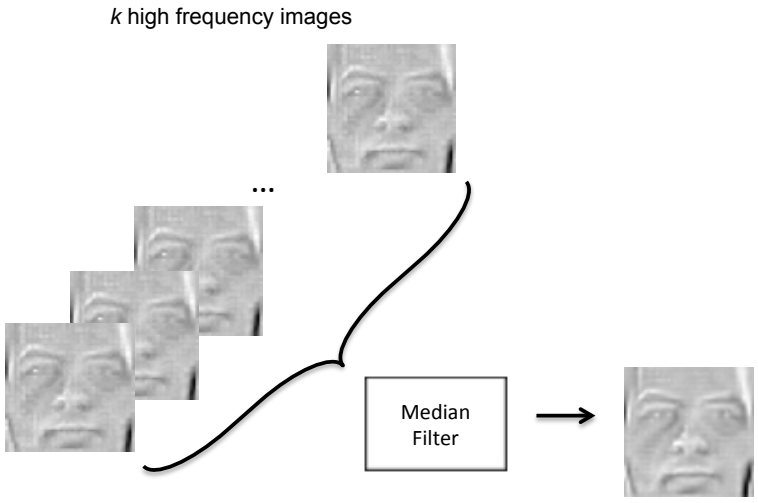


Figure 2.12: Median filter process

The image resulting of the median filter is added to the globally enhanced face ($I_{h,GlobalEnhanced}$) in order to improve the image detail. The method presented by Jia uses the information retrieved from a stack of tracked faces to enhance image quality. This process is done online.

On the other hand, methods like [1, 26] uses an offline training process. In these methods the image enhancement is done using a single frame image to estimate missing high resolution detail that is not present in the original image.

Figures 2.13 and 2.14 shows the overall process of ANN based on Jia's approach.

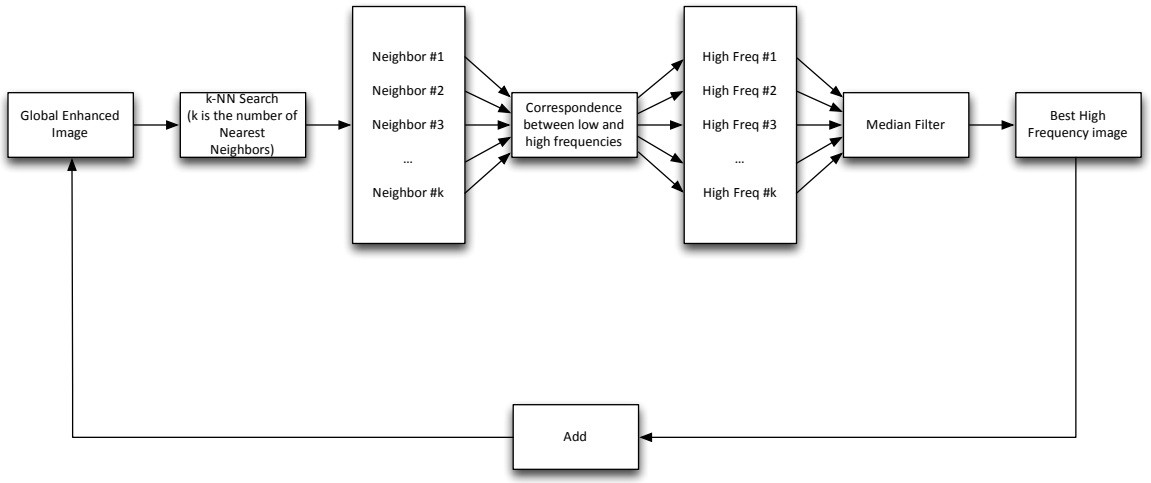


Figure 2.13: Full image ANN Search Process

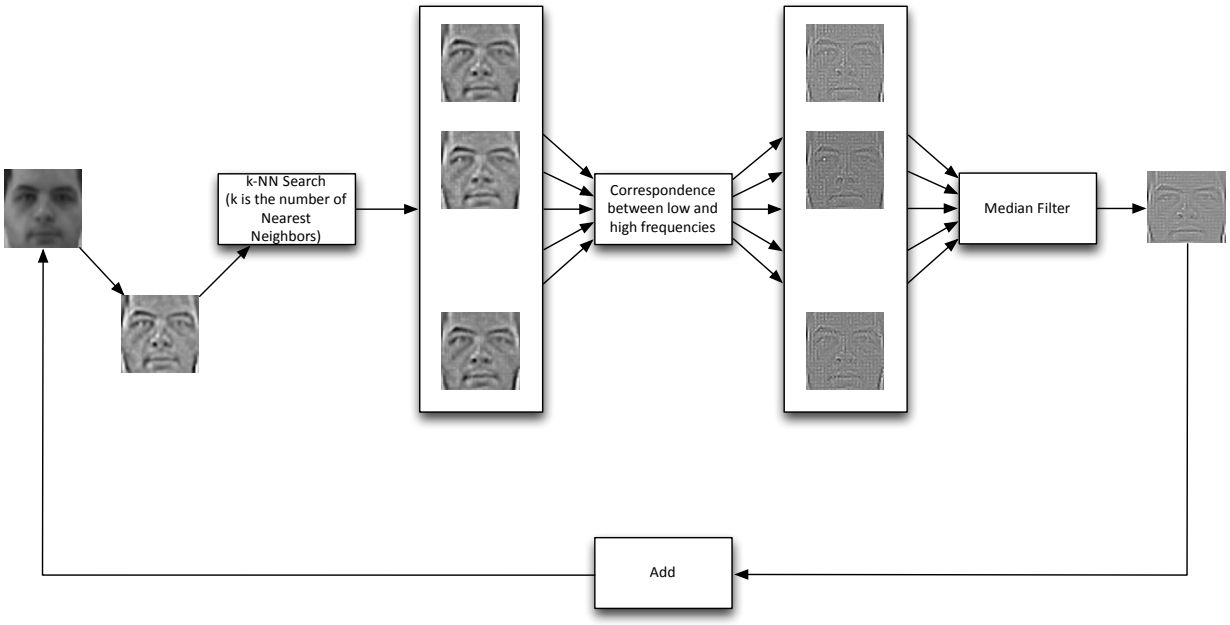


Figure 2.14: ANN Search more detailed process

Chapter 3

Face Recognition

The facial recognition system is a computer vision application for automatically identifying or verifying an individual from a digital image or a video frame. Face recognition is a research field in constant development. Traditional algorithms of face recognition uses faces from still images as an input query [2, 3, 4]. When the goal is to implement a face recognition system running in video surveillance, other approach is used instead of still images. In this case, it is used multiple video sequences of the same subject in order to exploit extra information available in video sequences like variations in resolution, illumination, pose and facial expressions. All of these contributions tend to increase the performance of a video-based face recognition system.

There are many face recognition systems based on video sequence [18, 19, 20, 21, 22]. This work follows an approach based on sparse coding. Sparse coding works modelling data vectors as sparse linear combinations of basis elements. It is becoming widely used in machine learning, neuroscience, signal processing and statistics.

In a video-based face recognition performance can be significantly improved by using the temporal and extra information present in a video instead of using frame-based approaches. To face the challenges of face recognition from unconstrained videos, it is used a generative approach based on dictionary learning methods. In this work it is closely followed the Chen's *et al.* [23] approach, which is reported as being more effective than others video-based face recognition algorithms. Chen's algorithm can be divided in three main steps. In the first step, it is described a method to split a video sequence into K different partitions. This step increases the performance of the process because each partition has a different condition such as illumination, pose or facial expressions. The second step concerns the dictionary learning based on video sequences. Each partition is used to learn a partition-specific dictionary (sub-

dictionary) via sparse coding, which is a robust process to represent the initial partition by the best linear combination of a number $\tilde{B}_a$ basis of a learned dictionary. Given a number of K partitions, a sequence-specific dictionary is created by combining all K sub-dictionaries from a video sequence. The training step is very important because it stores multiples sequences per person. Each sequence has K different conditions imposed by the sequence partition step. The last step is a process of identification and verification where an input video sequence is projected onto the span of atoms of every sequence-specific dictionary in order to compute and combine the residuals to perform recognition or validation.

Figure 3.1 shows an overview of this approach.

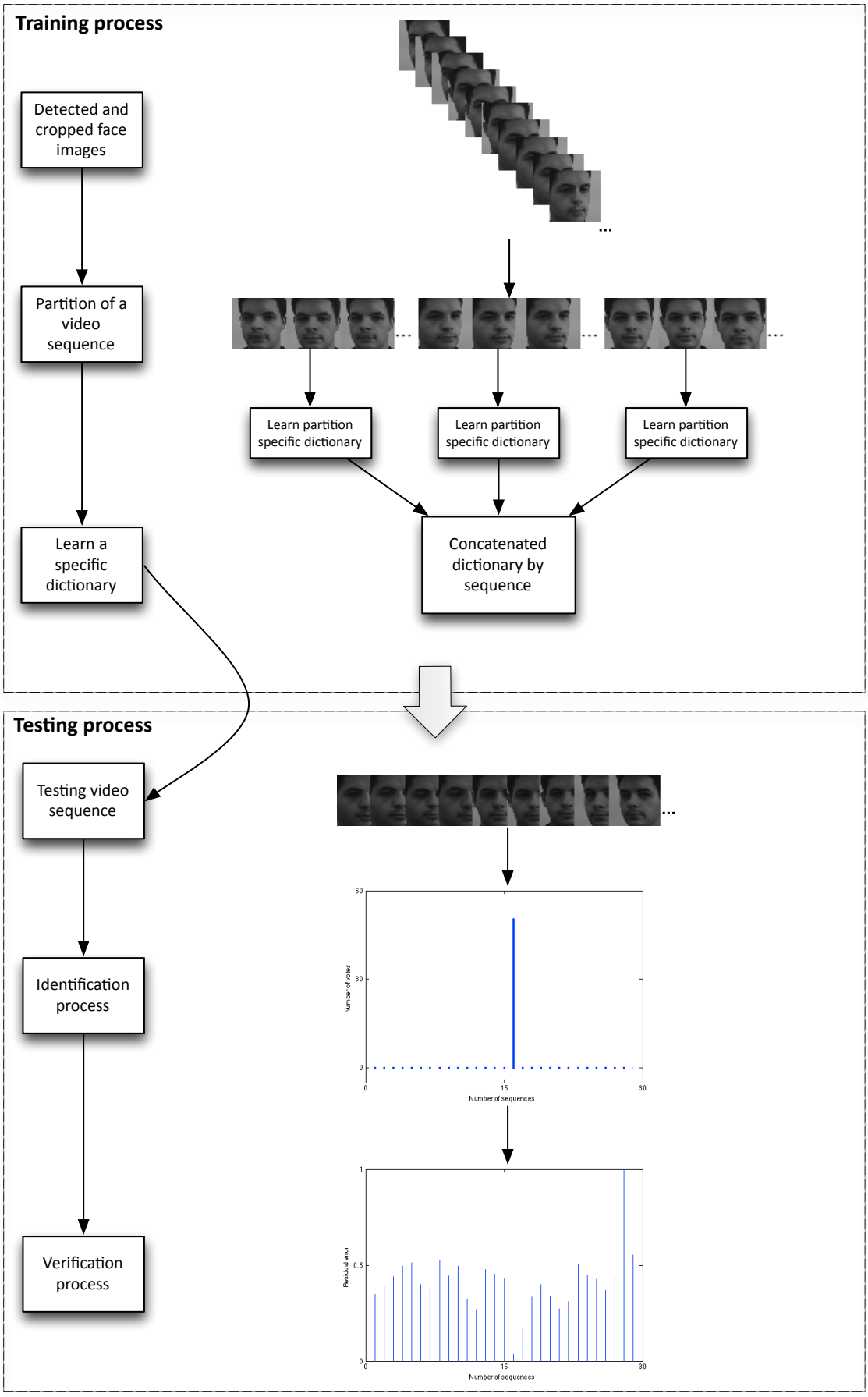


Figure 3.1: Face recognition process overview

3.1 Sequence Partition

Given a video sequence of cropped face images, the goal is to divide the video sequence into K different partitions. Let $S = \{f_1, f_2, \cdots, f_n\}$ be the set of all n cropped faces from a video sequence. The partitions are initialized deterministically. To divide S into K partitions it is necessary to choose the initial K representative images as far apart as possible. The corresponding images to each partition are determined by using a partition criterion. This criterion is based on minimizing the euclidean distance between all images from S and the initial representatives images. Since the first K partitions are already created, the partition algorithm keeps updating them over N iterations, in order to achieve the best combination possible for the partitions. The final combination is chosen based on the maximization of $M(S)$ which is given by 3.1

$$M(S) \triangleq \frac{div(S)}{err(S)}, \tag{3.1}$$

where $err(S)$ and $div(S)$ are the *square error* measure and the *diversity* measure of summary $S(\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_K)$, respectively. These two measures are represented by equations 3.2 and 3.3, respectively [34],

$$err(S) \triangleq tr\Big[\sum_{i=1}^{K}\sum_{\mathbf{s}\in S_i}(\mathbf{s}-\mathbf{s}_i)(\mathbf{s}-\mathbf{s}_i)^T\Big], \tag{3.2}$$

and

$$div(S) \triangleq tr\Big[\sum_{i=1}^{K}(\mathbf{s}-\bar{\mathbf{s}})(\mathbf{s}-\bar{\mathbf{s}})^T\Big], \tag{3.3}$$

where $\bar{\mathbf{s}} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{s}_i$ and $tr(\mathbf{A})$ denotes the trace of matrix $\mathbf{A}$. The *diversity* measure represents the scatter of representatives to their mean. The *square error* measure represents the total summation of partition-specific scatters, over all K partitions. The maximization of score $M(S)$ is achieved through maximizing the *diversity* while minimizing the *square error*.

This video sequence partition approach is summarized at the algorithm 1.

Algorithm 1 Video Sequence Partition

Initialization of sets:

$S = \{f_1, f_2, \cdots, f_n\}$; $I = \{1, 2, \cdots, n\}$; $T = \phi$.

Procedure:

1. Find $(i^*, j^*) = \arg\max_{i,j \in I, i \neq j} \|f_i - f_j\|$.
2. Update of sets: (a) $t_1 \leftarrow i^*, t_2 \leftarrow j^*$; (b) $T \leftarrow T \bigcup \{t_1, t_2\}$; (c) $I \leftarrow I\{i^*, j^*\}$.
3. Find $k^* = \arg\max_{k \in I} \prod_{l=1}^{|T|} \|f_{t_1} - f_k\|_2$.
4. Update of sets: (a) $t_{|T|+1} \leftarrow k^*$; (b) $T \leftarrow T \bigcup \{t_{|T|+1}\}$; (c) $I \leftarrow I\{k^*\}$.
5. Repeat steps 3 and 4 until $|T| = K$ (K is the number of partitions).
6. Given $\{f_{t_1}, \cdots, f_{t_K}\}$ (initial images for each partition), use a criterion to partition S into K partitions, for example the euclidean distance between two images. $S(f_{t_1}, \cdots, f_{t_K})$ is the initial partition, denoted by $S(f_{t_1}, \cdots, f_{t_K}) = \bigcup_{i=1}^{K} S_i$, which are followed by N iterations of updating described in step 7 and 8.
7. Randomly select s_i from S_i, $i = 1, 2, \cdots, K$, as representative. Find the corresponding nearest images partitions which are denoted by $S(s_1, s_2, \cdots, s_K)$, and calculate the score $M\Big(S(s_1, s_2, \cdots, s_K)\Big)$ for each iteration.
8. Repeat step 7 and keep updating for $\{s_1^*, s_2^*, \cdots, s_K^*\}$ which gives the highest score M, until the number of repeating iterations for step 7 reaches N. In other words,

$$\{s_1^*, s_2^*, \cdots, s_K^*\} = \underset{s_i \in S_i, i=1,2,\cdots,K, \text{ in } N \text{ iterations}}{\arg\max} M\Big(S(s_1, s_2, \cdots, s_K)\Big)$$

Output:

K partitions, $S(s_1^*, s_2^*, \cdots, s_K^*)$.

To illustrate the results of the proposed algorithm of video sequence partition, figure 3.2 shows the output from the algorithm with $K = 3$ partitions. Results are presented for three video sequences with different subjects and illumination conditions. The red lines are dividing each video sequence.

Figure 3.2: Video sequence partition results from 3 video sequences

3.2 Dictionary Learning

A dictionary is learned for each partition of a video sequence to remove the temporal redundancy while capturing variations due to changes in pose and illumination. A dictionary is learned via sparse coding with a least square problem with quadratic constraint. Sparse coding consists in modelling data vectors as a best linear combination of basis elements. Sparse coding is becoming widely used in image processing and it is very useful to learn dictionaries adapted to small patches, which training data that may include several millions of these patches. It is used for many applications like compression, regularization in inverse problems, feature extraction, and more. In this case, it is used to reduce the size of a dictionary in order to decrease the computational time in the step of searching against a dictionary.

There will be K sub-dictionaries built to represent a video sequence. The number of face images in a partition will vary, due to changes in pose and/or lighting in a video sequence. In cases where partitions have a small number of face images, an augment is done by introducing synthesized face images. This is done by creating a new image by shifting it horizontally, vertically and diagonally. Assume that each partition contains N_g face images. Let $\mathbf{G}^i_{j,k}$ be the augmented gallery matrix of images from partition k of the jth video sequence of subject i,

$$\mathbf{G}^i_{j,k} = \left[\mathbf{g}^i_{j,k,1}, \mathbf{g}^i_{j,k,2}, \cdots, \mathbf{g}^i_{j,k,N_g}\right] \in \mathbf{R}^{L \times N_g}, \tag{3.4}$$

where each column is the vectorized form of an image of size L. Given the augmented

matrix $\mathbf{G}^i_{j,k}$, a dictionary $\mathbf{D}^i_{j,k} \in \mathbf{R}^{L \times \tilde{B}_a}$ is learned such that the columns of $\mathbf{G}^i_{j,k}$ are best represented by linear combinations of $\tilde{B}_a$ basis of $\mathbf{D}^i_{j,k}$. In other words, the dictionary $\mathbf{D}^i_{j,k}$ can be represented by a specific number $\tilde{B}_a$ of basis against a gallery with N_g vectorized images where $N_g \geq \tilde{B}_a$. To learn the dictionary $\mathbf{D}^i_{j,k}$ it is used a sparse coding method to minimizing the following representation error

$$\begin{aligned} \left(\hat{\mathbf{D}}^i_{j,k}, \hat{\mathbf{\Gamma}}^i_{j,k}\right) &= \underset{\mathbf{D}^i_{j,k}, \mathbf{\Gamma}^i_{j,k}}{\operatorname{argmin}} \left\| \mathbf{G}^i_{j,k} - \mathbf{D}^i_{j,k} \mathbf{\Gamma}^i_{j,k} \right\|^2_F + \lambda \sum_{i=1}^{N_g} \left\| \mathbf{\Gamma}^i_{j,k} \right\|_1 \\ \text{subject to } & \left\| \mathbf{D}^i_{j,k_l} \right\|_2 \leq 1 \quad \text{for} \quad 1 \leq l \leq \tilde{B}_a, \end{aligned} \tag{3.5}$$

where $\mathbf{\Gamma}^i_{j,k}$ is the coefficient matrix. λ is the regularization parameter. $\| \cdot \|_F$ denotes the Frobenius norm. To compute equation (3.5) it is used the Lagrange dual method presented in [32] to learn the dictionary $\mathbf{D}^i_{j,k}$.

All of these dictionaries learned for each partition of a specific-sequence are grouped together in order to create a new dictionary. For each subject i and its jth sequence it is created a dictionary containing K concatenated sub-dictionaries learned by sparse coding. This sequence-specific dictionary is represented by:

$$\mathbf{D}^i_j = \left[\mathbf{D}^i_{j,1}, \mathbf{D}^i_{j,2}, \cdots, \mathbf{D}^i_{j,K} \right]. \tag{3.6}$$

Figure 3.3 shows an example of a learned dictionary via sparse coding (Lagrange dual method) with raw images. It was used four different video sequences which are separated by the green line.

3.3 Identification

The identification is a voting process. Let Q denote the total number of query video sequences. Given the mth query video sequence $\mathbf{Q}^{(m)}$, where $m = 1, 2, \cdots, Q$, $\mathbf{Q}^{(m)}$ can be write as $\mathbf{Q}^{(m)} = \cup_{k=1}^{K} \mathbf{Q}^{(m)}_k$. Partitions $\mathbf{Q}^{(m)}_k$ are denoted by $\mathbf{Q}^{(m)}_k = \left[\mathbf{q}^m_{k,1} \quad \mathbf{q}^m_{k,2} \quad \cdots \quad \mathbf{q}^m_{k,n_k} \right]$, where $\mathbf{q}^m_{k,l}$ is the vectorized form of the lth of the total n_k cropped face regions belonging to the kth partition. Each subject i has a number j of video sequences, and P is the total number of video sequences in the gallery. Thus, the sequence-specific dictionary can be expressed by $\mathbf{D}_{(p)}$ for $p = 1, 2, \cdots, P$ and each $\mathbf{D}_{(p)}$ corresponds to $\mathbf{D}^i_j$ for some subject i and its j video sequence. Image $\mathbf{q}^{(m)}_{k,l}$ votes for sequence $\hat{p}$ with the minimum residual as shown

Figure 3.3: Dictionary learned via sparse coding example with raw images

in 3.7

$$\hat{p} = \underset{p}{\operatorname{argmin}} \left\| \mathbf{q}_{k,l}^{(m)} - \mathbf{D}_{(p)} \mathbf{D}_{(p)}^{\dagger} \mathbf{q}_{k,l}^{(m)} \right\|_2, \tag{3.7}$$

where $\mathbf{D}_{(p)}^{\dagger} = \left(\mathbf{D}_{(p)}^T \mathbf{D}_{(p)} \right)^{-1} \mathbf{D}_{(p)}^T$ is the pseudoinverse of $\mathbf{D}_{(p)}$ and $\mathbf{D}_{(p)} \mathbf{D}_{(p)}^{\dagger} \mathbf{q}_{k,l}^{(m)}$ is the projection of $\mathbf{q}_{k,l}^{(m)}$ onto the span of atoms in $\mathbf{D}_{(p)}$.

The result of 3.7, $\hat{p}$, returns all the votes of each partition from each sequence, so it is necessary to make a sequence-level decision in order to obtain the sequence containing the maximum number of votes, so p^* is obtained by

$$p^* = \underset{p}{\operatorname{argmax}} \left(\sum_{k=1}^{K} w_k C_{p,k} \right), \tag{3.8}$$

where $C_{p,k}$ is the total number of votes from partition k for sequence p and w_k is the weight associated with partition $\mathbf{Q}_k^{(m)}$. To find out the correspondence between the subject and the highest voted sequence, p^*, it is created a correspondence function $m(\cdot)$ to assign the query video sequence $\mathbf{Q}^{(m)}$ to subject $i^* = m(p^*)$.

Figure 3.4 shows an example of resulting votes for each sequence from the dictionary and the highest value is the one which is closer to the input video sequence.

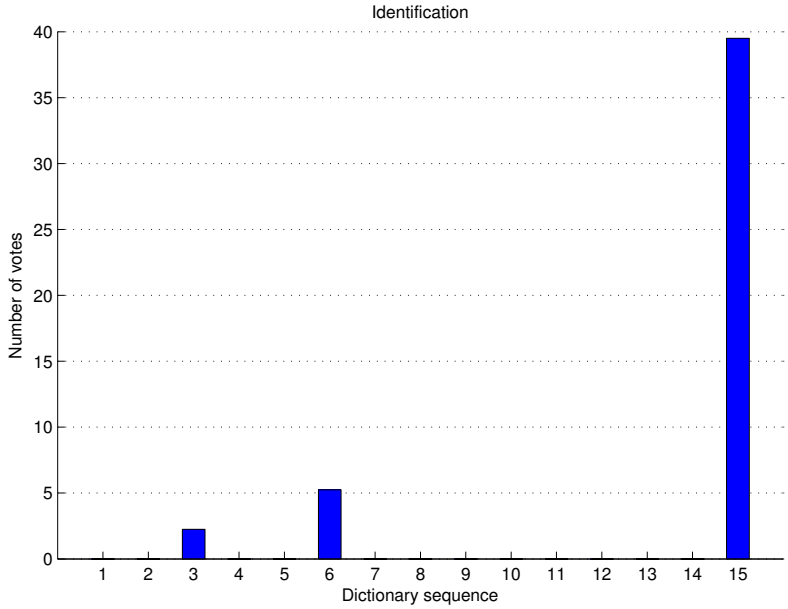


Figure 3.4: Identification vote system example

Given a video sequence and after compute the equation (3.7) and (3.8) the result is the number of votes for a sequence specific dictionary that are closest to the input video sequence. In this case, figure 3.4 tells that sequence number 15, from all sequence-specific dictionary, is the closest to the input video sequence, which has almost 40 votes and probably the input video sequence is from the same subject of the sequence 15 of the dictionary.

3.4 Verification

The goal of verification is to correctly determine whether a query video sequence and any gallery video sequence belong to the same subject. Receiver Operating Characteristic (ROC) curve describes the relations between False Acceptance Rates (FARs) and True Acceptance Rates (TARs) and it can be used to evaluate the performance of verification algorithm. The ROC curve can be computed creating a similarity matrix, the similarity matrix $\mathbf{R}^{(m,p)}$ is the residual error between a query $\mathbf{Q}^{(m)}$ and a dictionary $\mathbf{D}_{(p)}$ and it is expressed by

$$\mathbf{R}^{m,p} = \min_{k \in \{1,2,\cdots,K\}} \mathbf{R}_k^{(m,p)}, \tag{3.9}$$

where

$$\mathbf{R}_k^{(m,p)} \triangleq \min_{l \in \{1,2,\cdots,n_k\}} \left\| \mathbf{q}_{k,l}^{(m)} - \mathbf{D}_{(p)} \mathbf{D}_{(p)}^{\dagger} \mathbf{q}_{k,l}^{(m)} \right\|_2. \tag{3.10}$$

The minimum residual is computed against all $l \in \{1, 2, \cdots, n_k\}$ and all $k \in \{1, 2, \cdots, K\}$ as the similarity between the query video sequence $\mathbf{Q}^{(m)}$ and dictionary $\mathbf{D}_{(p)}$. Figure 3.5 shows the similarity between the input video sequence and all dictionaries $\mathbf{D}_{(p)}$ with $p = \{1, 2, \cdots, P\}$.

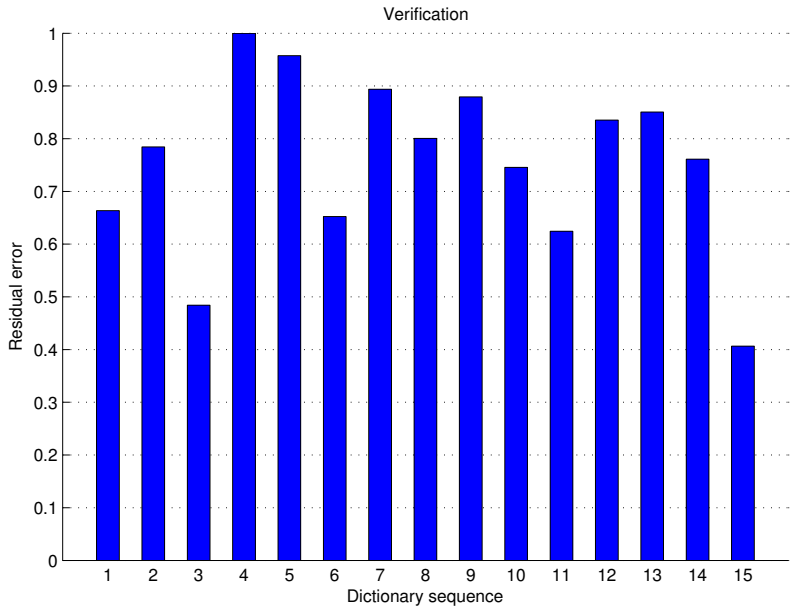


Figure 3.5: Verification example

Using the same video sequence like in figure 3.4, the result of the verification is shown in figure 3.5. Here the objective is to determine the minimum residual error between an input video sequence and a dictionary. In this case, the minimum value is for sequence-specific dictionary number 15.

To summarize this chapter of face recognition based on video sequences, algorithm 2 shows all steps to complete a validation. Identification and verification processes are combined in order to get the final decision about the success of the face recognition system.

3.5 Validation

The validation of the face recognition system is done by combining the information from the identification and verification processes (as related in previous sections). This information is combined through the score function (3.11)

$$V = \frac{\text{Identification}}{\text{Verification}}, \tag{3.11}$$

Algorithm 2 Proposed Dictionary-based Face Recognition from Video (DFRV)

Training process:

1. Extract all cropped face regions from a set S_j^i, where j is the video sequence from subject i.
2. Divide S_j^i into K partitions by using video sequence partition algorithm (algorithm 1)
3. Learn the partition specific dictionary $D_{j,k}^i, \forall\, k = 1, 2, \cdots, K$ via sparse coding. Create the sequence specific dictionary D_j^i by concatenating all the partition specific dictionaries.

Testing process:

1. Partition the mth query video sequence $\mathbf{Q}^{(m)} = \cup_{k=1}^{K} \mathbf{Q}_k^{(m)}$, where $\mathbf{Q}_k^{(m)} = \begin{bmatrix} \mathbf{q}_{k,l}^m & \mathbf{q}_{k,2}^m & \cdots & \mathbf{q}_{k,n_k}^m \end{bmatrix}$
2. Identification step. Use 3.7 to determine the vote from $\mathbf{q}_{k,l}^{(m)}$, $\forall k, l$. Then, use 3.8 and the correspondence function between the subject and sequence $m(\cdot)$ to make the final decision.
3. Verification. Find the similarity $\mathbf{R}^{(m,p)}$ between $\mathbf{Q}^{(m)}$ and $\mathbf{D}_{(p)}$ by equations 3.9 and 3.10.

where the identification is the total number of votes for a sequence-specific dictionary that is closest to the input video sequence, and the verification is the residual error between an input video sequence and each sequence-specific dictionary.

To ensure that the subject of the input video sequence corresponds to a sequence of the dictionary it is necessary that the maximum value of votes from identification and the maximum value of the score V (equation (3.11)) belong to the set of sequences from the same subject in the dictionary. If this requirement is met, the face recognition based on video sequences will be validated. Figure 3.6 shows an example of the result of the validation score V for same sequences used as input in figures 3.4 (identification) and 3.5 (verification)

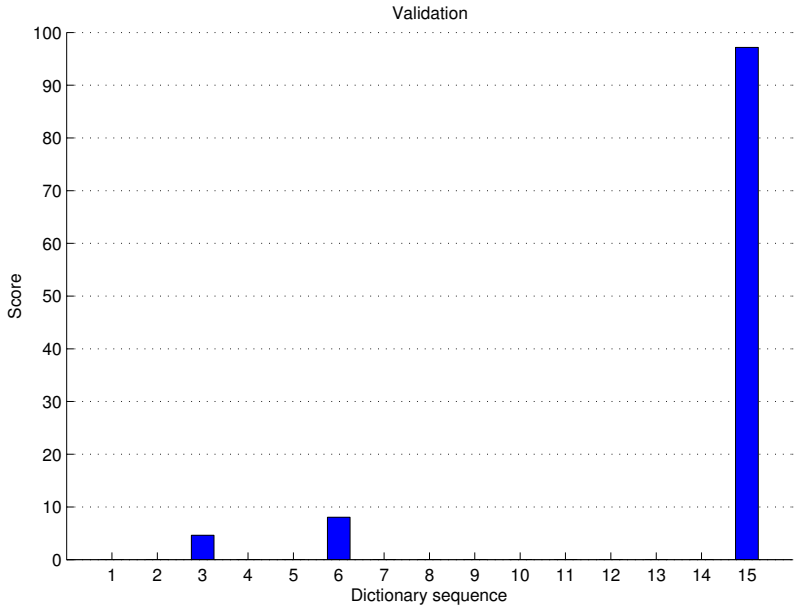


Figure 3.6: Validation score V example

In this case, the subject of the sequence used as an input belongs to the dictionary because

the highest number of votes (from the identification process at figure 3.4) and the maximum value of validation score (figure 3.6) belong to the same sequence-specific dictionary number (number 15). Thus, the validation is successful.

3.6 Image Descriptors

An image descriptor is a description of the visual features of the contents in images and it describes the characteristics of an image such as color, shape and others.

In this work, two different descriptors were used to learn the dictionary and to the identification process. The first descriptor is basically the appearance encoded on the gray-level of the face image (raw image). The second descriptor used is based on Region Covariance Matrix (RCM). A new way, proposed by Pang *et al.* [35], to use RCM is used to further enhance the discriminating ability of RCMs. This new approach is based on Gabor filters.

3.6.1 Raw Image

A raw image contains minimal data from the digital camera and it was used for many years in computer vision, e.g., [36], [37] and [38]. The raw image pixel information can be color, gray-level, gradient and filter responses. In this case, it is used the gray level image as a descriptor to learn dictionaries via sparse coding. These types of descriptors are not robust in the presence of illumination variation and it is not desired.

3.6.2 Region Covariance Matrices

Region Covariance Matrix (RCM), proposed by Tuzel *et al.* [39] is a covariance matrix of several image statistics computed inside a region of interest. $\mathbf{I}$ is an one dimensional intensity image (gray level) of size $W \times H$. Mapping function ϕ extracts d dimensional feature vector $\mathbf{z}_i$ from pixel (x, y) of $\mathbf{I}$, i.e.,

$$\phi(\mathbf{I}, x, y) = \mathbf{z}_i \in \mathbf{R}^d, \tag{3.12}$$

where $i = y \times W + x$ is the index of (x, y). A region R is defined by all pixels (x, y), i.e., $(x, y) \in R$ and the number of elements of R is n. That region can be represented by a $d \times d$ covariance matrix of the feature points $\mathbf{z}_i$ inside R

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{z}_i - \mathbf{u}_R)(\mathbf{z}_i - \mathbf{u}_R)^T, \tag{3.13}$$

where $\mathbf{u}_R$ is the mean of the points and is expressed by

$$\mathbf{u}_R = \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i. \tag{3.14}$$

In [39] they proposed a descriptor for object recognition where the mapping function ϕ is defined by pixel location (x, y), all components of RGB color and the norm of the first and second order derivatives of the intensities (gray level) with respect to x and y. This function is represented by

$$\phi(\mathbf{I}, x, y) = \mathbf{z}_i = \begin{bmatrix} x & y & R(x,y) & G(x,y) & B(x,y) & |\mathbf{I}_x| & |\mathbf{I}_y| & |\mathbf{I}_{xx}| & |\mathbf{I}_{yy}| \end{bmatrix}^T, \tag{3.15}$$

where

$$\mathbf{I}_x = \frac{\partial \mathbf{I}(x,y)}{\partial x} \quad \mathbf{I}_y = \frac{\partial \mathbf{I}(x,y)}{\partial y} \quad \mathbf{I}_{xx} = \frac{\partial^2 \mathbf{I}(x,y)}{\partial x^2} \quad \mathbf{I}_{yy} = \frac{\partial^2 \mathbf{I}(x,y)}{\partial y^2}. \tag{3.16}$$

Mapping function 3.15 is used for object detection, but for human detection, Tuzel *et al.* [40] defined a new mapping function

$$\phi(\mathbf{I}, x, y) = \mathbf{z}_i = \begin{bmatrix} x & y & \mathbf{I}(x,y) & |\mathbf{I}_x| & |\mathbf{I}_y| & |\mathbf{I}_{xx}| & |\mathbf{I}_{yy}| & \theta(x,y) \end{bmatrix}^T, \tag{3.17}$$

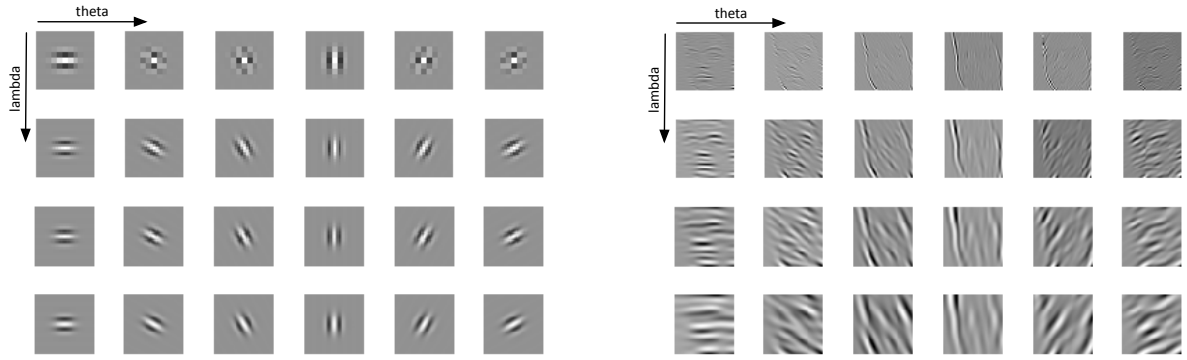
where $\theta(x, y)$ is the orientation component computed by

$$\theta(x,y) = \arctan\left(\frac{|\mathbf{I}_y|}{|\mathbf{I}_x|}\right). \tag{3.18}$$

RCM is a symmetric matrix (3.13) where the diagonal elements represent the variance of each feature and the nondiagonal elements represent their respective correlations. Compared to others descriptors the covariance matrix $\mathbf{C}_R$ has only $(d^2 + d)/2$ different elements, because it is symmetric. Thus, it is only used the bottom or upper triangular matrix from the covariance matrix as an image descriptor.

3.6.2.1 Gabor-based Region Covariance Matrices

Gabor-based Region Covariance Matrix (GRCM) was proposed by Pang *et al.* [35] and it introduces a new type of mapping function to a region. It is used 2-D Gabor kernels because it exhibit strong characteristics of spatial locality, scale, and orientation selectivity. Gabor filters have been showing great success in face representation [41]. Gabor filters can exploit salient visual properties such as spatial localization, orientation selectivity, and



(a) 2-D Gabor kernel family

(b) Gabor filters applied

Figure 3.7: Gabor kernel family

spatial frequency characteristics [42, 43]. The 2-D Gabor kernel is defined by a sinusoidal wave multiplied by a Gaussian function. The filter has two components: real and imaginary. The two components can be formed into a complex number or used individually, in this case it is used one individually and it is represented by

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \psi\right), \tag{3.19}$$

where

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -y\sin\theta + y\cos\theta. \end{aligned} \tag{3.20}$$

In these equations, λ is the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of the Gaussian envelope and finally γ is the spatial aspect ratio and it specifies the ellipticity of the support of the Gabor function. The Gabor features can be obtained by convolving the Gabor kernels with an image I as follow:

$$G_{(u,v)}(x, y) = \mathbf{I}(x, y) * g(x, y; \lambda, \theta, \psi, \sigma, \gamma). \tag{3.21}$$

A Gabor filter family is created by taking N_θ orientations and N_λ different wavelengths. So for each pixel (x, y) the dimensionality of the Gabor features is $N_\theta \times N_\lambda = N_{Total}$. In figure 3.7(a) it is shown an example of a family of Gabor filter with $N_\theta = 6$ orientations and $N_\lambda = 4$ wavelengths.

In equation 3.21, (u, v) are the index of the orientation and wavelength, respectively. In order to create a Gabor filter family, $u \in [0, 1, \cdots, N_\theta - 1]$ and $v \in [1, \cdots, N_\lambda]$. In figure

3.7(b) are shown the results of the convolution between an image with all the Gabor family and they are used to build a RCM.

The mapping function based on Gabor filter is represented by:

$$\mathbf{z}_i = \Big[x \quad y \quad G_{(0,1)}(x,y) \quad G_{(0,2)}(x,y) \quad \cdots \quad G_{(N_\theta - 1, N_\lambda)}(x,y) \Big]. \tag{3.22}$$

This new mapping function can be substituted in (3.13) to obtain the GRCM in region R, i.e. $\mathbf{C}_R$. As the covariance matrix $\mathbf{C}_R$ can be represented by their bottom or upper triangular matrix, the size of GRCM is drastically reduced, for example: given an image I (320×240), a region R with size of I and a mapping function like (3.22), the number of elements of I is 76800 pixels. Computing a GRCM with six different orientations ($N_\theta = 6$) and four different wavelength ($N_\lambda = 4$), the mapping function ($\mathbf{z}_i$) has size of $d = 24$ plus the pixel location (x, y), so $d = 26$. The size of $\mathbf{C}_R$ is $d \times d = 26 \times 26$. In fact, there are only $(d^2 + d)/2 = (26^2 + 26)/2 = 351$ different values due to symmetry of $\mathbf{C}_R$. This shows that RCM is independent of the image size.

The intensity component of the image $\mathbf{I}(, x, y)$ can also be added to the mapping function (3.22) resulting this new mapping function:

$$\mathbf{z}_i = \Big[x \quad y \quad \mathbf{I}(x,y) \quad G_{(0,1)}(x,y) \quad G_{(0,2)}(x,y) \quad \cdots \quad G_{(N_\theta - 1, N_\lambda)}(x,y) \Big]. \tag{3.23}$$

To increase robustness for possible occlusions and illumination variations, a single face image is represented by five different regions. Figure 3.8 shows the five RCMs (C1, C2, C3, C4 and C5) used to combine all regions into a vector.

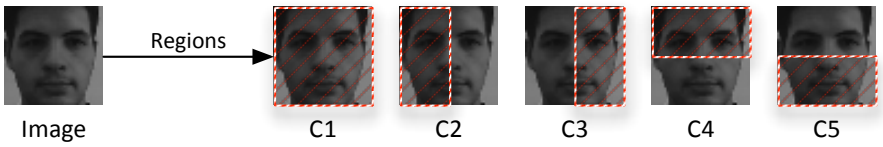


Figure 3.8: Five different regions of a single face image

In this case, a face image can be represented by five RCMs extracted from five different regions (figure 3.8). The C1 region is a global representation of the face because it is applied to the entire face. All the others regions (C2,C3,C4 and C5) are part-based representations of the face. The GRCM descriptor is defined by concatenating all results from the covariance matrix of each region into a column vector. This global and part-based combination increases the robustness to occlusions and illumination changes.

Chapter 4

Experimental Results

This work is divided in two parts. In the first part it was developed an algorithm in MATLAB to increase image resolution (face hallucination) of a face image. Secondly, it was created an algorithm of face recognition in C++, on UNIX operating system (Ubuntu).

To develop the face recognition algorithm it was used the OpenCV[1] library with a linear algebra library called Armadillo[2] and for the sparse coding dictionary learning it was used the mlpack machine learning library [44]. The video sequences used in this work were obtained using a PC webcam.

4.1 Face Detection

One of the main applications of face recognition is surveillance for security purpose, which involves real-time recognition of faces from an image sequence acquired by a video camera. A video-based face recognition system can be divided in three steps: face detection, face tracking and face recognition. The first step of any face processing system is detecting the locations in images where faces are present. However, face detection from a video-based sequence is a challenging task because of: low quality images, cluttered backgrounds, the presence of more than one face in the frame, and a large amount of data to process. In these cases, a robust face detection system must be used to perform the face detection followed by a tracking system. These systems are not the focus of this dissertation. For our purposes, it is used a face detection system available in OpenCV library, which allows a reduced range of head pose variation, but reveals to be suitable for our purposes. Face detection system

[1] `http://opencv.org/`
[2] `http://arma.sourceforge.net/`

consists of a previously trained classifier based on Haar-like[3] features. The classifier is trained with several hundred images of a particular object, in this case frontal face, called positive examples, and negative examples are arbitrary images of the same size as positive examples. This classifier is properly provided by OpenCV and it can be applied to a region of interest in an input image. It returns true when the input image corresponds to the classifier object (frontal face images). The object is searched in the whole image by moving a search window across the image and check every location using the classifier. The function available in OpenCV library has some particular parameters like the cascade input file, the input image for detection which corresponds to the actual frame, the scale factor which specifies how much the image size is reduced at each image scale, the minimum neighboring number of each rectangle should have to retain it, and the minimum possible object size. Object smaller than the minimum possible object size are ignored. The parameters used for the OpenCV face detection function are:

- Input cascade file $\rightarrow$ *haarcascade_frontalface_alt.xml*,
- Scale factor $\rightarrow$ 1.2,
- Minimum neighboring number $\rightarrow$ 4,
- Minimum possible object size $\rightarrow$ 45×45.

The frontal face classifier provides good results to our purposes. The range of the frontal face orientation is not very wide but it allows sufficient head pose variation for the purpose of face recognition. This range is illustrated in figure 4.1. Some examples of sequences used

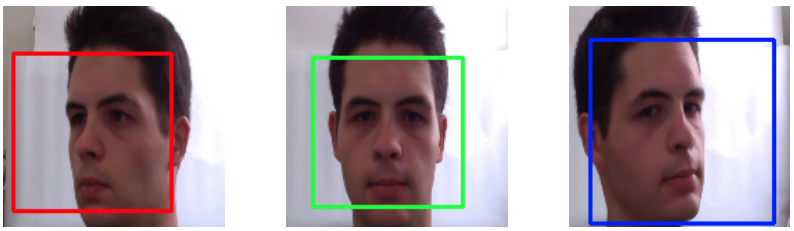


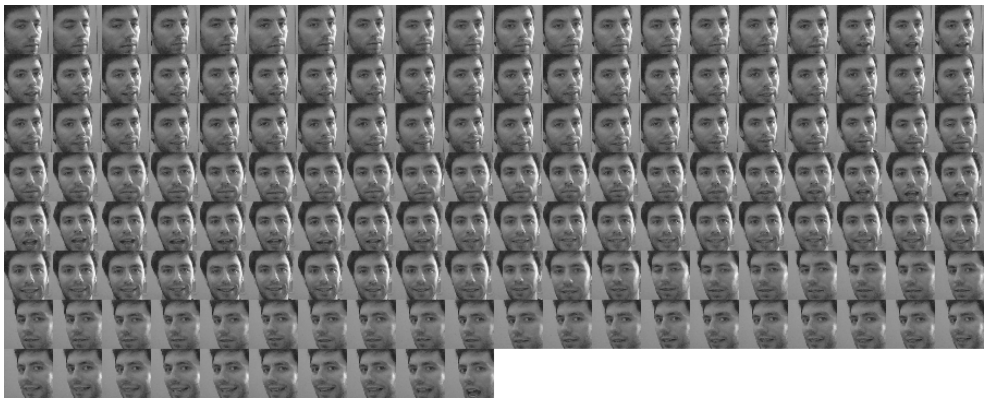
Figure 4.1: Maximum head pose variation

in our work are illustrated in figures 4.2 and 4.3.

[3] `http://docs.opencv.org/2.4.3/modules/objdetect/doc/cascade_classification.html`

Figure 4.2: Sequence with similar pose over time

(a) Sequence 1



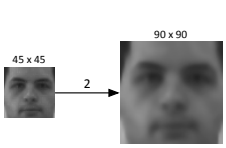
(b) Sequence 2

(c) Sequence 3

Figure 4.3: Sequences with different conditions over time

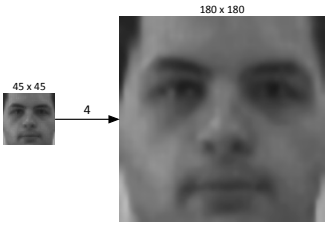
4.2 Face Hallucination

As mentioned before in chapter 2, face hallucination is Super-Resolution (SR) of face images. This process consists in combining multiple low resolution images to form a higher resolution image. The input low resolution images are like the video sequence illustrated in figure 4.2 where the face orientation is very similar over time. The minimum size of an acquired face image is 45×45 and one of the stages of the hallucination process includes an increase of the image size. In this stage it is used a bicubic interpolation to generate the interpolated high resolution image $\bar{I_H}$. Figure 4.4 shows some examples of this interpolation (different values of upgrade factor U were used). This stage is applied to an input video sequence (like figure



(a) $U = 2$

(b) $U = 3$



(c) $U = 4$

Figure 4.4: Interpolated face image with different upgrade factors

4.2) in order to generate a database of low and high frequency interpolated images. These images are created by applying low and high pass filters. Figure 4.5 shows the low and high frequency interpolated images of a single image from the sequence.

After generating a database with low and high frequency interpolated images, a database of eigenfaces is created by PCA training. Eigenfaces database is an over-complete dictionary because one query difference face can only match a small number of eigenfaces with similar shape. Thus, an image can be well-represented as a sparse linear combination of elements from an over-complete dictionary. The regularization parameter λ is related to the number of basis used to reconstruct an image. The bigger value of λ, the lower is the number of

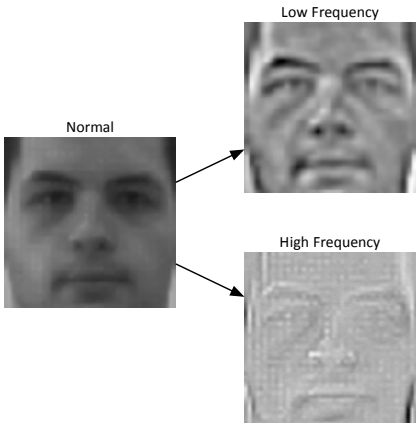


Figure 4.5: Example of an image and its low and high frequency representation

basis, and vice-versa. This number of basis variation is illustrated in figure 4.6.

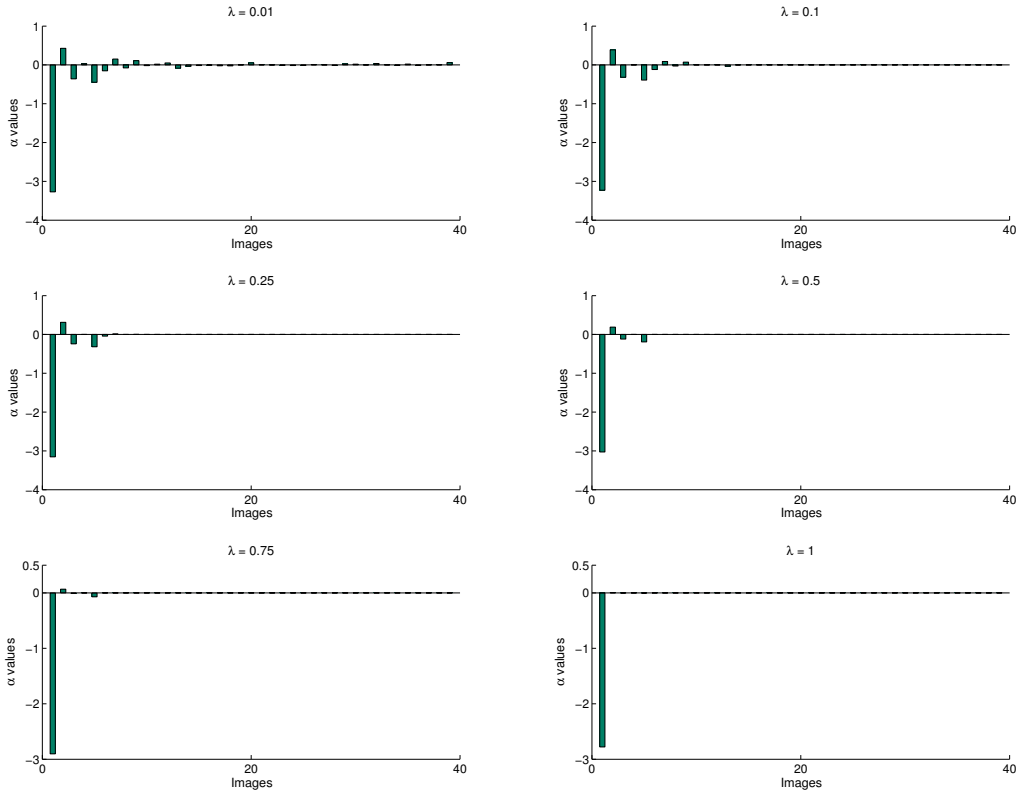


Figure 4.6: Sparse coefficients α with different values of λ

In this case, the final reconstructed image uses a sparse representation with few basis, it is used $\lambda = 0.5$. This value provides a sparse representation with a few number of basis, which includes one higher weight and a few others with low weights. To solve the ℓ_1-regularized least square (equation (2.7)) it is used the MATLAB function developed at [45].

The resulting image from the sparse linear combination is added to the interpolated

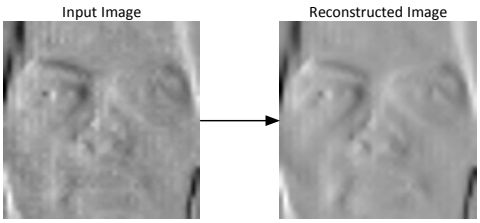
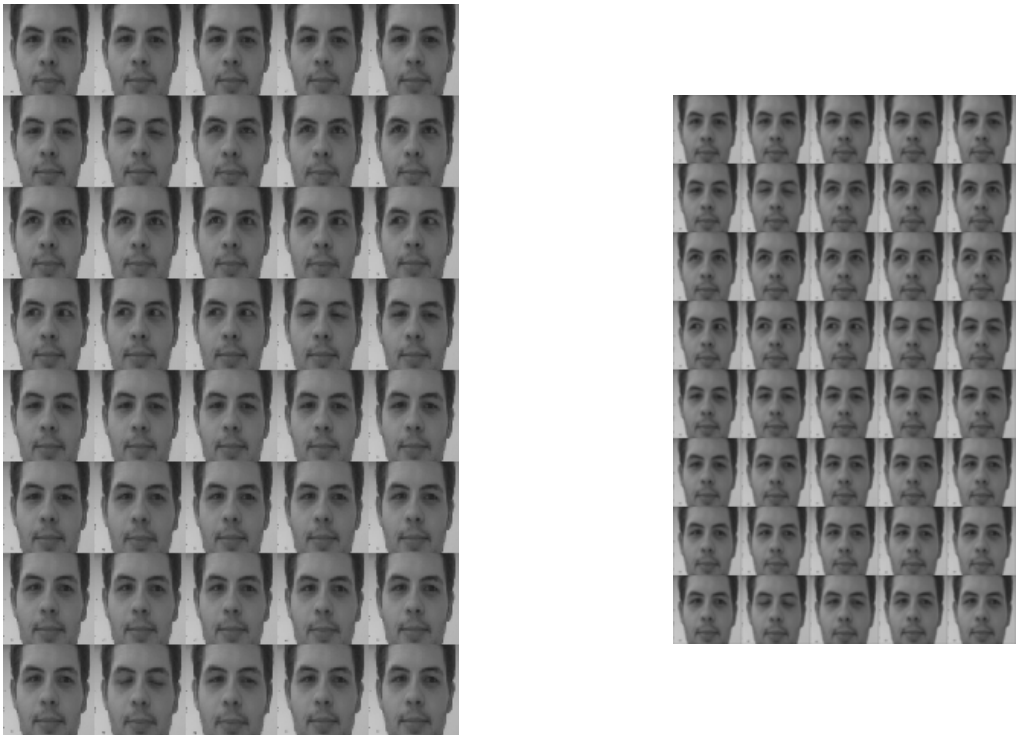


Figure 4.7: Reconstructed image from an eigenface database

high resolution image in order to increase the image detail ($I_{H,GlobalEnhanced}$). After that, $I_{H,GlobalEnhanced}$ is filtered by bilateral filtering to remove noise and artifacts. The parameters of bilateral filtering are the half-size of the Gaussian filter window (W) and the standard deviations ($[\sigma_1, \sigma_2]$), the spatial-domain standard deviation (σ_1) and the intensity-domain standard deviation (σ_2). The values used for these parameters are $W = 5$ and $\sigma_1 = 3$, $\sigma_2 = 0.1$. These values were obtained after performing some tests and these values were the best combination to reduce noise and artifacts.

After noise reduction, $I_{H,GlobalEnhanced}$ is further enhanced by Jia's method and returned as the final hallucinated face. This method do not increase the image resolution but simply add more high frequency information. This method is based on ANN search and it is computed by using a MATLAB function *knnsearch*[4]. The input parameters of this function are the input query image, which is the low frequency image of $I_{H,GlobalEnhanced}$, the $NSMethod$ which is the nearest neighbor search method ($kdtree$) and the parameter related to the number of nearest neighbors k. The number of nearest neighbor k is chosen based on the final result of face hallucination. To choose this value it was created a video sequence (40 images) with high resolution face images. Then, high resolution faces were filtered by a Gaussian filter in order to decrease the image quality. After that, a bicubic interpolation is used to reduce the size of high resolution face images to half-size. Thus, it is generated a sequence of low resolution face images. Figure 4.8 shows these two generated sequences.

[4] `www.mathworks.com/help/stats/knnsearch.html/`



(a) High resolution face sequence (b) Low resolution face sequence

Figure 4.8: Sequences used to validate the process

Using the low resolution face image sequence, face hallucination is applied in order to compare the final hallucinated face image with the original high resolution face image. Figure 4.9 shows the resulting hallucinated face with different values of nearest neighbor k.

To check the differences between the high resolution face image and the corresponding hallucinated using different values of k nearest neighbors, figure 4.10 shows the error between original high resolution face image and its hallucinated face.

Observing figure 4.10, the parameter of k nearest neighbors may be switched between 10 and 20 depending on the length of the input low resolution sequence. The face hallucination presented by Jia uses the information retrieved from a stack of tracked faces to generate the final high resolution face image. This process has a low computational time because of the online training. The hallucinated faces of this method are illustrated in figure 4.11.

Jia's method does not use a finding process to search the exact match for each low resolution patch like other methods [1, 26]. These methods use an offline dictionary trained with thousands of low and high resolution patches. The image enhancement is done by searching the low resolution patch from the dictionary. This searching information is not sufficient for SR, so a Markov network is used to model the relationships between high and

(a) Hallucinated face with $k = 5$

(b) Hallucinated face with $k = 10$

(c) Hallucinated face with $k = 15$

(d) Hallucinated face with $k = 20$

Figure 4.9: Hallucinated face with different k nearest neighbors

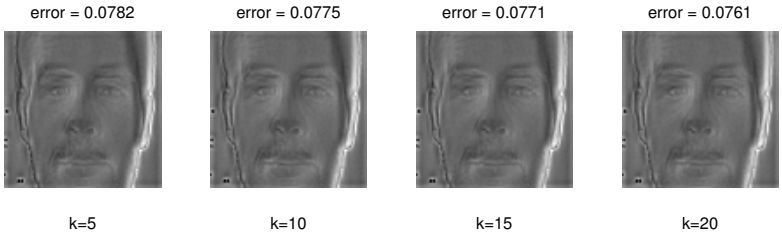


Figure 4.10: Error between original high resolution face image and its hallucinated face

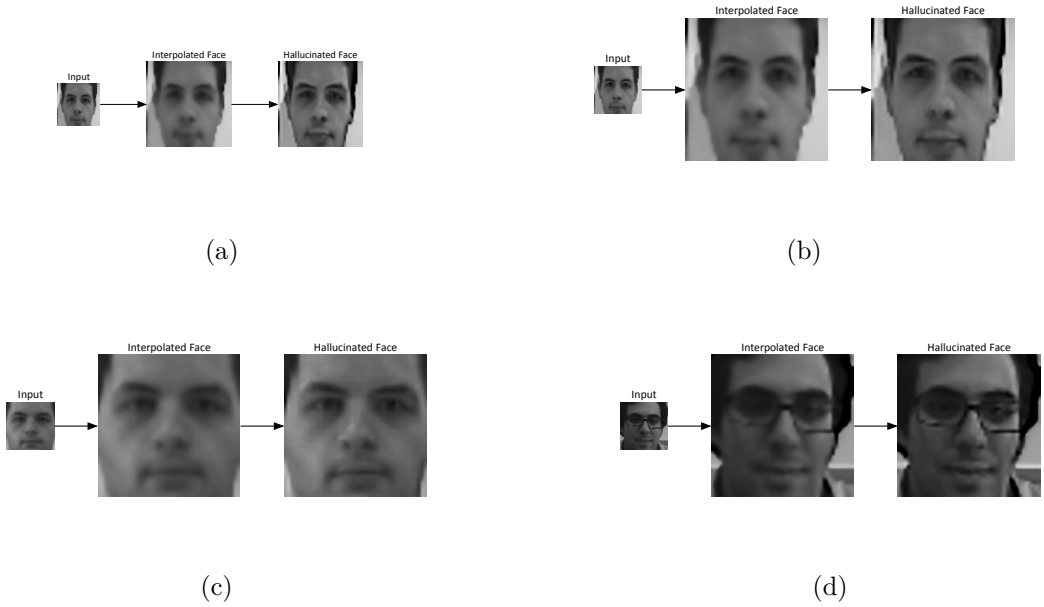


Figure 4.11: Set of final hallucinated faces

Figure 4.12: Face SR using online code

low frequency patches, and between neighboring high frequency patches. This step is very time consuming, which is a major difference comparing with the Jia's method. Figure 4.12 shows the steps to achieve the final SR face image using this method, which is available online[5].

4.3 Face Recognition

In video-based face recognition, a key challenge is exploiting the extra information available in video. In addiction, different video sequences of the same subject may contain variations in resolution, illumination, pose, and facial expressions. A generative approach based on dictionary learning methods is followed to minimize the challenges of recognition from videos.

[5] `http://people.csail.mit.edu/billf/project%20pages/sresCode/Markov%20Random%20Fields%20for%20Super-Resolution.html`

(a) Partition 1 (b) Partition 2 (c) Partition 3

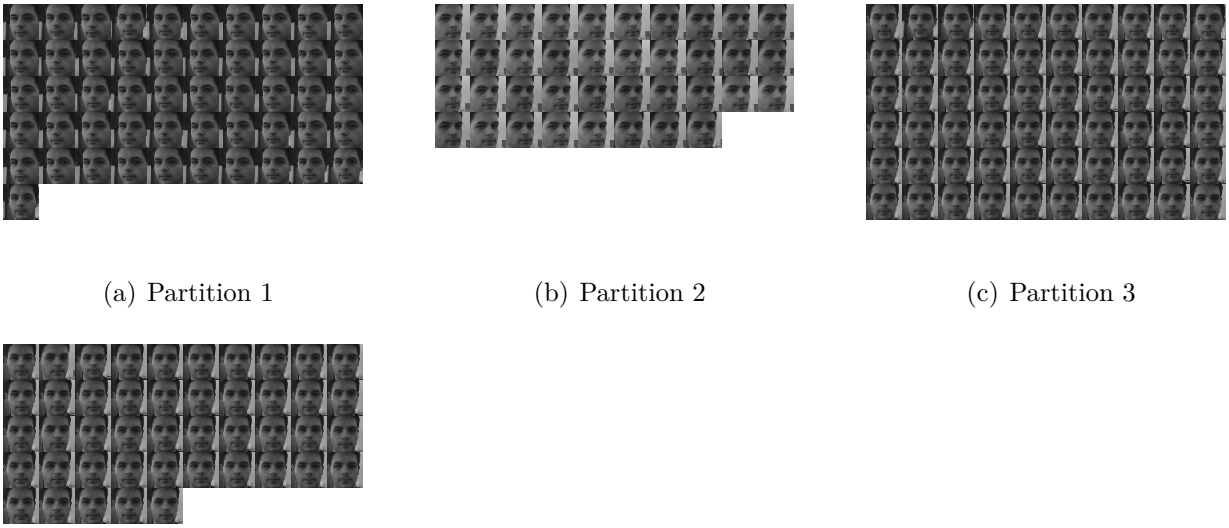
Figure 4.13: Sequence partition with $K = 3$

4.3.1 Video Sequence Partition

Given a video sequence of face images (like figure 4.3), the first step is to divide a video sequence into different partitions in order to achieve a set of images with the same conditions. Each partition encodes a particular pose and illumination condition. This partition step removes the temporal redundancy while capturing variations due to changes in pose and illumination.

Recalling the algorithm 1 presented in section 3.1. This algorithm divides a video sequence into K different partitions. Furthermore, this algorithm keeps updating the partitions over N iterations. For each iteration it is calculated the corresponding score $M(S)$ (equation 3.1). The maximization of $M(S)$ is achieved through maximizing the diversity while minimizing the square error. The final partitions are chosen with the highest score $M(S)$. Figures 4.13, 4.14 and 4.15 show the result of partition algorithm with different K values over $N = 100$ iterations. Each partition should have a minimum number of image because of dictionary learning step. To learn the dictionary it needs several images by partition to compute successfully the Lagrange dual method via sparse coding and to avoid the random initializations of basis elements of the dictionary.

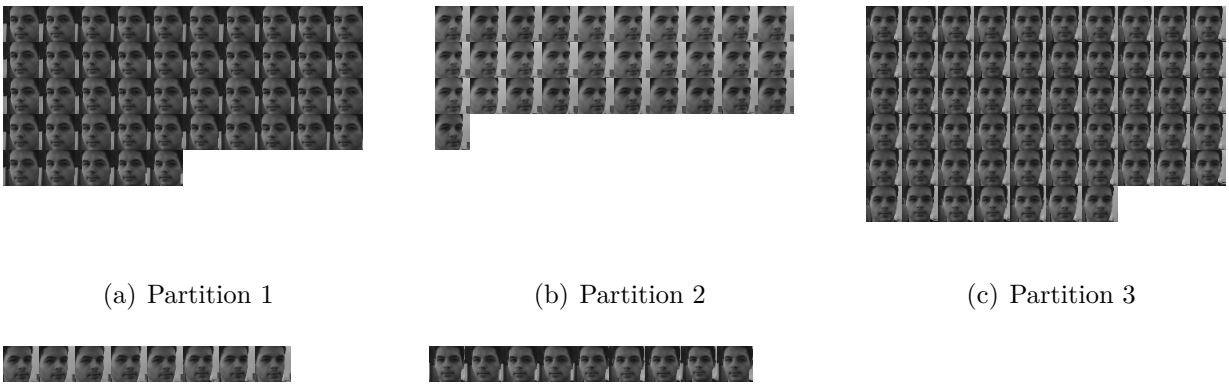
It was created a video sequence with many different conditions, such as pose, illumination and scale. The video sequence partition algorithm is used to split the sequence into K partitions and the result is shown in figure 4.16.



(a) Partition 1 (b) Partition 2 (c) Partition 3

(d) Partition 4

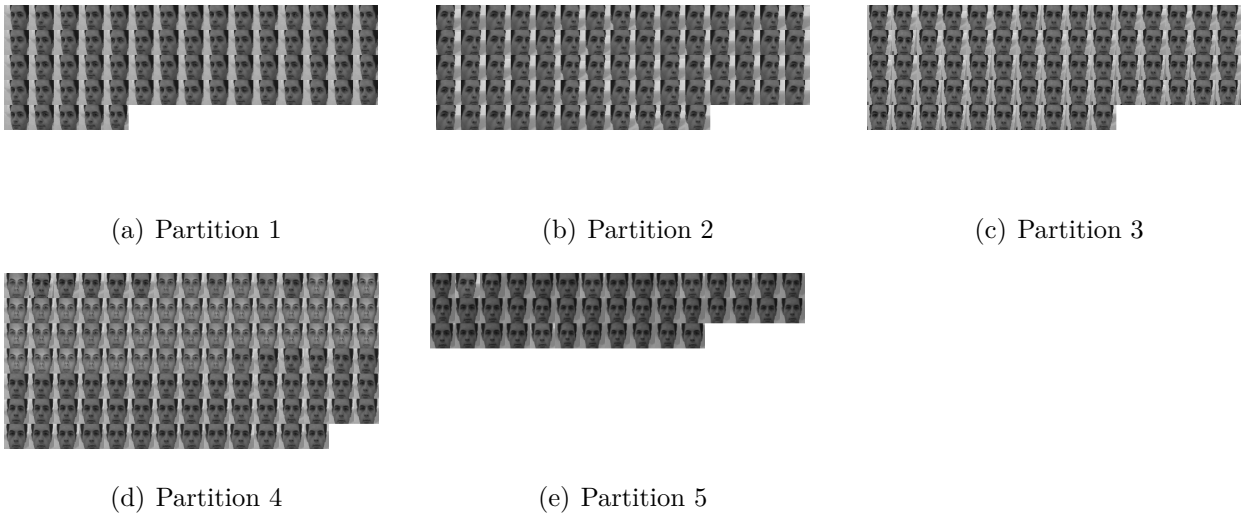
Figure 4.14: Sequence partition with $K = 4$



(a) Partition 1 (b) Partition 2 (c) Partition 3

(d) Partition 4 (e) Partition 5

Figure 4.15: Sequence partition with $K = 5$



(a) Partition 1 (b) Partition 2 (c) Partition 3

(d) Partition 4 (e) Partition 5

Figure 4.16: Sequence partition with $K = 5$ with some variations

4.3.2 Dictionary Learning

Each partition contains images with specific pose and/or illumination conditions. A dictionary is created for each partition in order to remove the temporal redundancy while capturing variations due to changes in pose and illumination. Thus, there will be K sub-dictionaries by video sequence. Before building any sub-dictionary, an augment of gallery images is done for the partitions with very few images. This increase is done by introducing synthesized face images, i.e., images are created by shifting a random image from the partition horizontally,vertically and diagonally. Different from Chen *et al.* [23], a dictionary is learned via sparse coding with a least square problem with a quadratic constraint.

An implementation of sparse coding with dictionary learning is available in the mlpack machine learning library [44] and the input parameters are the regularization parameter λ and the number of basis $\tilde{B}_a$ (the dimension of the learned feature space).

A sequence-level dictionary is created by grouping all the K sub-dictionaries by video sequence. Figure 4.17 shows an example of a sequence-level dictionary learned via sparse coding with $\tilde{B}_a = 7$ basis and $K = 3$ partitions. Green lines separates different sequence-level dictionaries and each row is the learned sub-dictionary via sparse coding. The α values were obtained using $\lambda = 0.01$. In this case, it is used a small value of λ to reduce the sensitivity of the optimization, i.e., the reconstructed images use more elements from the original dictionary (each partition from a video sequence). This value is low because the optimization is related to the number of images at the original dictionary and with a low value of λ, more elements from the dictionary are used.

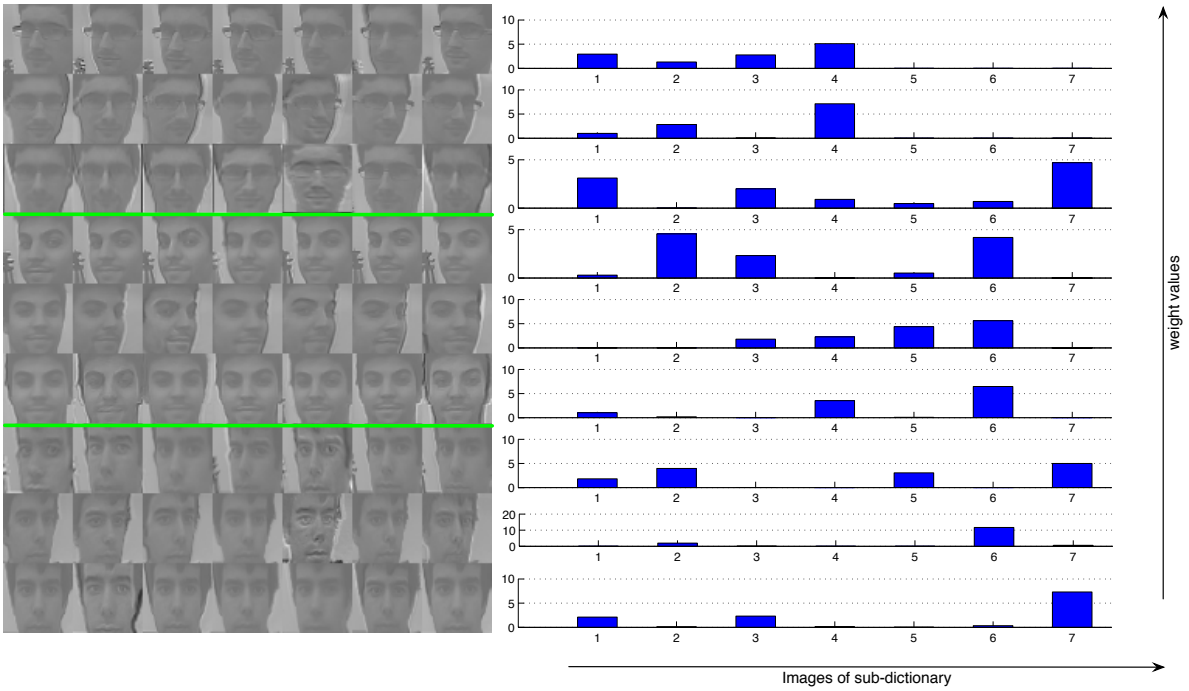


Figure 4.17: Sequence-level dictionary learned with $\tilde{B}_a = 7$ and $K = 3$

4.3.3 Image Descriptors

As described in section 3.6, two types of descriptors are used here. Firstly, raw images are used as an image descriptor. Figure 4.17 shows an example of a sequence-level dictionary learned with the gray level information of image. This kind of descriptor can not be used with face hallucination and face recognition at the same time. Given a video sequence, the size of any image is always the same. Thus, the dictionary based on that video sequence is directly related to the image size. Face hallucination process increases image size, so this new hallucinated face sequence images have bigger size than the sequence images used to learn the dictionary. Considering this, the new hallucinated face sequence can not be compared with the previous dictionary learned. To address this problem, a descriptor invariant to the image size is used. The descriptor used is a matrix of covariance of several image statistics computed inside a region of an image, Region Covariance Matrix (RCM).

4.3.3.1 RCM

The RCM is considered as a feature descriptor of the region. The RCM based on mapping function (3.17) uses simple features like pixel coordinates (x and y), the first-order gradient, the second-order gradient and the orientation. These features can be computed by using the *Sobel* function available in the OpenCV library. This function has a set of two parameters

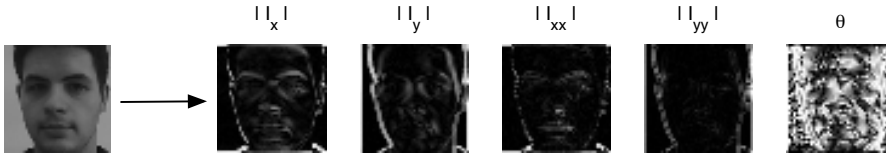


Figure 4.18: Derivative features used to create mapping function (3.17)

Figure 4.19: RCM

(dx and dy) to generate the first or the second-order gradient. The first and second-order gradient for x and y directions are computed by using a combination of these values:

- x order gradient (first-order) $\rightarrow$ $dx = 1$ and $dy = 0$
- y order gradient (first-order) $\rightarrow$ $dx = 0$ and $dy = 1$
- x order gradient (second-order) $\rightarrow$ $dx = 2$ and $dy = 0$
- y order gradient (second-order) $\rightarrow$ $dx = 0$ and $dy = 2$

The mapping function is computed for an image which is divided into five regions (figure 3.8). For each region it is calculated the covariance matrix in order to use only the upper triangular matrix as a region descriptor (because covariance matrix is symmetric). The final descriptor is generated by concatenating all region descriptors into a column vector. Figure 4.18 shows all the features of this mapping function. The resulting covariance matrix computed by using mapping function (3.17) is illustrated in figure 4.19.

4.3.3.2 GRCM

The RCM based on Gabor kernel is represented by the mapping function (3.23). This function is based on several Gabor filters applied to the same image in other to generate a Gabor's filter family to that image. This can be done by varying one or two parameters of Gabor function. This function has some input parameters such as: λ which represents the wavelength of the sinusoidal factor, θ represents the orientation, ψ is the phase offset, γ is the spatial aspect ratio and σ is the sigma of the Gaussian envelope which is calculated by

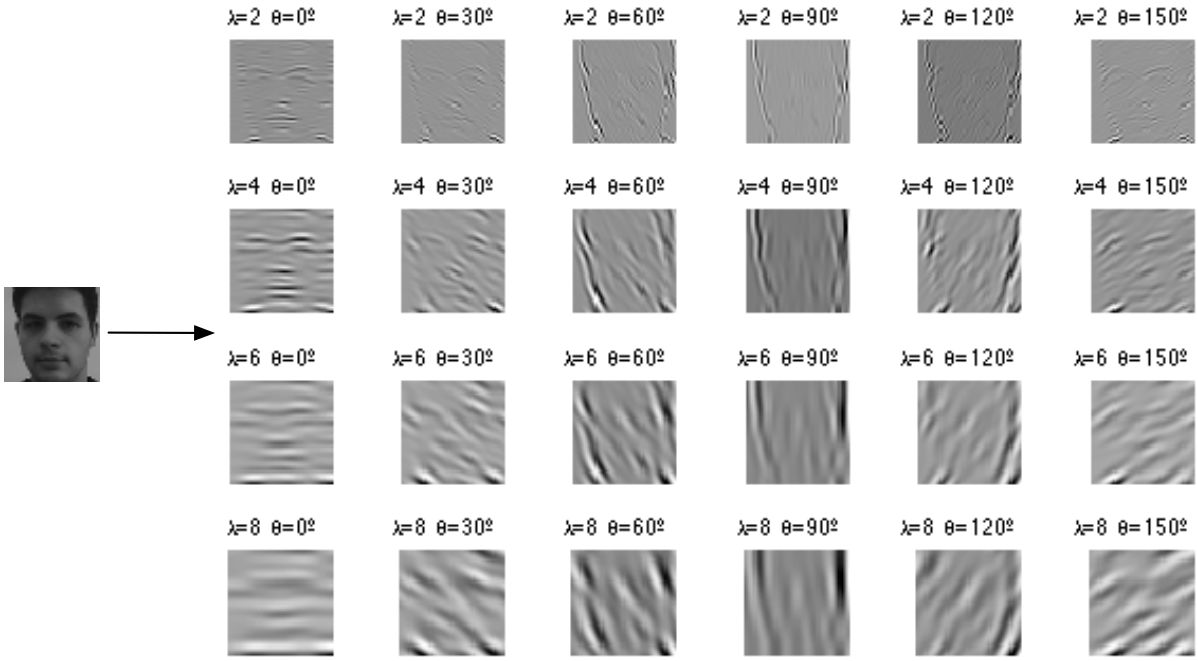


Figure 4.20: Gabor filter's family used in mapping function (3.23)

a given bandwidth bw (equation (4.1)):

$$\sigma = \frac{\lambda}{\pi} \times \sqrt{\frac{\log 2}{2}} \times \left(\frac{2^{bw}+1}{2^{bw}-1} \right). \tag{4.1}$$

The Gabor's filter family are created by varying λ and θ.

Used images have a small size, which may limit the use of some parameters of the Gabor function, which is available online[6] for *C++*. On these images, the edges (facial contours, eyes, mouth, nose, etc.) have a small size due to image size. After Gabor filter application, these edges are highlighted. Based on this, and after some tests, the input parameters used to compute Gabor function are: $\psi = 0$, $\gamma = 0.87$ and $bw = 1$. The other two parameters, λ and θ have a set of different values in order to achieve several wavelengths and orientations. The values used for these parameters are: the set of λ values are $\{2, 4, 6, 8\}$ and the set of θ are $\{0, \pi/6, \pi/3, \pi/2, 2\pi/3, 5\pi/6\}$. Figure 4.20 shows the resulting Gabor's filter family using this set of parameters.

The set of images illustrated in figure 4.20 are used to build the mapping function (3.23). This mapping function is used to compute the covariance matrix based on Gabor features. Figure 4.21 shows this covariance matrix.

[6] `http://www.eml.ele.cst.nihon-u.ac.jp/~momma/wiki/wiki.cgi/OpenCV/Gabor%20Filter.html`

[6] Figures 4.19 and 4.21 are not with the true scale, because they have a small size and to illustrate the resulting RCMs are shown a scaled version.

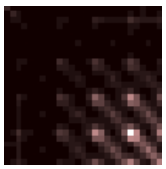


Figure 4.21: RCM generated by mapping function (3.23)

In this work it is used the RCM mapping function (function (3.8)) considering all five regions (figure 3.8), the GRCM only with the global representation of the face (C1 region), GRCM-1, and finally, the GRCM based on the five regions, GRCM-5. Table 4.1 shows the image descriptor size of each of them.

Table 4.1: Image descriptor size comparison

	RCM	GRCM-1	GRCM-5
Descriptor Size (elements)	180	378	1890

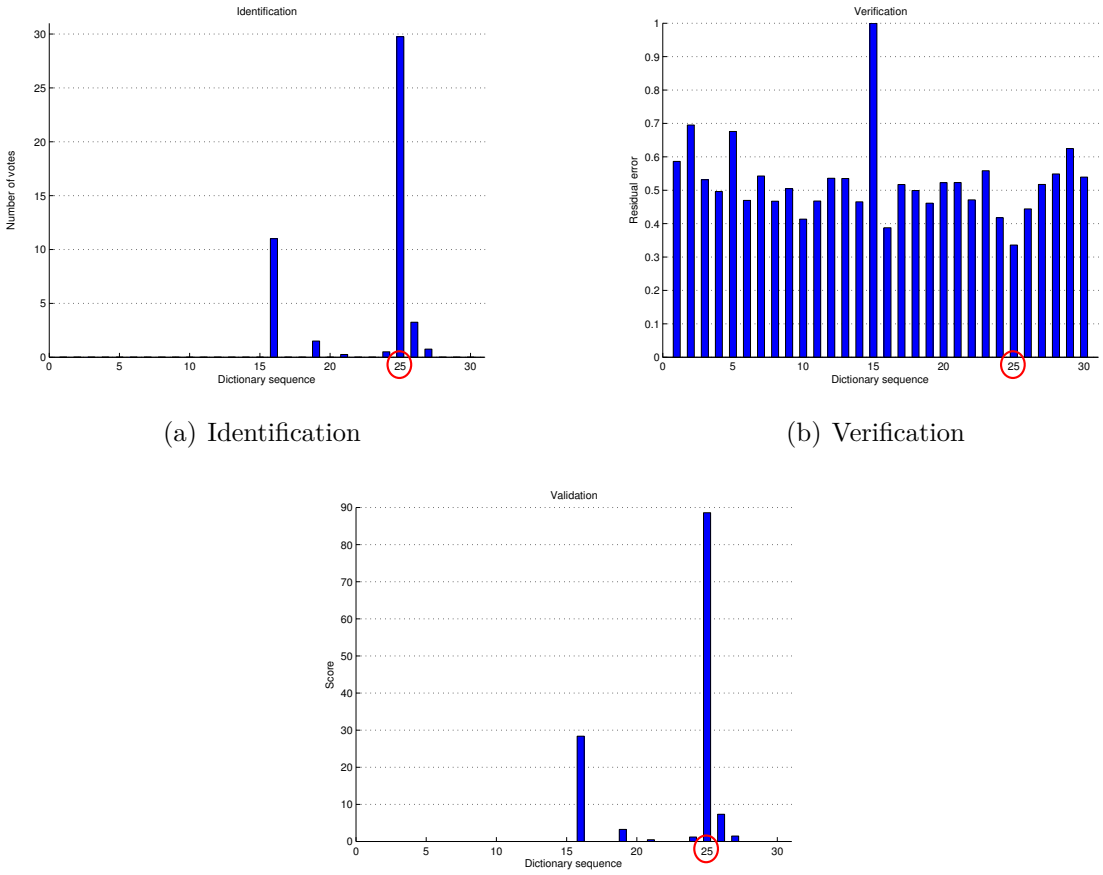
4.3.4 Validation

The validation process consists of combining information from the identification voting process and the residual information from the verification. This combination results in a validation score V. If the highest value of votes (from identification) and the maximum value of validation score belong to the set of sequences of the same subject, the validation will be successful. For example, a dictionary is created by using 3 video sequences of 5 subjects. Lets assume that some subject X belongs to the dictionary. Using another video sequence from the X as an input of face recognition system, the validation is done by combining the identification and the verification process. Figures 4.22(a) and 4.22(b) show the results of these processes, respectively.

In figure 4.22 the red circle represents the values used to perform the validation. In this case, the number of maximum votes is for sequence-specific dictionary number 25, the minimum residual error is for number 25. Thus, the validation will have the maximum value at the sequence-specific dictionary number 25 as figure 4.22(c) shows. So, the validation of subject X is successful.

4.3.5 Discussion

The face recognition algorithm was tested in different conditions. At the first, it was tested the influence of the number of partitions used to create the dictionary. To perform this test



(a) Identification

(b) Verification

(c) Validation

Figure 4.22: Face recognition validation

it was used a total number of 30 sequences of 10 different persons. For validation it was used 30 sequences from the same individuals of the dictionary but with different conditions. Table 4.3 shows the results of each descriptor used for different values of K partitions. These values were calculated based on the confusion matrix (table 4.2 is an example of a confusion matrix for GRCM-5 descriptor and $K = 3$), and the final result is the average of the elements of the diagonal of confusion matrix.

Using a large number of partitions, the performance of the recognition system would increase. Table 4.3 shows that the number of partitions have influence on the success of the recognition. Each partition is related to a specific condition presented in a video sequence such as illumination, pose and others. The robustness of the dictionary tends to increase on the number of conditions in each sequence. The following results are obtained by using $K = 3$ and $K = 4$.

After using different values of K for the video sequence partition, it was tested the influence of the illumination variation. In this case, the dictionary is created by a set of 2 sequences of 9 persons without illumination variation. It was used a set of 3 sequences

Table 4.2: Confusion matrix (%)

Dictionary Sequence	1	2	3	4	5	6	7	8	9	10
1	100	0	0	0	0	0	0	0	0	0
2	0	99.95	0	0	0	0	0	0	0	0.05
3	0	0	92.88	0	0	0	0	2.48	4.62	0
4	7.03	0	0.52	71.83	1.01	13.38	0	6.22	0	0
5	0.92	4.15	9.63	0	67.74	15.67	0	0.99	0.90	0
6	0.30	0	0.35	0	2.61	90.28	0	1.52	4.94	9
7	0	0	0	0	0	0.07	99.93	0	0	0
8	0	0	0	0	0	0	0	100	0	0
9	0	1.28	1.36	2.99	5.66	18.43	0	5.90	64.38	0
10	0	0	0	0	0	0	0	0	0	100

Table 4.3: Recognition accuracy (%) for different values of partitions (K)

	$K = 1$	$K = 2$	$K = 3$	$K = 4$
RCM	83.59	86.51	88.34	91.28
GRCM-1	77.22	84.72	85.90	87.88
GRCM-5	84.80	89.15	88.77	89.36

by individual with illumination variation as the input for recognition. Table 4.4 shows the results of this test.

Table 4.4: Recognition accuracy (%) using a dictionary without illumination changes

	$K = 3$	$K = 4$
RCM	24.70	24.51
GRCM-1	19.57	20.86

On the next step it is created a dictionary with a set of 3 sequences (by individual) with changes in illumination in order to see the results of including this variation. It is used the same sequences of the previous test as the input of recognition system. Table 4.5 shows the results of this test.

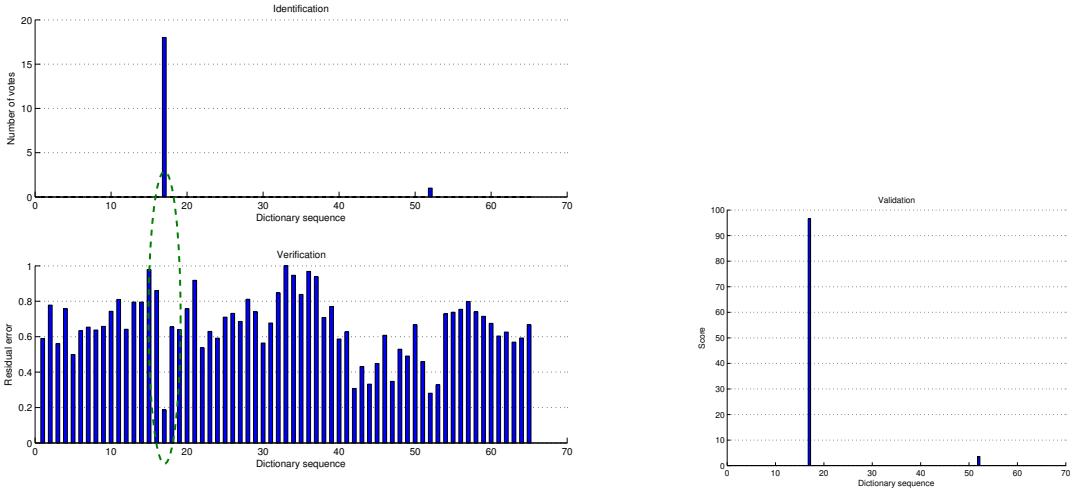
Table 4.5: Recognition accuracy (%) using a dictionary with illumination changes

	$K = 3$	$K = 4$
RCM	92.62	93.28
GRCM-1	88.44	88.95

Tables 4.4 and 4.5 show that using sequences with variations like illumination or pose to create a dictionary, the performance of recognition system would increase.

It was also tested the face hallucination algorithm with the face recognition system. A dictionary is learned with some sequences of hallucinated faces. To evaluate this test it is

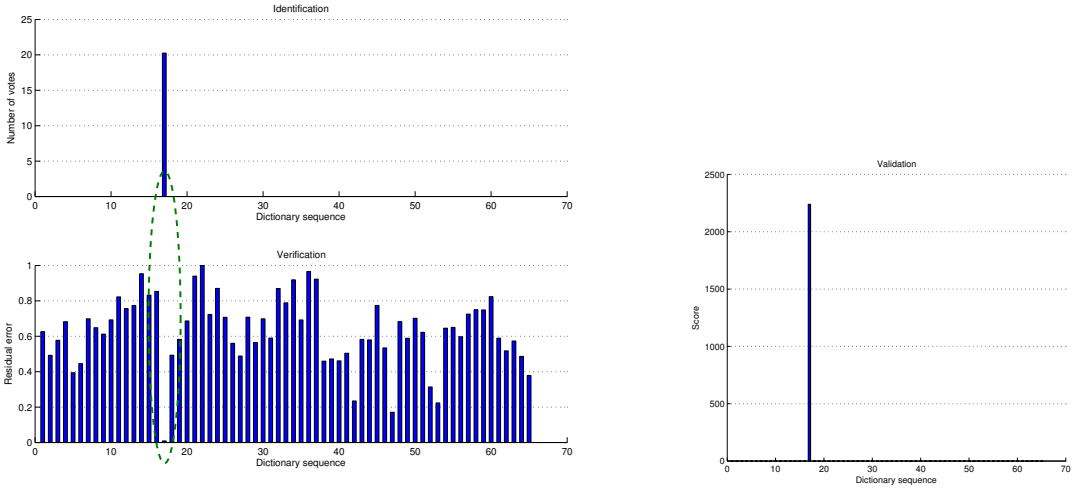
used a sequence of hallucinated faces. Figure 4.23 shows the result for a sequence before applying the face hallucination.



(a) Identification and Verification (b) Validation

Figure 4.23: Sequence before the face hallucination

Figure 4.24 shows the result of the same sequence as used in figure 4.23 but now after the face hallucination.



(a) Identification and Verification (b) Validation

Figure 4.24: Sequence after the face hallucination

Using the sequence with hallucinated faces, the number of votes of identification tend to increase for the sequence of the dictionary corresponding with the same person. The same goes to verification, the value of the residual error tend to decrease for the sequence of the dictionary corresponding with the same person. Thus, the value of the validation score is the highest one, which validates the recognition.

Chapter 5

Conclusion and Future Work

The main focus of this dissertation is face recognition from video surveillance with low resolution images. Sometimes it is important to recognize a face from a video stream, but not always the image resolution is fair enough to achieve good recognition rates. A method of face hallucination is used to increase image resolution. This method yields good results when the tracked faces are very similar to each other. If the tracked face are not well aligned there will be serious "ghost effects" in the resulting images. In this face hallucination algorithm it is used a stack of faces with very similar poses, shapes and illumination conditions. The final hallucinated face is enhanced based on the information retrieved from the stack of tracked faces from the same person.

Concerning to dictionary-based face recognition from video, the temporal redundancy existing while capturing variations due to changes in pose and illumination is removed. This temporal information is removed by using an algorithm to divide a video sequence into different partitions. The partitions are used to learn a dictionary based on sparse coding. Two kind of descriptors were implemented in this work. These descriptors are based on Region Covariance Matrix (RCM). Firstly, it was used the initial approach of RCM's which is based on the image derivatives. Secondly, it was used a RCM based on Gabor filters, Gabor-based Region Covariance Matrix (GRCM). These descriptors yield good results to the face recognition system.

Throughout the implementation of this work, important conclusions and observations have been made. Firstly, it was observed that the face hallucination algorithm is capable to increase image resolution based on a previous stack of low resolution images. Regarding to the dictionary-based face recognition, this method has the capability to efficiently recognize a person given a video sequence. The descriptors used in this method present good results.

However, it was realized that the artificial illumination introduces an undesired variation to video processing. As the descriptors used are based on the image derivatives, these undesired variations have a negative contribute to the effectiveness of the system. Considering this, the results of the recognition were significantly positive.

Concluding, this works presents methods able to enhance image quality based on a sequence of low resolution face images and also able to recognize individuals based on video sequences.

5.1 Future Work

Some issues are still left open to future work. Concerning the context of this dissertation, a face detection and tracking system able to detect and track faces in video where the object of interest is far away from the camera would increase the performance of face hallucination and subsequent recognition. Considering this improvement of face detection and tracking system, a real-time system concerning to face hallucination and then recognition can be implemented. Thus, this system can be used in a real video surveillance applications, such as public and private surveillance, criminal identification, etc. Regarding face recognition system, a dictionary with an online update based on sparse coding would increase the effectiveness of recognition. This online dictionary update could be done by updating a sequence of an individual based on the validation result. For example, if the validation is successful, an update for the sequences corresponding to the same individual will be done in order to increase the robustness to variations in facial appearance, such as beard, hairstyle, etc., and in illumination and/or pose variations.

Bibliography

[1] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *Computer Graphics and Applications, IEEE*, vol. 22, no. 2, pp. 56–65, 2002.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, pp. 399–458, Dec. 2003.

[3] P. Phillips, "Matching pursuit filters applied to face identification," *Image Processing, IEEE Transactions on*, vol. 7, no. 8, pp. 1150–1164, 1998.

[4] V. Patel, T. Wu, S. Biswas, P. Phillips, and R. Chellappa, "Dictionary-based face recognition under variable lighting and pose," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 3, pp. 954–965, 2012.

[5] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 10, pp. 1042–1052, 1993.

[6] M. A. Grudin, "On internal representations in face recognition systems," *Pattern Recognition*, vol. 33, pp. 1161–1177, 2000.

[7] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: Component-based versus global approaches." Computer Vision and Image Understanding, 2003.

[8] T. S. Jebara, "3d pose estimation and normalization for face recognition," tech. rep., Center for Intelligent Machines, McGill University, Undergraduate Thesis, 1996.

[9] R.-J. J. Huang, *Detection Strategies for Face Recognition Using Learning and Evolution.* PhD thesis, George Mason University, Fairfax, Virginia, 1998.

[10] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of The Optical Society of America A-optics Image Science and Vision*, vol. 4, 1987.

[11] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[12] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," 1997.

[14] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: the problem of compensating for changes in illumination direction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 721–732, 1997.

[15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[16] R. Jafri and H. R. Arabnia, "A Survey of Face Recognition Techniques," *Journal of Information Processing Systems*, vol. 5, pp. 41–68, June 2009.

[17] L. Torres, "Is there any hope for face recognition?." Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004)., 2004.

[18] R. Chellappa, V. Kruger, and S. Zhou, "Probabilistic recognition of human faces from video," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–41–I–44 vol.1, 2002.

[19] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Visual tracking and recognition using probabilistic appearance manifolds," *Comput. Vis. Image Underst.*, vol. 99, pp. 303–331, Sept. 2005.

[20] O. Arandjelovic and R. Cipolla, "Face recognition from video using the generic shape-illumination manifold," in *Proceedings of the 9th European conference on Computer Vision - Volume Part IV*, ECCV'06, (Berlin, Heidelberg), pp. 27–40, Springer-Verlag, 2006.

[21] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2273–2286, 2011.

[22] Y. Hu, A. Mian, and R. Owens, "Sparse approximated nearest points for image set classification," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 121–128, 2011.

[23] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition from video," in *Proceedings of the 12th European conference on Computer Vision - Volume Part VI*, ECCV'12, (Berlin, Heidelberg), pp. 766–779, Springer-Verlag, 2012.

[24] S. Baker and T. Kanade, "Hallucinating faces," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 83–88, 2000.

[25] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 9, pp. 1167–1183, 2002.

[26] C. Liu, H. yeung Shum, and W. T. Freeman, "Face hallucination: Theory and practice," *International Journal of Computer Vision*, vol. 75, pp. 115–134, 2007.

[27] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *Image Processing, IEEE Transactions on*, vol. 19, no. 11, pp. 2861–2873, 2010.

[28] Z. Jia, H. Wang, Z. Xiong, and A. Finn, "Fast face hallucination with sparse representation for video surveillance," in *Pattern Recognition (ACPR), 2011 First Asian Conference on*, pp. 179–183, 2011.

[29] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pp. 586–591, 1991.

[30] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust super-resolution," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. II–291–4 vol.3, 2003.

[31] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, pp. 839–846, 1998.

[32] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), pp. 801–808, Cambridge, MA: MIT Press, 2007.

[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[34] N. Shroff, P. Turaga, and R. Chellappa, "Video precis: Highlighting diverse aspects of videos," *Multimedia, IEEE Transactions on*, vol. 12, no. 8, pp. 853–868, 2010.

[35] Y. Pang, Y. Yuan, and X. Li, "Gabor-based region covariance matrices for face recognition," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 7, pp. 989–993, 2008.

[36] A. Rosenfeld and G. Vanderburg, "Coarse-fine template matching," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 2, pp. 104–107, 1977.

[37] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 10, pp. 1042–1052, 1993.

[38] R. Maree, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 34–40 vol. 1, 2005.

[39] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proceedings of the 9th European conference on Computer Vision - Volume Part II*, ECCV'06, (Berlin, Heidelberg), pp. 589–600, Springer-Verlag, 2006.

[40] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, 2007.

[41] M. Rahman and M. Bhuiyan, "Face recognition using gabor filters," in *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pp. 510–515, 2008.

[42] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, pp. 1160–1169, Jul 1985.

[43] J. Buhmann, J. Lange, and C. von der Malsburg, "Distortion invariant object recognition by matching hierarchically labeled graphs," in *Neural Networks, 1989. IJCNN., International Joint Conference on*, pp. 155–159 vol.1, 1989.

[44] R. R. Curtin, J. R. Cline, N. P. Slagle, W. B. March, P. Ram, N. A. Mehta, and A. G. Gray, "MLPACK: A scalable C++ machine learning library," *Journal of Machine Learning Research*, vol. 14, pp. 801–805, 2013.

[45] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale l1-Regularized Least Squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2008.