Enhancement and Extensions of Principal Component Analysis for Face Recognition

by

Ana-Maria Sevcenco
B.E., University Politehnica of Bucharest, 2001
M.A.Sc., University of Victoria, 2007

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Enhancement and Extensions of Principal Component Analysis for Face Recognition

by

Ana-Maria Sevcenco
B.E., University Politehnica of Bucharest, 2001
M.A.Sc., University of Victoria, 2007

Supervisory Committee

Dr. Wu-Sheng Lu (Department of Electrical and Computer Engineering)

Supervisor

Dr. Pan Agathoklis (Department of Electrical and Computer Engineering)

Departmental Member

Dr. Hong-Chuan Yang (Department of Electrical and Computer Engineering)

Departmental Member

Dr. Julie Zhou (Department of Mathematics and Statistics)

Outside Member

Supervisory Committee

Dr. Wu-Sheng Lu (Department of Electrical and Computer Engineering)

Supervisor

Dr. Pan Agathoklis (Department of Electrical and Computer Engineering)

Departmental Member

Dr. Hong-Chuan Yang (Department of Electrical and Computer Engineering)

Departmental Member

Dr. Julie Zhou (Department of Mathematics and Statistics)

Outside Member

ABSTRACT

Primarily due to increasing security demands and potential commercial and law enforcement applications, automatic face recognition has been a subject of extensive study in the past several decades, and remains an active field of research as of today. As a result, numerous techniques and algorithms for face recognition have been developed, many of them proving effective in one way or another. Nevertheless, it has been realized that constructing good solutions for automatic face recognition remains to be a challenge.

The last two decades have witnessed significant progress in the development of new methods for automatic face recognition, some being effective and robust against pose, illumination and facial expression variations, while others being able to deal with large-scale data sets. On all accounts, the development of state-of-the-art face recognition systems has been recognized as one of the most successful applications of image analysis and understanding. Among others, the principal component analysis (PCA) developed in the early 1990s has been a popular unsupervised statistical method for data analysis, compression and visualization, and its application to face recognition problems has proven particularly successful. The importance of PCA consists in providing an efficient

data compression with reduced information loss, and efficient implementation using singular value decomposition (SVD) of the data matrix. Since its original proposal, many variations of the standard PCA algorithm have emerged.

This thesis is about enhancement and extensions of the standard PCA for face recognition. Our contributions are twofold. First, we develop a set of effective pre-processing techniques that can be employed prior to PCA in order to obtain improved recognition rate. Among these, a technique known as perfect histogram matching (PHM) is shown to perform very well. Other pre-processing methods we present in this thesis include an extended sparse PCA algorithm for dimensionality reduction, a wavelet-transform and total variation minimization technique for dealing with noisy test images, and an occlusion-resolving algorithm. Second, we propose an extended two-dimensional PCA method for face recognition. This method, especially when combined with a PHM pre-processing module, is found to provide superior performance in terms of both recognition rate and computational complexity.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1-D	One-dimensional
2-D	Two-dimensional
2DPCA	Two-dimensional principal component analysis
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
D-SRC	Downsampled sparse representation-based classification
DWT	Discrete wavelet transform
E-2DPCA	Extended two-dimensional principal component analysis
E-sparse	Extended-sparse
FN	False negative
FP	False positive
FPR	False positive rate
HE	Histogram equalization
HVS	Human visual system
ICA	Independent component analysis
IDCT	Inverse discrete cosine transform
IDFT	Inverse discrete Fourier transform
IDWT	Inverse discrete wavelet transform
LDA	Linear discriminant analysis
LLE	Locally linear embedding
MAD	Mean absolute deviation
MSE	Mean squared error
NP-hard	Non-deterministic polynomial-time hard
OCCL	Occlusion-resolving algorithm
PCA	Principal component analysis
PHM	Perfect histogram matching
PSNR	Peak signal-to-noise ratio
ROF	Rudin, Osher and Fatemi
R-SRC	Random sparse representation-based classification
SRC	Sparse representation-based classification
SVD	Singular value decomposition
TN	True negative
TP	True positive
TPR	True positive rate
TV	Total variation
WII	Wavelet illumination invariant
WPCA	Whitenedfaces principal component analysis
WT	Wavelet transform

Acknowledgments

First and foremost, I express my sincere gratitude to my supervisor, Dr. Wu-Sheng Lu, for his ongoing support and encouragement, and his continuous guidance in the fields of digital signal processing and optimization techniques. This thesis would never have been written the way it is without his generous help and support. Dr. Wu-Sheng Lu is the one who steered my research efforts to deal with up-to-date topics in the face recognition field, and came with many productive ideas as the work progressed. His energy, dedication, creativity and vast knowledge of so many different research topics, gave me invaluable guidance in completing this thesis. So here is a wholehearted THANK YOU, Dr. Lu!

I would like to thank Dr. Pan Agathoklis, Dr. Hong-Chuan Yang and Dr. Julie Zhou for their constructive ideas and suggestions for my work. Having them as committee members helped improve and enrich the content of this thesis. Many thanks also go to Dr. Jie Liang for serving as my external examiner.

It is a pleasure to express my gratitude towards all my professors I had at University of Victoria, during my MASc and PhD programs. They helped broaden my research horizons and gain a better understanding of the related areas of image processing.

I am also very grateful to the staff and faculty of the Department of Electrical and Computer Engineering who have provided assistance during my graduate studies. Thank you Vicky, Lynne and Moneca for your professional assistance, and warm and supportive advice.

I would also like to thank our good friends, Noreen, Diane, Jane and John, Carl and Joanne, Di and Jinhe, Carmen and Mihai, and new friends we made in the last three years, Lia and Cosmin, Barbara and Monte, Jie, Sahasi, for helping and encouraging us in difficult moments, making us laugh, cooking us delicious food, introducing us amazing places in or nearby Victoria, or just spending beautiful time with us.

My deepest gratitude goes to those people who are the most important in my life, my family, and especially my husband Sergiu, who is always by my side, giving me his unconditional love and support, and our daughter Victoria, the sunshine of our life.

Dedication

To my precious Victoria and Sergiu.

Chapter 1

Introduction

In this thesis we consider the problem of face recognition, and present enhanced and extended approaches in a principal component analysis framework. The purpose of this chapter is to introduce the problem addressed in the thesis, motivate the necessity for improved approaches, and describe the main contributions and organization of the thesis.

1.1 The Face Recognition Problem

The face is a primary focus of attention in social activities and plays a critical role in conveying identity and emotions [96]. Although the ability to speculate on character or intelligence from facial expressions remains suspicious, the human ability to recognize faces is astonishing. In fact, one can recognize a great number of faces throughout his/her lifetime and, even after years of separation, just at a glance one can identify almost instantly familiar faces that have undergone considerable changes due to aging and distractions like glasses and changes in hairstyle and facial hair, demonstrating the amazing robustness of the human visual system (HVS) [96]. It is therefore natural and desirable to develop computer-aided systems that mimics the HVS and can be used to automate the process of face recognition with satisfactory accuracy and improved speed. As a matter of fact, such development had started four decades ago, although the success of the system reported there was rather limited from today's standard. Extensive research has been conducted by psychophysicists, neuroscientists, and engineers on various

aspects of human and machine face recognition, such as whether face perception is a dedicated process [37], and whether it is done by global or local feature analysis [13]. Studies have shown that distinctive faces are better retained in memory and faster recognized than typical faces [11], [12]. The role of spatial frequency analysis was also examined. In [83] it has been observed that gender classification can be successfully accomplished using low-frequency components only, while identification requires the use of high frequency components. Some experiments suggest that memory for faces is highly viewpoint-dependent [45], and various lighting conditions make harder to identity even familiar faces [49]. In addition, based on neurophysiological studies [11], it seems that analysis of facial expressions is not directly related to face recognition. On the other hand, from a machine recognition point of view, dramatic changes in facial expressions may affect face recognition performance.

Speaking automatic identification systems, we remark that although several other reliable methods of biometric personal identification exist, for example methods based on fingerprint analysis and retinal or iris scans, the success of these methods depends critically on cooperation of the participant. On the other hand, automatic face recognition is often effective independent of the participant's cooperation [17], [103].

Primarily due to increasing security demands and potential commercial and law enforcement applications, automatic face recognition has been a subject of extensive study in the past several decades [17], [103], and remains to be an active filed of research as of today. As a result, numerous techniques and algorithms for face recognition have been developed, many of them proving effective in one way or another. Nevertheless, it has been realized that constructing good solutions to automatic face recognition remains to be a challenge. One of the main sources of difficulty has to do with variations in pose, illumination and expression that may occur across the images involved in a face recognition system. Another source of difficulty is related to possible large data scale, especially when one seeks a sparse representation of a facial image of interest in an overcomplete dictionary for robust face recognition in presence of measurement noise and face occlusion [97]. The last two decades have witnessed significant progress in developing new methodologies [3], [33], [52], [96], [100], some being effective and robust against pose, illumination and facial expression variations [36], [38], [42], [97],

while others being able to deal with large-scale data sets due to their superior ability to reduce data dimensionality [50].

On all accounts, the development of state-of-the-art face recognition systems has been recognized as one of the most successful applications of image analysis and understanding [103]. Among other things, the principal component analysis (PCA) developed in early 1990s [52], [96] has been a popular unsupervised (based on software analysis of images without human-interaction) statistical method for data analysis, compression and visualization, and its application to face recognition problems has proven particularly successful. Essentially, PCA finds the directions of maximized variance of a given set of data (e.g. a training set containing facial images), also known as principal components or eigenfaces in the context of a face recognition problem, and use them to represent an input signal (e.g. a test facial image) in reduced dimensionality. The importance of PCA lies in the fact that it provides a way to compress the data with reduced information loss and it can be carried out efficiently using singular value decomposition (SVD) of the data matrix. Since its original proposal, many variations of PCA that enhance or extend the standard PCA have emerged. Noticeable development in this regard includes the 2-D PCA [100] and sparse PCA [50].

Based on literature review of the field of face recognition and preliminary studies of several papers that have pioneered the field, the author was highly motivated and decided to concentrate her research on the methods of PCA with application to the face recognition problems. In short, this thesis is about enhancement and extensions of PCA for face recognition. Our contributions are twofold. First, we develop several pre-processing techniques that can be employed prior to the application of PCA in order to obtain improved recognition rate. Of these new techniques, we mention a technique known as perfect histogram matching (PHM) that is shown to perform very well. Other proposed pre-processing methods include an extended sparse PCA for dimensionality reduction, a wavelet-transform and total variation minimization technique for dealing with noisy test images, and an efficient occlusion-resolving algorithm. Second, we propose an extended 2-D PCA method for face recognition. This method, especially when combined with a PHM pre-processing module, is found to provide superior performance in terms of both recognition rate and computational complexity.

We now conclude this section with a note on defining the face recognition problem. A general face recognition problem can be formulated as follows. Given still or video images of a scene, identify or verify one or more individuals in the scene using a stored database of faces [103]. The solution to the problem involves face detection from cluttered scenes, feature extraction of the facial region, and recognition or verification. A simplified and more focused version of the face recognition problem starts with an input (or test) picture (image) and attempts to determine whether or not the picture is a record of a human face and, if it is, whether or not it matches to one of the individuals that have been included in a certain database. It is this version of the problem on that this thesis will be concentrated.

1.2 Contributions and Organization of the Thesis

1.2.1 Contributions of the Thesis

In this thesis, we investigate the problem of face recognition subject to varying illumination conditions and facial expression, and possible face occlusions and noise contamination. Our aim is at developing algorithms for face recognition with improved performance in terms of both recognition rate and computational complexity. The research mission is carried out through enhancement and extensions of the standard principal component analysis.

In summary, the main contributions of the thesis include:

- A general pre-processing technique based on histogram equalization that alters the spatial information of an image by perfectly matching its histogram to a desired histogram;
- A new image de-noising strategy that makes use of wavelet transform for a noise variance estimation and total variation minimization for noise reduction;
- A new face occlusion-resolving algorithm to deal with facial images that are partially occluded;
- An extended 2-D PCA algorithm with both row and column processing and a new classification criterion that demonstrates superior performance;
- An extended sparse PCA algorithm for improved dimensionality reduction;

- Proposal of a face recognition system that integrates the best techniques developed for superior system performance.

1.2.2 Organization of the Thesis

The rest of the thesis is divided into several chapters and an appendix. The main content of each chapter is outlined below.

Chapter 2 – Preliminaries

This chapter introduces some background material related to the basic concepts of face recognition problem in general, and methods, techniques and algorithms that are of direct relevance to the methods to be developed in the subsequent chapters of this thesis. These include discrete cosine and wavelet transforms, histogram of images, total-variation based methods for noise removal, principal component analysis, sparse representation of signals, and performance measures for face recognition.

Chapter 3 – Pre-Processing Methods

This chapter presents three pre-processing methods, namely the histogram-enhancing method, the de-noising technique and the occlusion-resolving algorithm, which are referred to as PHM, WT – TV and OCCL, respectively. We start with a brief overview of several pre-processing techniques that are usually encountered in face recognition methods and are most relevant to the proposed algorithms. These include whitenedfaces [61], discrete cosine transform [79], wavelet transform [38] and histogram equalization. Then, the three pre-processing methods for performance enhancement are described in detail. The performance of the proposed algorithms is evaluated and compared with the previously mentioned existing methods. The chapter concludes by addressing several implementation issues.

Chapter 4 – An Extended Two-Dimensional Principle Component Analysis Technique

This chapter presents an extended 2-D PCA algorithm. We first introduce some background information related to the 2DPCA algorithm in [100], then describe in detail the proposed technique, referred as E-2DPCA. The performance of the proposed

algorithm is evaluated and compared with the standard PCA and 2DPCA methods. Several implementation issues arising in the simulations are also addressed.

Chapter 5 – Face Recognition Using Sparse Representation

Here we present a preliminary dimensionality reduction technique based on the algorithm in [50]. The technique is an integral part of an extended sparse PCA algorithm, but it also can be regarded as a stand-alone pre-processing step. We start by providing some background material related to the concept of sparse representation of facial images [97] and sparse PCA for preliminary dimensionality reduction of large-scale data sets [50]. An extended sparse PCA (E-sparse PCA) algorithm is then developed and an E-sparse SRC algorithm that combines two of the studied algorithms is proposed for enhanced performance and efficient processing. Experimental results are presented to support the proposed techniques. Several implementation issues are also addressed.

Chapter 6 – Face Recognition Systems - Integrating the Proposed Techniques

In this chapter we compared two most promising face recognition techniques based on the PHM E-2DPCA and PHM E-sparse SRC algorithms. Simulation results are presented in search of the technique which provides the best performance in terms of recognition rate and elapsed computational time. Finally, a face recognition system integrating WT – TV, OCCL, PHM and E-2DPCA modules is proposed, and the chapter concludes by several general implementation issues.

Chapter 7 – Conclusions and Future Research

This chapter summarizes the main ideas and contributions of the thesis and suggests several directions for future research.

Chapter 2

Preliminaries

2.1 Introduction

The objective of this chapter is to provide background information about computer-aided face recognition in general and several specific techniques that are of particular relevance to the methods to be developed in the subsequent chapters of this thesis. These include discrete cosine and wavelet transforms, histogram of images, total-variation based methods for noise removal, principal component analysis, and sparse representation of signals. We also include a concise review of several performance measures that are applicable to face recognition problems.

2.2 Face Recognition

2.2.1 Introduction and Motivation

Although the digital image processing as an engineering field is built on a quite analytical foundation, the human intuition and analysis play a central role in choosing the adequate techniques that fit in different situations, and this choice is often made based on subjective and visual judgments [61]. Taking into account the similarities and differences between human visual system and electronic devices in terms of resolution and ability to

adapt to changes in illumination, many digital image applications have been developed since early 1920s.

Recognizing faces is one of the recent digital image processing and computer vision applications, and also one of the fundamental tasks of the human visual system (HVS). The astonishing and deceptively simple face recognition skill of humans is robust, despite large changes in the visual stimulus caused by viewing conditions, expressions, aging, and distractions, such as glasses or changes in hair style or facial hair. As a consequence, the mechanism of feature extraction and coding for recognition of faces by the HVS has fascinated scientists from various disciplines including psychophysics, psychology, computer vision and pattern recognition [79].

One imagines that a computer can be taught to recognize faces by using facial images as inputs. It turns out that this task is extraordinarily complicated [17]. In fact, the development of a general computational model for face recognition is quite difficult, because faces are complex visual stimuli and are quite distinct from sine-wave gratings, or other artificial stimuli used in human and computer vision research.

Any description of faces in terms of features is not simple. For instance, a face can have lighter or darker skin; larger or smaller eyes or mouth; and black, brown, or blonde hair. Other attributes refer to image formation, like illumination or viewpoint from which the face is seen. Therefore, face recognition is a high-level task, and there seem to be no perfect computational schemes.

Driven by growing application demands like authentication for banking and security system access, research in automatic face recognition has increased significantly over the past several decades. Fast, automatic, non-intrusive and non-intimidating, face recognition modules can be combined with other biometric options such as fingerprints and eye-iris recognition systems to improve the accuracy of recognition process. Unlike the other two biometrics that require the subject's action such as putting one's hand on a device, face recognition has the advantage of recognizing the subjects in a passive manner. However, a weak side of this technology is that to date it has not yet achieved the high accuracy rate that the other two can offer.

In 1966, the first attempt to construct a semi-automated face recognition human-computer system was made [9], [10]. The system was based on the extraction of the

coordinates of a set of features from the photographs, which were then used by the computer for recognition. Later, feature extraction and pattern classification techniques [39] were employed for face recognition purposes. In [35] and [101], a template matching approach was developed and improved, using automatic feature measurements and deformable templates which are parameterized models of face.

Early 1990s have witnessed the beginning of a new wave of developments for face recognition, with considerable research endeavors made for enhancing recognition performance. These include principal component analysis (PCA) [96], independent component analysis (ICA) [3], linear discriminant analysis (LDA) [33], and non-linear dimensionality reduction methods such as Laplacianfaces [44], isomaps [95], and several more recent approaches [46], [54], [56], [61], [72], [75], [78], [97] that strive to improve the recognition process by combining known basic techniques, employing new pre-processing modules and modifying existing steps.

2.2.2 Pre-Processing for Face Recognition

In what follows we present a brief overview of several mathematical concepts and algorithms which are encountered in pre-processing steps for face recognition. Throughout, an image may be regarded either as a 2-D discrete signal denoted by $x(m,n)$ for $m, n = 0, 1, \ldots, N-1$, or as a continuous signal $u(x,y)$ in the spatial domain with $u:\Omega \subset R^2 \to R$, for (x,y) in Ω.

A. Discrete Fourier Transform (DFT) and Filtering

The discrete Fourier transform (DFT) offers considerable flexibility in the design and implementation of filtering solutions for enhancement, restoration, compression, de-noising of digital images, and other applications of practical interest [40].

Let $\{x(m, n)$ for $m, n = 0, 1, \ldots, N-1\}$ be a digital image of size $N \times N$, where m and n are spatial variables. The two-dimensional DFT (2-D DFT) of x, denoted by $X(u,v)$, is given by

$$X(u,v) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) e^{-j2\pi(um/N + vn/N)} \tag{2.1}$$

for $u, v = 0,1,..., N-1$, which is a frequency domain description of signal x with u and v as frequency variables.

The 2-D inverse DFT (2-D IDFT) is defined by

$$x(m,n) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} X(u,v) e^{j2\pi(um/N + vn/N)} \tag{2.2}$$

for $m, n = 0,1,..., N-1$. The 2-D DFT and its inverse establish a one-to-one correspondence between a frequency-domain representation and a spatial domain representation of a 2-D signal [40].

The value of the transform at the origin of frequency domain (i.e. $X(0,0)$) is called the *DC coefficient of the DFT* and is equal with N^2 times the average value of the image $x(m, n)$.

Even if $x(m, n)$ is real, its DFT is in general complex. It is straightforward to verify that DFT of a real signal is conjugate symmetric about the origin, and is periodic.

The foundation of filtering in both spatial and frequency domains resides in the convolution theorem which may be written as [40]

$$x(m,n) * h(m,n) \Leftrightarrow X(u,v)H(u,v) \tag{2.3}$$

and, conversely,

$$x(m,n)h(m,n) \Leftrightarrow X(u,v) * H(u,v), \tag{2.4}$$

where the symbol "$*$" indicates convolution of the two functions. Expression (2.3) indicates that the DFT of the convolution of two spatial functions can be obtained by multiplying the DFT of the functions. Conversely, (2.4) states that the DFT of the product of two spatial functions gives the convolution of the DFT of the functions.

Filtering a digital image in the spatial domain consists of convolving the image $x(m,n)$ with a filter mask with finite impulse response $h(m, n)$. According to (2.3), one can obtain the same result in the frequency domain by multiplying the DFT of image $x(m, n)$, namely $X(u,v)$, by the DFT of the spatial filter $h(m, n)$, namely $H(u,v)$, also referred as the filter's transfer function.

B. Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) is widely used in many signal processing applications. It is a Fourier-related transform similar to the DFT, which converts the signal from spatial domain to frequency domain [40], [48]. One of the features, that

distinguishes DCT from DFT, is that it transforms real-valued signals to real-valued DCT coefficients, which give DCT the advantage of convenience in applications. However, it is the energy compaction capability that makes DCT a popular and useful transform.

For a two-dimensional signal such as a digital image of size $N \times N$, $\{x(m, n)$ for $m, n = 0, 1, \ldots, N-1\}$, the two-dimensional DCT (2-D DCT) of $\{x(m,n)\}$ is defined by [84]

$$C(i,k) = \frac{2\alpha(i)\alpha(k)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cos\left(\frac{(2m+1)i\pi}{2N}\right) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \qquad (2.5)$$

where $i, k = 0, 1, \ldots, N-1$, and $\alpha(\cdot)$ takes two possible values:

$$\alpha(k) = \begin{cases} 1/\sqrt{2} & \text{for } k = 0 \\ 1 & \text{for } 1 \le k \le N-1 \end{cases}$$

Perfect reconstruction of the original data can be obtained by using the 2-D inverse DCT (2-D IDCT) of $\{C(i,k)\}$ which is given by

$$x(m,n) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \alpha(i)\alpha(k)C(i,k) \cos\left(\frac{(2m+1)i\pi}{2N}\right) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \qquad (2.6)$$

where $m, n = 0, 1, \ldots, N-1$.

Among the DCT coefficients defined by (2.5), the one with $(i, k) = (0, 0)$ is called the *DC coefficient of the DCT* and is equal to the mean value of the image. The remaining $C(i, k)$ with $(i, k) \neq (0, 0)$ are called *AC coefficients*.

It can be verified that the 2-D DCT is an orthogonal, real, separable transform, and it possesses an energy compaction property [48], [79], which means that the most of the energy of the DCT coefficients is concentrated in a small number of coefficients typically corresponding to low frequencies. For reduced computational complexity, 2-D DCT is usually applied to 8×8 image blocks (this implies that the image must be divided into 8×8 blocks and 2-D DCT is computed with $N = 8$); however, there are algorithms which apply 2-D DCT to entire image (the reader is referred to Section 3.2.2 for details).

C. Discrete Wavelet Transform (DWT)

As applied to digital images, the discrete wavelet transform (DWT) is a powerful mathematical tool that leads to multiresolution analysis and synthesis of images. In addition to being an efficient framework for representation and storage of multiresolution images, the spatial-frequency analysis provided by DWT exploits an image‘s spatial and

frequency characteristics [40]. On comparison, the DFT reveals only the image's frequency attributes.

A DWT can be implemented efficiently through digital filter banks. As an example, a filter bank with one level of decomposition and reconstruction for 2-D discrete signals is illustrated in Figures 2.1 and 2.2.

In Figure 2.1, suppose the input signal $x(m,n)$ is an image of size $N \times N$, the one level 2-D DWT subband decomposition produces four subimages, LL, LH, HL and HH, each of size $N/2 \times N/2$, where LL is a low-resolution approximation of the input, while LH, HL and HH represent information about the image details along its horizontal, vertical and diagonal directions, respectively.

The building block of the analysis filter bank shown in Figure 2.1 can be used to construct an analysis filter bank with a tree structure up to K levels of decomposition with $K = \log_2 N$.

In the corresponding one-level subband reconstruction illustrated in Figure 2.2, the subband signals LL, LH, HL and HH produced by the analysis filter bank, or a processed version of these signals, are taken as input signals and used to reconstruct the image $x(m,n)$. The one-level synthesis filter bank in Figure 2.2 can be used as a building block to construct a synthesis filter bank with a mirror-image symmetric tree structure (with regard to that in the analysis filter bank) up to $K = \log_2 N$ levels to match a K-level analysis filter bank for image reconstruction.

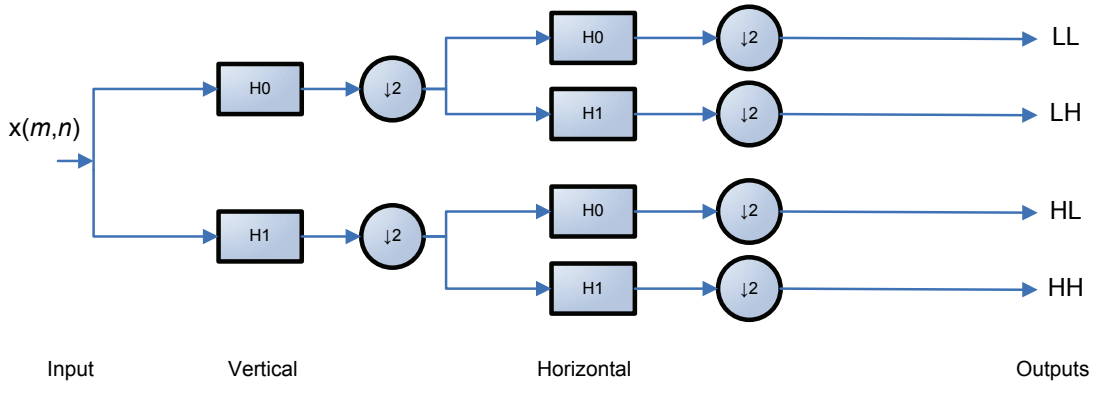


Figure 2.1. One-level 2-D wavelet decomposition.

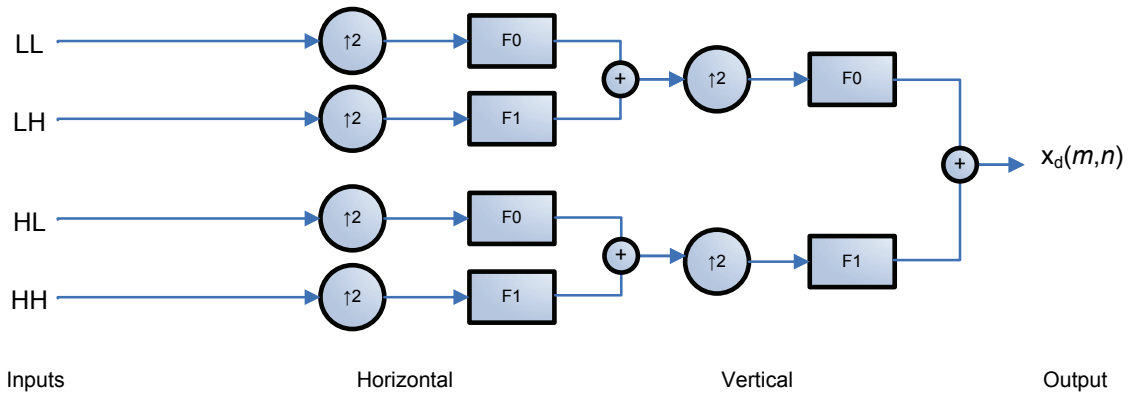


Figure 2.2. One-level 2-D wavelet reconstruction.

Perfect reconstruction of the original input signal is obtained if the low-pass and high-pass analysis filter H_0 and H_1 and corresponding synthesis filters F_0 and F_1 are orthogonal filters satisfying the perfect reconstruction conditions [40]

$$\begin{aligned} F_0(z)H_0(z)+F_1(z)H_1(z) &= 2z^{-l} \\ F_0(z)H_0(-z)+F_1(z)H_1(-z) &= 0 \end{aligned} \tag{2.7}$$

with l being the number of samples of delay.

D. Histogram

One effective method to deal with varying light conditions in images of a dataset is to apply a histogram-based pre-processing. The histogram of a digital image of size $N \times N$ with G gray levels is a discrete function that maps each k^{th} gray level to the number of pixels in the image characterized by that gray level. Analytically, the expression of the histogram is given by $h(r_k) = n_k$ with $k = 0,1,...,G-1$, where r_k denotes the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k. For an 8-bit digital image, for example, $G = 2^8 = 256$ and each r_k assumes one of the discrete values {0, 1, …, 255}. The relative frequency of a pixel having gray level r_k in the image is equal to $p_r(r_k) = n_k / n$, where $n = N^2$ represents the total number of pixels in the image. It follows that $p_r(r_k)$ is merely a normalized version of the histogram, satisfying

$$0 \le p_r(r_k) \le 1 \quad \text{and} \quad \sum_{k=0}^{G-1} p_r(r_k) = 1. \tag{2.8}$$

For this reason $p_r(r_k)$ is often referred to as the probability of occurrence of gray level r_k [40].

E. Total Variation (TV) Minimization Methods

Signal deterioration often occurs during signal acquisition, formation, transformation and recording. For images, the most frequently encountered forms of signal deterioration include noise contamination, defocusing and motion blur. Considering an image as a 2-D continuous function $u(x, y)$, a common image restoration model used as a general framework when dealing with the above degradation aspects [65] is given by

$$u_0(x, y) = (Hu)(x, y) + w(x, y) \qquad \text{for } (x, y) \text{ in } \Omega, \tag{2.9}$$

where $u : \Omega \subset R^2 \to R$ is the original image, u_0 is the observed image, which is a degraded version of u, w denotes white additive Gaussian noise with zero mean and variance σ^2, and H is typically a convolution type integral operator for modeling several common blurring processes such as averaging, Gaussian low-pass, Laplacian of Gaussian and motion blur.

Given an observed data u_0, the problem at hand is to find an estimate of the original image u based on model (2.9). An approximation of u can be identified by solving the least-squares problem [65]

$$\inf_u \iint_\Omega (Hu - u_0)^2 dxdy, \tag{2.10}$$

where the minimum must satisfy $H^*Hu - H^*u_0 = 0$, in which H^* denotes the adjoint of the operator H. To address this ill-posed problem, it is necessary to first regularize the functional in (2.10). One way to do this is to minimize the modified functional

$$F(u) = \iint_\Omega |\nabla u| dxdy + \frac{\lambda}{2} \iint_\Omega (Hu - u_0)^2 dxdy, \tag{2.11}$$

where the first term is a regularization term (called total variation of image u and defined as the magnitude of the gradient of image u, namely $J(u) = \iint_\Omega \sqrt{u_x^2 + u_y^2}\, dxdy$, with $u_x = \partial u / \partial x$ and $u_y = \partial u / \partial y$) that helps to reduce the noise, the second term is a fidelity

term to ensure that the solution u obtained by minimizing the functional in (2.11) is a close resemblance of u_0, and λ is a positive weight which balances the two terms.

In a variational optimization framework, Rudin, Osher and Fatemi (ROF) [81] investigated and formulated the de-noising problem where the model given by (2.9) becomes

$$u_0(x,y) = u(x,y) + w(x,y) \qquad \text{for } (x,y) \text{ in } \Omega, \tag{2.12}$$

and the problem is formulated as

$$\begin{aligned} &\text{minimize } J_0(u) = \iint_\Omega \sqrt{u_x^2 + u_y^2}\; dxdy \\ &\text{subject to: } \iint_\Omega u\; dxdy = \iint_\Omega u_0\; dxdy \\ &\qquad\qquad \iint_\Omega (u - u_0)^2\; dxdy = \sigma^2 . \end{aligned} \tag{2.13}$$

It can be shown [65] that the Euler-Lagrange equation for problem (2.13), which is the first order necessary condition for u to be a solution of (2.13), is given by

$$\begin{aligned} &\frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right) + \frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2+u_y^2}}\right) - \lambda(u-u_0) = 0 \\ &\qquad\qquad \left.\frac{\partial u(x,y)}{\partial N}\right|_{\partial\Omega} = 0 . \end{aligned} \tag{2.14}$$

In [81], problem (2.14) is solved by embedding it into a nonlinear parabolic equation with time t as an evolution parameter, namely,

$$\begin{aligned} &\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right) + \frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2+u_y^2}}\right) - \lambda(t)\cdot(u-u_0) \;\text{ for } t>0, (x,y)\in\Omega \\ &\qquad\qquad \left.\frac{\partial u(x,y)}{\partial N}\right|_{\partial\Omega} = 0 , \end{aligned} \tag{2.15}$$

where the Lagrange multiplier $\lambda(t)$ is updated using

$$\lambda(t) = -\frac{1}{\sigma^2}\iint_\Omega \left[\sqrt{u_x^2+u_y^2} - \left(\frac{(u_0)_x \cdot u_x}{\sqrt{u_x^2+u_y^2}} + \frac{(u_0)_y \cdot u_y}{\sqrt{u_x^2+u_y^2}}\right)\right] dxdy . \tag{2.16}$$

The peak signal-to-noise ratio (PSNR) is typically used to evaluate the de-noising performance of ROF algorithm applied to an $N \times N$ size noise-contaminated image u_0. A commonly employed definition of PSNR for gray-scale images is

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{2.17}$$

where the term MSE in (2.17) stands for mean squared error and is obtained as $MSE = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [u_0(i,j) - u(i,j)]^2$, with u being the de-noised image.

2.2.3 Methods for Face Recognition

Face recognition has been an active area of research in image processing and computer vision for more than two decades and is certainly one of the most successful applications of contemporary image analysis and understanding. The past two decades have witnessed sustained research endeavors that have led to methods and algorithms with improved face recognition capability. These include principal component analysis (PCA) [96], [100], independent component analysis (ICA) [3], linear discriminant analysis (LDA) [33], isomaps [95], locally linear embedding (LLE) [80], [82], Laplacianfaces [44], [72] based on Laplacian eigenmaps [4], [5], and whitenedfaces [61].

Despite of the emerging nonlinear mapping techniques which preserve the local structure of face images and provide dimensionality reduction [4], [5], [72], [78], [80], [82], research interest in PCA-based algorithms for face recognition remains strong. In [79], a method that combines the discrete cosine transform (DCT), PCA, and the characteristics of the human visual system (HVS) is proposed. In [100], face images are treated as matrices instead of vectors as in the original PCA algorithm and a corresponding image projection technique is used for face recognition. These methods are shown to offer better recognition rates with improved computational efficiency. In [61], the authors propose a whitening filter as a pre-processing step, while in [20] a down-sampling step is considered as pre-processing and, in PCA, the eigenfaces are computed directly as the eigenvectors of the covariance matrix. In [46], an image partition technique is combined with vertically centered PCA and whitened horizontally centered

PCA to obtain a novel hybrid approach with better recognition performance relative to the traditional PCA.

Sparse representations of signals have received a great deal of attentions in recent years. Typically the technique searches for the sparsest representation of a signal in terms of linear combination of atoms in an overcomplete dictionary [47]. Research has focused on three aspects: (1) methods for obtaining sparse representations. These include matching pursuit [67], orthogonal matching pursuit [76], and basis pursuit [18]; (2) methods for dictionary design. These include the K-SVD method [2]; (3) applications of sparse representation in various fields. These include signal separation, denoising, coding [31], [32], [59], [74], [91]. In [91], sparse representation is used for image separation, where an overcomplete dictionary is generated by combining multiple standard transforms, including curvelet transform, ridgelet transform and discrete cosine transform. In [59], application of the sparse representation to blind source separation is discussed and experimental results on EEG data analysis are demonstrated. In [74], a sparse image coding method with the wavelet transform is presented. In [31], sparse representation with an adaptive dictionary is shown to have state-of-the-art performance for image denoising. The widely used shrinkage method for image denoising is shown to be equivalent to the first iteration of basis pursuit that solves the sparse representation problem [32].

In the following sections, we outline two of the techniques employed for face recognition, namely the conventional PCA and the signal sparse representation, as they are the ones that are most closely related to the work reported in this thesis.

A. Principal Component Analysis (PCA)

The PCA [96] is an eigenface-based approach for face recognition that seeks to capture the variation in a collection of face images and uses this information to encode and compare images of individual faces. Over the years, the conventional PCA initiated in [96] has inspired a great deal of research interest in the field that in turn has led to a number of new PCA-based methods and algorithms with improved performance.

The eigenfaces are defined as the eigenvectors of the covariance matrix of the set containing all face images, where each image is treated as a point in a high dimensional space. Eigenfaces extract relevant facial information, which may or may not match

human perception of face features such as eyes, nose, and lips, by capturing statistical variation between face images. Therefore, eigenfaces may be regarded as a set of features which offers a characterization of the global variation among the analyzed face images. Other advantages of using eigenfaces are efficient image representation using a small number of parameters and reduced computational and dimensional complexity [96], [103].

Given a data set $\mathcal{D}$, also called training set, consisting of M face images of K individuals, the PCA algorithm proposed by Turk and Pentland in 1991 [96] starts by transforming each $N \times N$ image in $\mathcal{D}$ into a column vector Γ_i of dimension N^2, by concatenating the image rows. The K individuals involved are called *classes*, each one having $L = M / K$ images in $\mathcal{D}$. Next, an average face Ψ is computed as $\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$, and subtracted from each vector Γ_i to construct vector Φ_i as $\Phi_i = \Gamma_i - \Psi$. The data matrix is then formed as $A = [\Phi_1 \ \ ... \ \ \Phi_M] / \sqrt{M}$ and the covariance matrix is constructed as $C = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T = AA^T$. Note that C is a matrix of large size $N^2 \times N^2$. Instead of directly computing the eigenvectors u_i and eigenvalues λ_i of matrix C, which usually is an intractable task for typical image sizes, the eigenvectors v_i and eigenvalues λ_i of a much reduced size $M \times M$ matrix $L = A^T A$ are computed, and the eigenvectors of matrix C are then found to be

$$u_i = \lambda_i^{-1/2} A v_i \quad \text{for } i = 1, ..., M . \tag{2.18}$$

These eigenvectors u_i, called eigenfaces, are used to represent the face images from $\mathcal{D}$, so as to examine an input image Γ (in the form of a column vector) as whether or not it is a face image and, if it is, whether or not it is a member of a class or a stranger (non-member).

A p-dimensional face space is generated by the span of the p most significant eigenvectors (i.e. eigenfaces) that are associated with the p largest eigenvalues of C, and the matrix composed of these p eigenfaces is denoted by $\tilde{U}$. The value of p can be determined based on the distribution of eigenvalues λ_i, or as a certain percentage of the

available number of eigenvectors u_i. Matrix $\tilde{U}$ is used to yield a p-dimensional pattern vector $\Omega = \tilde{U}^T \Phi$ where $\Phi = \Gamma - \Psi$, and is also used to project the input image onto the face space as $\Phi_f = \tilde{U}\tilde{U}^T \Phi = \tilde{U}\Omega$. The Euclidean distance d_0 between the input image Γ and the face space is computed as

$$d_0 = \left\| \Phi - \Phi_f \right\|_2 . \tag{2.19}$$

If distance d_0 is found to be below a chosen threshold δ_0, the input image Γ is classified as a face image, otherwise it is considered a non-face image.

Furthermore, if Γ turns out to be a face image, it can be classified as a class member or non-member face. And if it is a member then the class it belongs is identified. These are achieved by (i) evaluating $d_k = \left\| \Omega - \Omega_k \right\|_2$ for $k = 1, ..., K$ where the class pattern vector Ω_k is calculated as $\Omega_k = \frac{1}{L} \sum_{i=1}^{L} \Omega_k^{(i)}$ with $\Omega_k^{(i)} = \tilde{U} \Phi_k^{(i)}$ being the pattern vector of the i^{th} image of the k^{th} class; and (ii) comparing

$$d_{\min} = \min_k d_k \tag{2.20}$$

with a prescribed threshold δ_1. If $d_{\min} = \left\| \Omega - \Omega_{k^*} \right\|_2$ and $d_{\min} < \delta_1$, then the input image Γ is identified as a member of class k^*, else Γ is considered a non-member.

B. Sparse Representation

Sparse representation of digital signals has been a subject of intensive study in the past several years, and it has recently found applications for face recognition [47].

The problem of finding a sparse representation of a signal in an overcomplete dictionary can be formulated as follows. Given an $m \times n$ matrix A with $n > m$ (usually $n \gg m$) which contains the elements of an overcomplete dictionary in its columns, and a signal $y \in \mathbb{R}^m$, one seeks to find an $n \times 1$ coefficient vector x, such that $y = Ax$ and $\|x\|_0$ is minimized [47], [97], i.e., a vector that solves

$$\text{minimize } \|x\|_0 \qquad \text{subject to } Ax = y \tag{2.21}$$

where $\|x\|_0$ denotes the l_0-norm which counts the number of nonzero entries in vector x. Finding the solution to (2.21) is NP-hard [47] due to its combinational nature; that is, in

general no known procedure for finding the sparsest solution is significantly more efficient than exhaustively searching all subsets of the entries for x [97]. Suboptimal (sometimes optimal) solutions to problem in (2.21) can be found by an alternative problem where the objective function is replaced by the l_1-norm of x while the constraint $Ax = y$ remains unaltered [23], namely,

$$\text{minimize } \|x\|_1 \qquad \text{subject to } Ax = y \tag{2.22}$$

where $\|x\|_1$ denotes the l_1-norm which sums up the absolute values of the entries in vector x. Problem in (2.22) can be solved in polynomial time by standard linear programming methods [47].

2.3 Performance Measures for Face Recognition

In this section we familiarize the reader with the terminology employed to express and measure the performance of a face recognition algorithm.

2.3.1 True Positive Rate and False Positive Rate

The concept of true positive (TP) is equivalent with a *correct recognition* of the test face image as being a certain member from the testing set, while true negative (TN) corresponds to a *correct rejection*. False positive (FP), also known as type I error or error of the first kind or α error, means an *incorrect recognition* of the test face image as being a certain member from the testing set, while it is not. Finally, false negative (FN), also known as type II error or error of the second kind or β error, corresponds to the error of *failing to recognize* the test face image as being a certain member from the testing set while it truly is.

Using the above terminology, the true positive rate (TPR) is defined as the ratio between the number of TP and the total number of TP and FN

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.23}$$

while the false positive rate (FPR) is obtained in a similar way as

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{2.24}$$

These two measures are employed for class discrimination to be studied in Sections 3.7.1, which includes face/non-face and member/non-member classification. Figure 2.3 illustrates the four possible situations that may be encountered in a class discrimination procedure, where the generic *Class 1* and *Class 2* represent either the face and non-face classes, or the member and non-member classes, respectively. Typically, *Class 1* is constructed using the available images from training set, while *Class 2* may not be explicitly given, in which case any image not recognized as belonging to *Class 1* is assumed to be in *Class 2*.

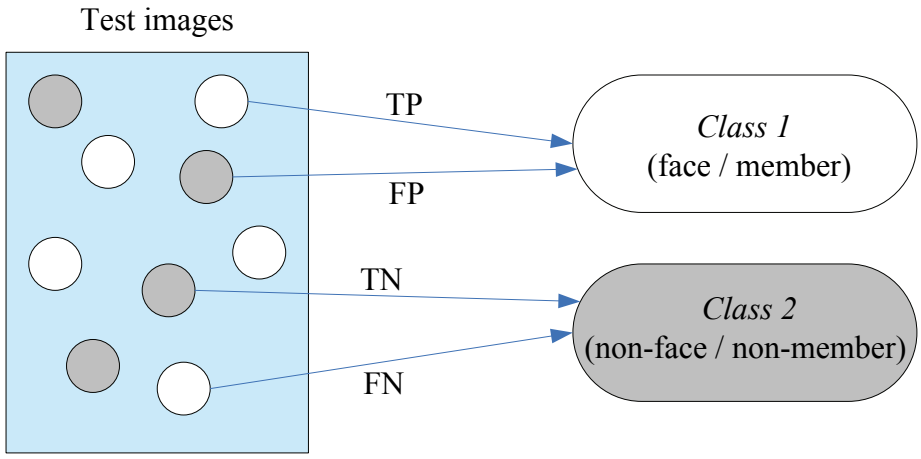


Figure 2.3. Example of TP, TN, FP,FN for class discrimination.

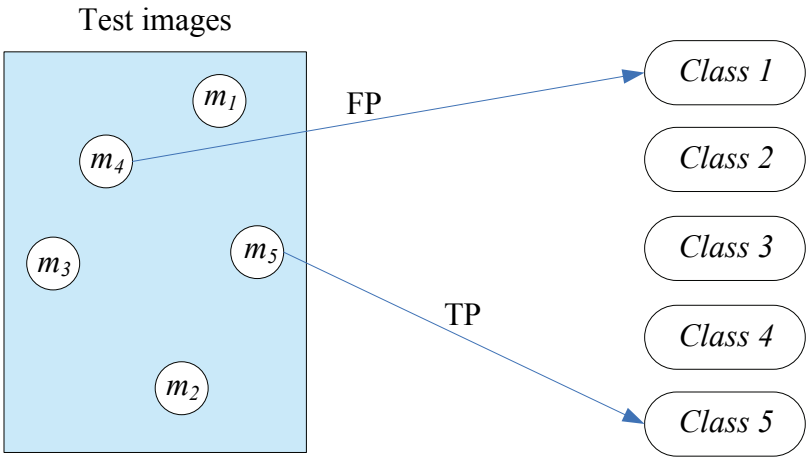


Figure 2.4. Example of TP, FP for face (member) identification.

2.3.2 Recognition Rate and Misdetection Rate

As illustrated in Figure 2.4, when dealing with face (member) identification, only the notions of TP and FP are utilized for evaluating algorithms' performance for face recognition. In this figure, *Classes 1*, *2*, …, *5* represent the classes containing images of individual 1, 2, …, 5, respectively, while m_1, m_2, …, m_5 denote the members of *Class 1*, *2*, …, *5*, respectively. Corresponding to the measure TPR in (2.23), the recognition rate can be defined as

$$\text{recognition rate} = \frac{\text{TP}}{\text{total number of test images}} \cdot 100 \ (\%) . \tag{2.25}$$

Similar to the measure FPR in (2.24), the misdetection rate can be calculated as

$$\text{misdetection rate} = \frac{\text{FP}}{\text{total number of test images}} \cdot 100 \ (\%) . \tag{2.26}$$

One may note that the two measures defined in (2.25) and (2.26) are complementary to each other in the sense that recognition rate + misdetection rate = 100 (%). As a result, for evaluating the performance of an algorithm, computing only one of them (e.g., the recognition rate) suffices.

2.4 Databases for Performance Evaluation of Face Recognition

Two databases have been employed to evaluate the performance of the face recognition algorithms. These are the Yale Face Database [112] and extended Yale Face Database B [106], [107]. These bases were chosen as they include face images under various illumination conditions and slightly different poses. In addition, they contain more frontal images per class (subject) than several other test data sets (such as FERET) and their images do not need to be rescaled.

2.4.1 Yale Face Database

The Yale Face Database contains a set of 165 grayscale images of 15 subjects (Figure 2.5), with 11 poses per subject (Figure 2.6), namely center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink, denoted as pose 'a', 'b', …, and 'k', respectively. The Yale Face Database images

Figure 2.5. The 15 individuals from the Yale Face Database.

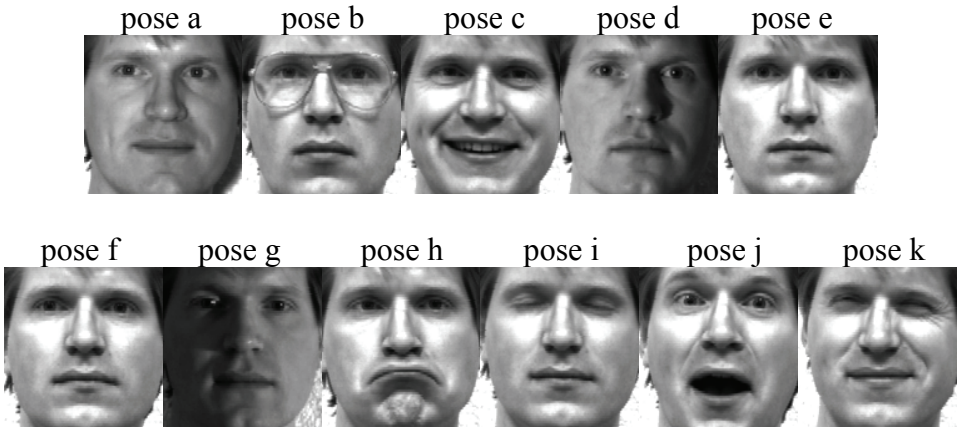


Figure 2.6. The 11 poses of one individual from the Yale Face Database.

employed in our simulations were cropped to 128×128 pixel size to minimize non-face areas such as hair and neck, with the image center approximately placed between the 2 nostrils of subject's nose, as illustrated in Figures 2.5 and 2.6.

2.4.2 Extended Yale Face Database B

The extended Yale Face Database B contains a total of 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. In our experiments, we used the cropped version of this database which contains images of individuals taken under 64

Figure 2.7. The 20 individuals selected from the extended Yale Face Database B.

different illumination conditions, with no pose variation, and each facial image was further manually aligned and cropped to size 168×168 pixels. To avoid working with some corrupted images in the cropped database, we selected only 20 individuals, with 64 images each, which are illustrated in Figures 2.7 and 2.8.

Simulations were performed using several datasets obtained from the above two face databases. Ten datasets (cases) covering a variety of facial expressions for testing images and various number of training images were created using Yale Face Database. Most of algorithms were tested on these ten cases, with an exception on sparse representation-based algorithms which require a very large number of training images. From the extended Yale Face Database B, six and eight cases were constructed, having various illumination conditions, ranging from "easy" light conditions which produce small shadows on individuals' faces, to "hard" light conditions which generate large shadowed regions. For the fourteen cases from the extended Yale Face Database B, the same training set was considered. These fourteen cases were utilized in the same context as the ten cases from Yale Face Database, as they do not include a very large number of training images.

Figure 2.8. The 64 images of one individual from the extended Yale Face Database B.

Especially for the sparse representation framework, four data sets containing numerous training and testing images were created based on images from the extended Yale Face Database B. These four data sets were also employed for the performance evaluation of 2-D PCA-based algorithms, as it will be seen in Chapter 6 of thesis.

For more details about the images included in the training and testing sets of the datasets utilized for algorithms' performance evaluation, the reader is referred to Appendix.

Throughout the thesis, the experimental results were presented in two forms: using charts or tables. Charts are an illustrative way of showing the results, easy to read and compare, and they have been employed for the large majority of simulation results. The only exceptions are found in Chapters 5 and 6 where the four data sets from the extended Yale Face Database B were utilized, or more information was necessary to be included for the completeness of the results. In these situations, we have decided to use tables as they appear to be a better choice for illustrating the experimental results.

2.5 Summary

In this chapter, we have provided background information about computer-aided face recognition and several specific techniques that are most relevant to the methods to be examined and developed in the thesis. Measures for evaluating performance of face recognition methods have been reviewed, and two databases of face images to be employed in our simulation studies have been described.

Chapter 3

Pre-Processing Methods

3.1 Introduction

Various face recognition techniques have been proposed in the past decades with a great deal of success. On the other hand, these methods often encounter difficulties when dealing with images captured under drastically different illumination conditions. Numerous tests have demonstrated that lighting variation is one of the bottlenecks in face recognition [38], [42]. Thus, there has been much work dealing with illumination variation for improved recognition rate. Generally, the algorithms in the literature can be classified into two categories: model-based and pre-processing-based. While the model-based approaches are theoretically ideal, when applied to real situations they require additional constraints and assumptions, and their computational cost increases accordingly, making them not very practical for real time systems. On the other hand, pre-processing-based techniques transform images directly without any assumptions or prior knowledge, being more commonly used in practical systems due to their simplicity and efficiency [27].

Section 2 of this chapter introduces background material related to the concept of pre-processing. We provide an overview of several techniques that are most relevant to this thesis. These include whitenedfaces, discrete cosine transform, wavelet transform and histogram equalization. In Section 3 we propose a method that modifies a given

image such that the histogram of the modified image perfectly matches a desired histogram. Section 4 illustrates how the proposed histogram-based method can be combined with a wavelet-based illumination invariant pre-processing for enhanced face recognition. A de-noising algorithm that employs wavelets and total variation minimization and takes into account the noise that contaminates a test face image is presented in Section 5. The performance of the pre-processing algorithms is evaluated, discussed and compared in Section 6, where several implementation issues are also addressed. The chapter concludes with Section 7, which summarizes the improvements offered by the proposed techniques over the existing methods.

The main contribution of this chapter is the proposal of a new pre-processing strategy based on perfect histogram matching (PHM) which can be incorporated into the standard PCA for face recognition. As will be demonstrated, this method yields a considerably more homogeneous tonal distribution in the face images involved in a PCA-based face recognition process. As a result, the recognition rate is improved while the required computational complexity is maintained at a low level. The possibility of combining PHM method with other pre-processing steps to achieve higher recognition rates is illustrated. Furthermore, a simple de-noising algorithm effective on additive white Gaussian noise-contaminated images is presented.

3.2 An Overview of Pre-Processing Methods

Human-face recognition is known to be a challenging task, especially because it has to deal with images of a subject that most of the times contain variations in illumination, pose, and expression. Several approaches to tackle the illumination issue in face recognition have been proposed. These include a low-dimensional representation of human faces for arbitrary lighting conditions [43], a post-processing method to accommodate lighting changes [6], a generalized *k*-linear reflection model to deal with illumination changes [88], and theoretical analysis of illumination in vision systems [7] and [102]. Considerable progress in face recognition under pose variations has also been made in the recent years. In [60] and [63], the authors propose to learn face dynamics from images with continuous pose variation. Recognition-by-synthesis approaches are

proposed in [57] where a test image with an arbitrary pose is transformed into frontal view and in [73] where each of the training images is transformed into the same pose as the test image. A probabilistic approach to face recognition is proposed in [51]. In [8] and [64], geometric information of human head is taken into account for enhanced recognition.

In what follows we briefly present several pre-processing methods which take variations in illumination into consideration and may be combined with a PCA-based face recognition process in order to improve the recognition rate.

3.2.1 Pre-Processing Using Whitenedfaces

As demonstrated by Liao et al. in [61], a pre-processing step of whitening and low-pass filtering that flattens the power spectrum of face images and controls the noise at high frequencies, can improve the recognition rate. A PCA method preceded by this type of processing is referred to as *whitenedfaces* recognition.

The motivation behind the whitening technique resides in the spectral behavior of natural scenes and facial images: their power spectra roughly fall with the increasing spatial frequency according to a power law $1/f^{\alpha}$. This unbalanced power spectra may result in potential problems when used in searching for structural information in an image space, as the information at low frequencies may swamp the equally useful information at high frequencies [61]. The solution in [61] is given basically by a combination of two filters: a whitening filter that attenuates low frequencies and boosts high frequencies yielding a roughly flat power spectrum across all spatial frequencies, and a low-pass filter that controls the noise at high frequencies. This combined filtering component may be integrated as a pre-processing step into a conventional PCA/ICA system.

In *whitenedfaces* recognition employing PCA (WPCA), a low-pass filter with frequency response

$$L(f) = \exp(-(f / f_c)^n) \tag{3.1}$$

Figure 3.1. The effect of whitenedfaces pre-processing: original image (left hand side) and its processed counterpart (right hand side).

is applied in order to avoid increasing the noise amplitude in image. The parameters are set to $f_c = 0.6 f_{\max}$ and $n = 5$, and the absolute spatial frequency is computed as $f = \sqrt{u^2 + v^2}$.

Subsequently, the whitening filter is applied to balance the power spectrum. Its frequency response assumes the following expression

$$W(f) = f^{\alpha_\omega / 2} \tag{3.2}$$

where the optimal value of whitening parameter α_ω is found to be 2.5. From (3.1) and (3.2), the whitening pre-processing is achieved by applying the combined filter as

$$W_L(f) = W(f) L(f) . \tag{3.3}$$

Multiplying the Fourier coefficients $F(u,v)$ of the face image $I(x,y)$ with the combined filter $W_L(f)$, the result in frequency domain is obtained as

$$F_w(u,v) = W(f) L(f) F(u,v) . \tag{3.4}$$

As a final step of the pre-processing module, the whitenedface image $I_w(x,y)$ is computed by taking the inverse Fourier transform of (3.4). As an example, Figure 3.1 shows the result of applying the whitenedface algorithm to one of the face images from extended Yale Face Database B.

3.2.2 Pre-Processing Using Discrete Cosine Transform (DCT)

In Ramasubramanian and Venkatesh [79], a method that combines the discrete cosine transform (DCT) and standard PCA while at the same time takes into consideration the characteristics of the human visual system (HVS) is proposed.

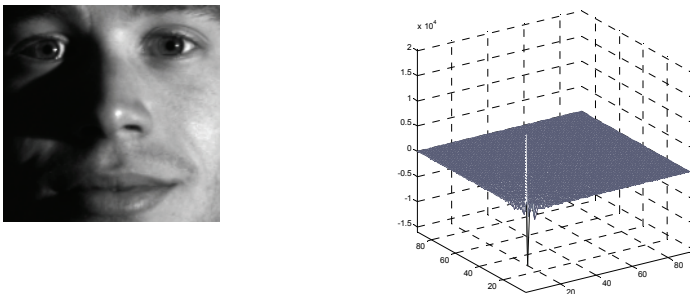


Figure 3.2. Applying 2-D DCT: original image (left hand side) and a 3-D representation of 2-D DCT coefficients (right hand side).

Two dimensional (2-D) DCT of the i^{th} face image in data set $\mathcal{D}$ (as denoted in Section 2.2.3) is performed. Mimicking the HVS that is known to be more sensitive to the low-frequency components of an image, the algorithm keeps only a certain number of low-frequency DCT coefficients (see Figure 3.2) arranged as a column vector F_i. In this way, a corresponding data set $\hat{D}$ in the frequency domain is obtained as $\hat{D} = \{F_1, F_2, ..., F_M\}$. By treating $\hat{D}$ as the training set just like in the standard PCA, a PCA-based algorithm can be applied for face recognition purpose. The orthonormal eigenvectors of the corresponding covariance matrix are called *cosine-faces* [79]. Because insignificant spectral components (i.e. the high-frequency DCT coefficients) have been discarded, DCT-PCA based algorithms can perform faster recognition with improved accuracy. In [79], it is observed that the recognition rate improves when 30% of DCT coefficients in the low frequency range are kept.

3.2.3 Pre-Processing Using A Wavelet Illumination Invariant (WII) Approach

In Goh et al. [38], a wavelet-based illumination invariant (WII) algorithm employing the illumination-reflectance model is proposed. An image $I(x, y)$ is regarded as the product of a reflection function $R(x, y)$ and a luminance function $L(x, y)$, i.e.,

$$I(x, y) = R(x, y)L(x, y), \tag{3.5}$$

Figure 3.3. The effect of WII pre-processing: original image (left hand side), its processed counterpart (middle) and the power spectrum of the processed counterpart (right hand side).

where the reflectance component contains information about the object of the scene and the luminance component contains only geometric properties of the scene. The illumination component L is basically a low-pass component, as the illumination does not present high spatial changes, while the reflectance component R represents the structure or edges of the image. Therefore, the assumption that function L varies slowly while function R can change abruptly is employed and used to extract function R by applying high-pass filtering to the logarithm of image. Taking logarithm of (3.5) transforms the product of illumination-reflection model into a sum with two components that are low-pass and high-pass respectively, and (3.5) becomes

$$\log I(x,y) = \log R(x,y) + \log L(x,y)\,. \tag{3.6}$$

Next, a multi-level discrete wavelet transform (DWT) is performed to separate these components. The more levels of decomposition are applied, the more the reduced-size approximation image concentrates on the low frequencies. To eliminate the illumination effects that may degrade the recognition rate, the wavelet coefficients of the reduced-size approximation in low-low subband are set to zero, yielding a decomposed and modified image function $S(x,y)$. Subsequently, the image recovery process is performed in two stages by taking the inverse DWT (IDWT) of $S(x,y)$ followed by computing $e^{T(x,y)}$ where $T(x,y)$ is the IDWT of $S(x,y)$. As an example, this technique is illustrated in Figure 3.3. The resulting image is used as an input in standard PCA process.

3.2.4 Pre-Processing Using Histogram Equalization (HE)

Histogram equalization (HE) (see Section 2.2.2 for the definition of histogram) is a pre-processing approach which exploits a simple, yet efficient image processing

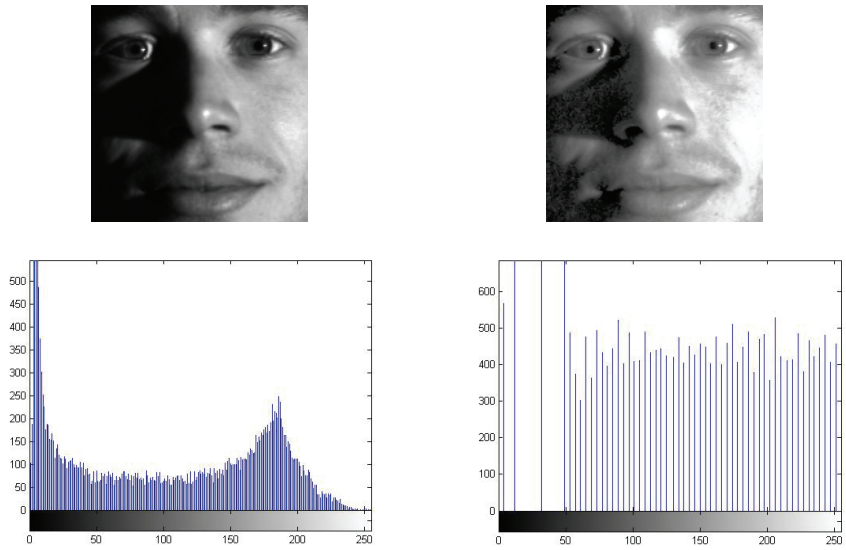


Figure 3.4. The effect of HE pre-processing: original image and its processed counterpart (top row), and their corresponding histograms (bottom row).

technique for contrast adjustment by using the histogram of the image. Ideally, it attempts to create an image with pixel intensities uniformly distributed over the entire intensity scale. In practice, HE effectively spreads out the most frequent intensity values, enhancing the contrast of the image. Figure 3.4 illustrates the effect of HE pre-processing when the MATLAB function "histeq" is employed.

The HE in conjunction with PCA-based algorithms has been employed in recent work [70], [89] and [98] with improved performance for face recognition. The authors in [98] propose a novel illumination compensation algorithm which uses block-based HE to compensate for illumination degrees that may be encountered in human face images. In [70], HE is applied as a photometric normalization technique, while in [89] it is utilized as an initial step for a novel biometric authentication approach using PCA, regularized-LDA and supervised neural networks.

3.3 A Pre-Processing Technique Based on Perfect Histogram Matching

In this section, we propose a new pre-processing method based on histogram matching that can be incorporated into the standard PCA for improved face recognition.

The objective of the histogram matching is to obtain a homogeneous tonal distribution for the face images in the data set $\mathcal{D}$ as well as for the input image Γ (see Section 2.2.3 for notation) by modifying the histogram of each image involved to match a desired histogram. In this way, the lighting conditions and light intensities across the entire image set tend to be equalized that in turn reduces the lighting-condition related discrepancy between $\mathcal{D}$ and Γ, leading to an improved recognition rate.

3.3.1 Desired Histogram

Histograms have been employed for numerous spatial-domain processing techniques for image enhancement, compression and segmentation, as they are straightforward to compute and allow efficient hardware implementations for real-time image processing applications [40]. It is well-known that images with either fairly dark characteristics or predominantly light tones can be enhanced by histogram equalization (HE). HE produces an image with nearly uniform distribution over the whole intensity scale, i.e. an image with a flat histogram. More generally, image histogram may be modified to match a particular histogram so as to highlight certain gray-level regions of the image [40]. On the other hand, however, current methods for histogram matching achieve its goal only approximately, see for example Section 4.2 of [40] and Section 7.3 of [48].

For a natural, well-balanced and homogeneous appearance across the face images in data set $\mathcal{D}$, the Gaussian function

$$h_d(r) = ae^{-\frac{(r-b)^2}{2c^2}}, \quad r \in [r_0, r_{G-1}] \tag{3.7}$$

is chosen to be the desired reference histogram [84], [85]. Parameter b represents the position of the center of the peak, c controls the width of the bell shape, a is the height of the curve's peak, and $[r_0, r_{G-1}]$ defines the interval where the Gaussian function specifies the desired histogram.

Because the gray levels of a digital image always assume integer values and the number of image pixels possessing a given gray level is also an integer, a discrete version of the Gaussian histogram assumes the form

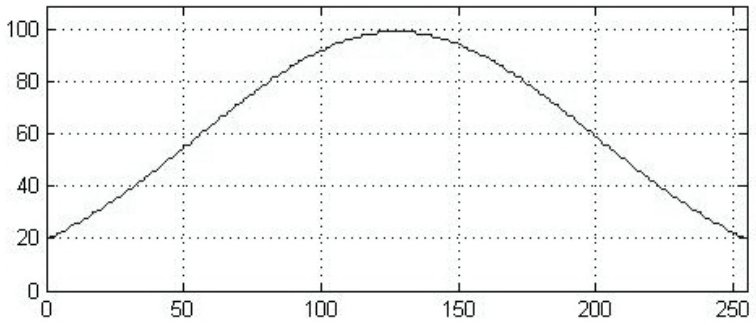


Figure 3.5. Gaussian shape of the imposed histogram.

$$h_d(r_k) = \text{round}[ae^{\frac{(r_k-b)^2}{2c^2}}], \qquad \text{for } k = 0,\ 1,\ ...,\ G-1 \tag{3.8}$$

and, for images of size $N \times N$, $h_d(r_k)$ must satisfy the constraint

$$\sum_{k=0}^{G-1} \text{round}[ae^{-\frac{(r_k-b)^2}{2c^2}}] = N^2 \tag{3.9}$$

which means that the total number of pixels in an image with the desired histogram remains to be N^2. For example, for an 8-bit image of size 128×128, we have $N = 128$, $G = 256$, $r_0 = 0$, and $r_{255} = 255$. For a smooth tonal distribution of gray levels, one needs to set the values of parameters *b* and *c* to provide midtones [14], [55] and a high image contrast [14], respectively. Under these circumstances, the left hand side term of (3.9) becomes a function of one single variable *a* which can be readily tuned to satisfy (3.9). For example, for $b = 127.5$, $c = 100$ and $N = 128$ (see Figure 3.5), the value of *a* satisfying (3.9) is found to be $a = 99.4568$.

3.3.2 Perfect Histogram Matching (PHM)

In what follows, we describe a technique that modifies the histogram of an image to precisely match a desired histogram [85].

Let the histogram of a digital image Γ of size $N \times N$ be given by $\{h(r_k) = n_k, k = 0,\ 1,\ ...,\ G-1\}$ and a desired histogram be given by $\{h_d(r_k) = n_k^{(d)}, k = 0,\ 1,\ ...,\ G-1\}$ having the same total number of pixels $n = N^2$ as the

original histogram $\{h(r_k)\}$. Viewing the image as a matrix $\Gamma = \{g_{ij},\ i, j = 1,\ 2,\ ...,\ N\}$, we define index set

$$I_k = \{(i, j) : g_{ij} = r_k\}. \tag{3.10}$$

Note that (i) the index set I_k contains the pixel locations in the image having gray level r_k; (ii) its length, $|I_k|$, is equal to n_k; and (iii) $\sum_{k=0}^{G-1} |I_k| = N^2 = n$. With these I_k defined, an ordered index-set sequence *I* can be constructed as

$$I = \{I_0, I_1, ..., I_{G-1}\}. \tag{3.11}$$

We remark that $I_l \cap I_k = \varnothing$ for $l \neq k$ and $\bigcup_{k=0}^{G-1} I_k$ covers the entire index set $\{(i, j),\ i, j = 1, 2, ..., N\}$.

Now if we write index set *I* explicitly as $I = \{(i_1, j_1),\ (i_2, j_2),\ ...,\ (i_n, j_n)\}$ and the desired histogram in terms of a sequence as $n^{(d)} = \{n_0^{(d)}, n_1^{(d)},\ ...,\ n_{G-1}^{(d)}\}$ which is associated with the gray-level sequence $r = \{r_0,\ r_1,\ ...,\ r_{G-1}\}$, then a natural way to modify the gray levels (hence histogram) of the given image to match $n^{(d)}$ is to assign the first gray level r_0 to the first $n_0^{(d)}$ pixels whose locations are specified by the first $n_0^{(d)}$ indices in sequence *I*. Next, one assigns the second gray level r_1 to the next $n_1^{(d)}$ pixels whose locations are specified by the next $n_1^{(d)}$ indices in *I*, and so on. The process continues until the last gray level r_{G-1} is assigned to the $n_{G-1}^{(d)}$ pixels whose locations are specified by the last $n_{G-1}^{(d)}$ indices in *I*. From the way the index sequence *I* is constructed and the histogram of the given image is modified, it follows that (i) the histogram of the given image so modified matches perfectly the desired histogram (see Figure 3.6); and (ii) subject to perfect histogram matching, the changes made in the histogram of the given image are minimized in the sense that the average difference between the original and modified gray levels at any given pixel location remains the smallest.

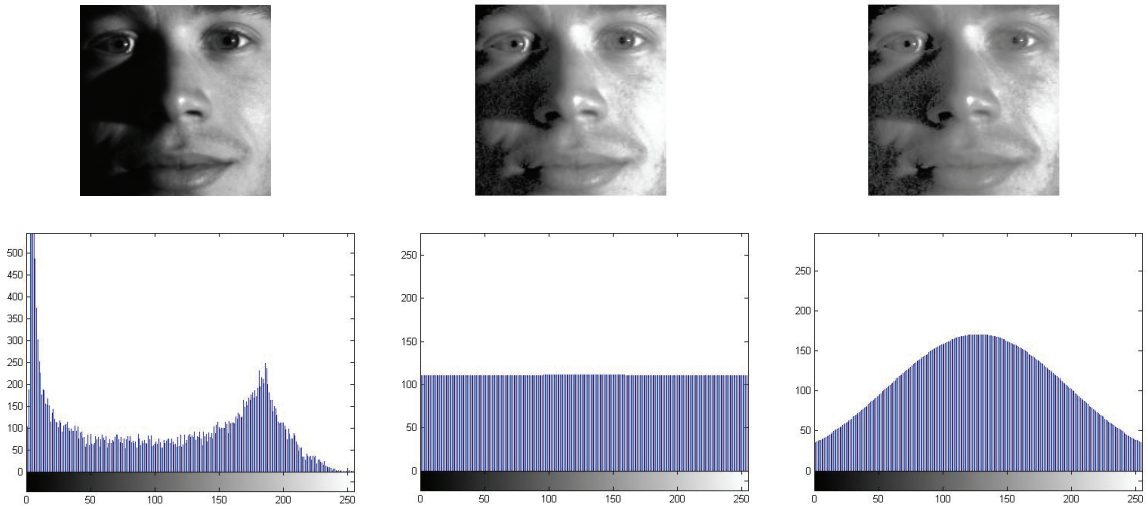


Figure 3.6. The effect of PHM pre-processing: original image and its processed counterparts using b = 127.5 and c = 2000 (for flat histogram) and 100, respectively, (top row) and their corresponding histograms (bottom row).

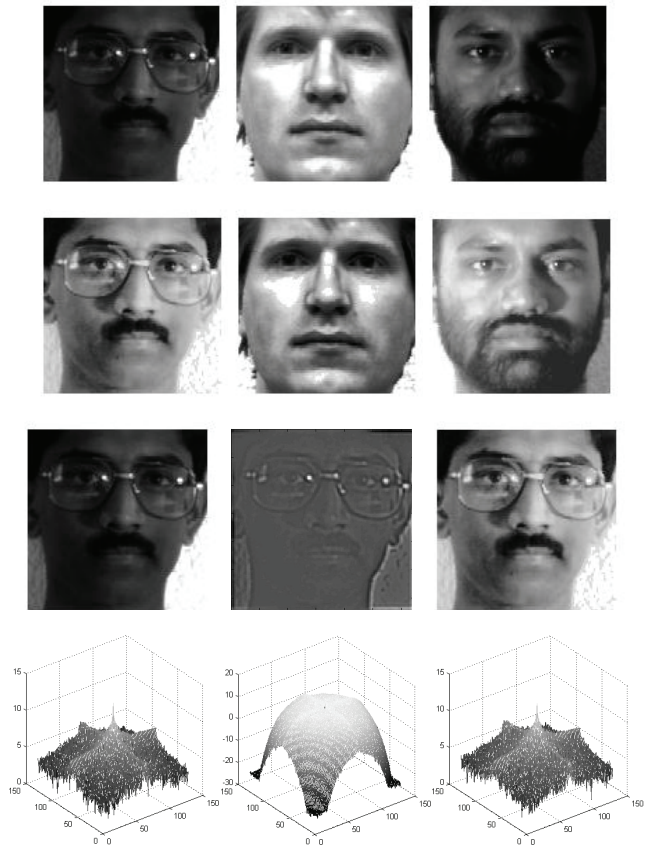


Figure 3.7. The effect of PHM and whitening pre-processing: three original images (top row) and their PHM-enhanced counterparts (second row); one original face image, its whitened version and its PHM-enhanced version (third row) and their corresponding power spectra (bottom row).

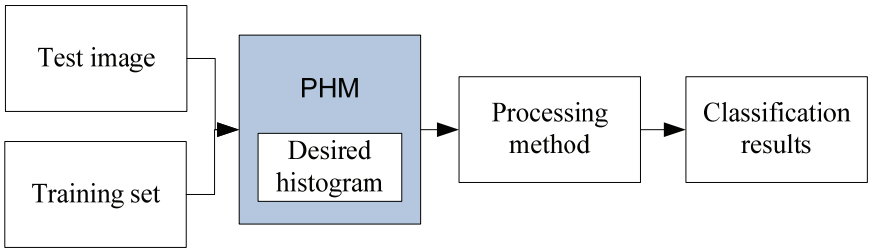


Figure 3.8. A block diagram of the proposed method incorporating PHM as a pre-processing module.

3.3.3 An Algorithm for PHM

The implementation of the proposed perfect histogram matching (PHM) pre-processing follows the outline [85]:

Begin: set *index_length* = 0;

For $k = 0 : 1 : G-1$, do:

- construct the working index set I_w of length $n_k^{(d)}$ as the subset of set *I*, which consists of the (*index_length* + 1)th element through the (*index_length* + $n_k^{(d)}$)th element in *I*;
- assign each of the pixels whose locations are specified by I_w to gray level r_k;
- set *index_length* := *index_length* + $n_k^{(d)}$ and $k := k + 1$;

End

As an example, the PHM algorithm was applied to three facial images of size 128×128 shown in the first row of Figure 3.7 with parameters set to $b = 127.5$ and $c = 100$. The pre-processing results are illustated in the second row of Figure 3.7.

For comparison purposes, the effect in spatial and frequency domains of the whitening filter using (3.3) and the histogram-based processing using (3.8) is shown in the third and forth rows of Figure 3.7 using one of the face images (for display purposes, the whitened face image was re-scaled into range [0, 255]).

The proposed method can be summarized by the block diagram in Figure 3.8. Central in the system, the PHM module is applied to the training set as well as the test image to obtain a homogeneous tonal distribution. This is followed by a processing step that yields feature descriptors which are subsequently used to represent and classify the face image. The classification consists of two phases: one in which the discrimination

between face/non-face and member/non-member images is attempted and another in which face identification for member images is performed.

3.4 A Combined PHM – WII Pre-Processing Technique for PCA

The two techniques described above, namely the PHM [85] and WII [38], can be combined into a pre-processing module for PCA. A block diagram of the combined pre-processing algorithm used in conjunction with standard PCA method for face recognition is shown in Figure 3.9.

As a first step, the PHM is applied to the training and testing sets to provide image sets with more homogenous illumination conditions. This is followed by a WII pre-processing module, described in Section 3.2.3, which mainly consists in decomposing the image into its frequency components and discarding the low frequency subband from the decomposed image. Subsequently, the output images are fed into a PCA algorithm to generate classification results. Figure 3.10 illustrates the result of applying this combined pre-processing module to a face image from the extended Yale Face Database B.

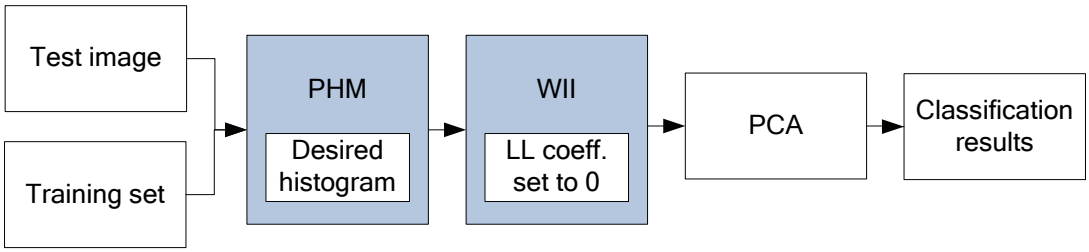


Figure 3.9. A block diagram of PHM – WII PCA algorithm.

Figure 3.10. The effect of PHM - WII pre-processing: original image and its processed counterparts after applying PHM and subsequently WII, respectively.

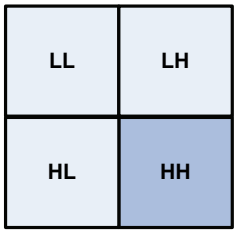


Figure 3.11. Image decomposition after one level of DWT.

3.5 De-Noising of Face Images by DWT and TV Minimization

Image de-noising has been a subject of extensive research in the past two decades. Although very promising de-noising results have been reported using methods such as wavelets, custom filtering, or anisotropic diffusion [77], the noise level in these algorithms is often assumed to be known and constant for varying brightness values.

In image de-noising literature, noise is often assumed to be additive white Gaussian. A commonly used noise estimation method is based on Donoho's wavelet shrinkage technique, the mean absolute deviation (MAD) of wavelet coefficients at the highest resolution [21], [24], [25]. In [92], the authors propose three methods to estimate the degree of noise corruption based on training samples and statistics of natural images, starting from MAD. In [62], the authors introduce a technique for estimating the value of an upper bound of the noise level in a single image based on a piecewise smooth model image.

Our approach to the de-noising problem consists of two steps, namely wavelet based noise estimation using MAD technique and total variation (TV) minimization for noise reduction.

3.5.1 Noise Variance Estimation Using Wavelets

A noise-variance estimation technique was introduced by Donoho [21] as an intermediate step from his de-noising algorithm. The estimation is evaluated from the information located in the subimage that keeps the image details, i.e. the diagonal component HH obtained in one-level of wavelet decomposition, see Figure 3.11. As illustrated in Section 2.2.2, the subimage HH is obtained after high-pass filtering and

down-sampling in both the vertical and horizontal directions. As a result, it contains high frequency components of the image and a substantial portion of the noise.

Based on [21] and [25] an estimate of the noise variance is given by:

$$\hat{\sigma}^2 = \frac{4}{N^2-4}\sum_{i=1}^{N/2}\sum_{j=1}^{N/2}\left[HH(i,j)-\overline{HH}\right]^2 \tag{3.12}$$

with $\overline{HH} = \frac{4}{N^2}\sum_{i=1}^{N/2}\sum_{j=1}^{N/2} HH(i,j)$ being the mean value of the diagonal component HH, and $N\times N$ being the size of the original image. This estimation is of importance in adjusting the weighting parameter λ for the de-noising algorithm based on TV minimization.

3.5.2 De-Noising Using TV Minimization

In this section, we consider the optimization problem (2.13) described in Section 2.2.2, with solution given in (2.15) and (2.16).

To speed up the algorithm we consider $\lambda > 0$ in (2.16) as a time invariant weighting parameter and tune its value depending on the noise variance estimation (3.12). In this perspective, (2.15) becomes:

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{u_x^2+u_y^2}}\right)+\frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{u_x^2+u_y^2}}\right)-\lambda\cdot(u-u_0) \text{ for } t>0,\ (x,y)\in\Omega \\ &\left.\frac{\partial u(x,y)}{\partial N}\right|_{\partial\Omega} = 0\ .\end{aligned} \tag{3.13}$$

An explicit difference scheme for solving (3.13) is given in [65]. With weighting parameter λ, time step-size Δt and number of iterations N as input parameters, one starts with notation

$$\frac{\partial u}{\partial t} \cong \frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t} \Rightarrow u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t\cdot\frac{\partial u}{\partial t}$$

$x_i = ih,\ y_j = jh,$ for $i,j = 0,1,...,K,$ with K being the number of grid points

$t_n = n\Delta t,$ for $n = 0,1,...$

$u_{i,j}^{n} = u\left(x_i, y_j, t_n\right)$

$u_{i,j}^{0} = u_0\left(x_i, y_j\right).$

and computes solution $u_{i,j}^{n+1}$ iteratively as follows

For $i, j = 0,1,...,K-1,$ do

$$u_{i,j}^{n+1} = u_{i,j}^{n}$$

$$+\Delta t\left[\delta_x^-\left(\frac{\delta_x^+ u_{i,j}^n}{\left[\left(\delta_x^+ u_{i,j}^n\right)^2+\left(m\left(\delta_y^+ u_{i,j}^n,\delta_y^- u_{i,j}^n\right)\right)^2\right]^{1/2}}\right)\right]$$

$$+\Delta t\left[\delta_y^-\left(\frac{\delta_y^+ u_{i,j}^n}{\left[\left(\delta_y^+ u_{i,j}^n\right)^2+\left(m\left(\delta_x^+ u_{i,j}^n,\delta_x^- u_{i,j}^n\right)\right)^2\right]^{1/2}}\right)\right]$$

$$-\Delta t\cdot\lambda\cdot\left(u_{i,j}^n - u_0\left(ih, jh\right)\right)$$

with

$u_{0,j}^n = u_{1,j}^n, u_{K,j}^n = u_{K-1}^n, u_{i,0}^n = u_{i,1}^n,$ and $u_{i,K}^n = u_{i,K-1}^n$

$$\delta_x^{\mp} u_{i,j} = \mp\left(u_{i\mp1,j} - u_{i,j}\right)/h$$

$$\delta_y^{\mp} u_{i,j} = \mp\left(u_{i,j\mp1} - u_{i,j}\right)/h$$

$$m\left(a,b\right) = \min \mod\left(a,b\right) \equiv \left(\frac{\operatorname{sgn} a + \operatorname{sgn} b}{2}\right)\min\left(\left|a\right|,\left|b\right|\right) .$$

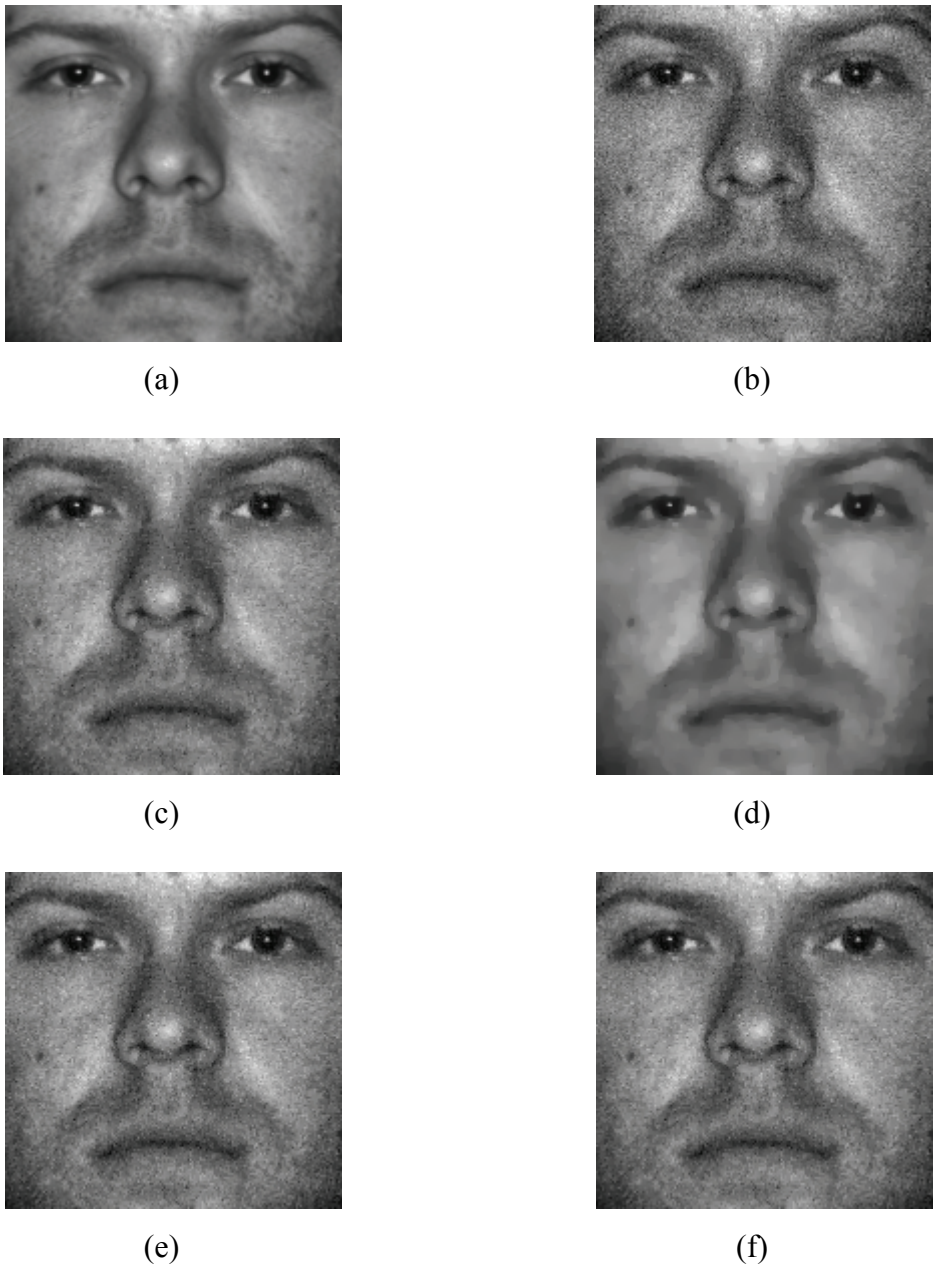
3.5.3 Tuning the De-Noising Parameters

There are several input parameters which need to be properly adjusted in order to achieve a satisfactory de-noising performance.

- The parameter of critical importance is λ. As we can see from (2.11), a small value of λ should be used to emphasize the regularization term when we deal with a severely contaminated image, where the value of the noise variance estimation $\hat{\sigma}^2$ computed via (3.12) is large. Similarly, for an image with less noise contamination, the value $\hat{\sigma}^2$ is small and a relatively large λ should be chosen to emphasize the fidelity term. Experimentally, it was found that λ should be chosen within the range of [0.005, 0.5], depending on the statistical features of the white additive Gaussian noise

w. However, selecting a value too low for parameter λ is not recommended, since this may yield image blurring due to an intensive smoothing action.

- A reasonable choice for the time step size Δt appears to be in the range of $[0.1,\ 0.3]$. The value of $\Delta t = 0.25$ was chosen in all our simulations.



(a) (b)

(c) (d)

(e) (f)

Figure 3.12. TV de-noising with $\Delta t = 0.25$ for an original image (a), noise-contaminated with noise amount of 6% (b), using $\lambda = 0.005$ and $N = 10$ (c), $\lambda = 0.005$ and $N = 50$ (d), $\lambda = 0.5$ and $N = 10$ (e), and $\lambda = 0.5$ and $N = 50$ (f).

- The algorithm runs a given number of iterations, *N*. An *N* too large leads to a very smooth image of low quality. An optimal choice for the number of iterations is a number less than 100.

Figure 3.12 illustrates the effect of varying parameters λ and *N* when the TV-based de-noising algorithm with parameter Δt set to 0.25 was applied to a noise contaminated image, where the maximum amount of noise added to the original image was set to 6% of the maximum image light intensity.

Table 3.1. PSNR results for three sets of input parameters in TV de-noising step

Noisy image		WT	TV		
			$\lambda = 0.5,\ N = 10$	$\lambda(\sigma^2),\ N(\sigma^2)$	$\lambda = 0.005,\ N = 50$
Noise amount (%)	$PSNR_{before}$	$\hat{\sigma}^2$	$PSNR_{after}$	$PSNR_{after}$	$PSNR_{after}$
0.4	48.11	0.0000	44.03	44.03	33.73
0.8	42.07	0.0000	42.57	42.57	33.71
1.2	38.53	0.0001	40.84	40.84	33.73
1.6	36.09	0.0001	39.10	39.10	33.69
2	34.15	0.0002	37.37	37.37	33.63
2.4	32.60	0.0002	35.83	35.83	33.67
2.8	31.23	0.0003	34.35	35.76	33.65
3.2	30.07	0.0004	32.99	35.28	33.66
3.6	29.10	0.0005	31.82	34.57	33.56
4	28.08	0.0007	30.60	34.31	33.55
4.4	27.33	0.0008	29.70	34.16	33.42
4.8	26.57	0.0009	28.78	33.84	33.33
5.2	25.86	0.0011	27.92	33.46	33.17
5.6	25.23	0.0013	27.17	33.25	33.12
6	24.60	0.0015	26.43	32.94	32.94
6.4	24.00	0.0017	25.71	32.71	32.71
6.8	23.49	0.0019	25.12	32.52	32.52
7.2	23.01	0.0021	24.56	32.18	32.18
7.6	22.52	0.0025	23.98	31.94	31.94
8	22.10	0.0027	23.50	31.58	31.58
8.4	21.68	0.0030	23.01	31.18	31.18
8.8	21.33	0.0032	22.62	30.95	30.95
9.2	20.90	0.0036	22.13	30.48	30.48
9.6	20.51	0.0038	21.68	29.96	29.96
10	20.22	0.0041	21.36	29.53	29.53

In our simulations, a set of 20 images from extended Yale face database B were chosen for determining appropriate values for parameters λ and N, with one pose per each individual. The step performing noise variance estimation using WT with Daubechies filters of length 6 as described in Section 3.5.1 and the step performing TV de-noising as described in Section 3.5.2 were carried out using the same set of images. Three sets of parameters shown in Table 3.1 were chosen for the TV de-noising. Zero-mean additive white Gaussian noise with various variance values was independently generated from image to image (see Section 3.7.5 for more details). The noise variance estimation $\hat{\sigma}^2$ was calculated using the normalized image pixel intensities (with values between 0 and 1), and the PSNR values before ($PSNR_{before}$) and after ($PSNR_{after}$) applying the TV de-noising algorithm with three different sets of parameters were recorded. The PSNR values averaged over the 20 images are shown in Table 3.1.

Assuming that a higher $PSNR_{after}$ value leads to higher recognition rates, one may adjust the parameters λ and N based on estimated noise variance $\hat{\sigma}^2$, as indicated in the following decision block:

$$\begin{cases} \text{if} \quad \hat{\sigma}^2 < 0.0003, \text{ then } \lambda = 0.5 \text{ and } N = 10 \\ \text{if} \quad 0.0003 \le \hat{\sigma}^2 < 0.0006, \text{ then } \lambda = 0.25 \text{ and } N = 30 \\ \text{if} \quad 0.0006 \le \hat{\sigma}^2 < 0.0009, \text{ then } \lambda = 0.1 \text{ and } N = 40 \\ \text{if} \quad 0.0009 \le \hat{\sigma}^2 < 0.0012, \text{ then } \lambda = 0.05 \text{ and } N = 45 \\ \text{if} \quad 0.0012 \le \hat{\sigma}^2 < 0.0015, \text{ then } \lambda = 0.02 \text{ and } N = 48 \\ \text{if} \quad \hat{\sigma}^2 \ge 0.0015, \text{ then } \lambda = 0.005 \text{ and } \mathrm{N} = 50 \end{cases} \tag{3.14}$$

Parameters λ and N were chosen as piecewise constant functions approximating the behavior of an exponential decay and recovery, respectively, as illustrated in Figure 3.13. This choice provided satisfactory results for $\hat{\sigma}^2$ in range [0, 0.0041] as shown in Table 3.1.

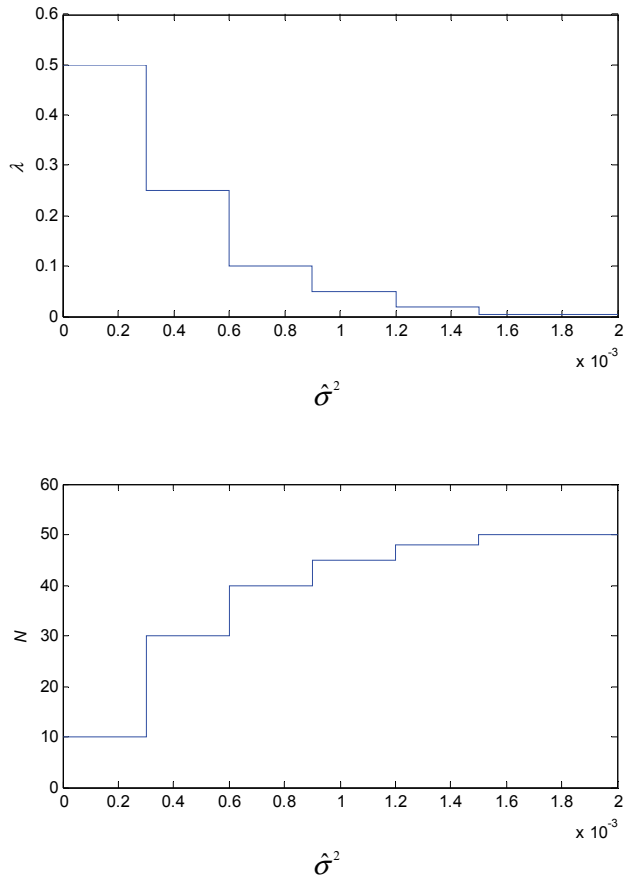


Figure 3.13. Piecewise constant functions $\lambda(\hat{\sigma}^2)$ - top, and $N(\hat{\sigma}^2)$ - bottom.

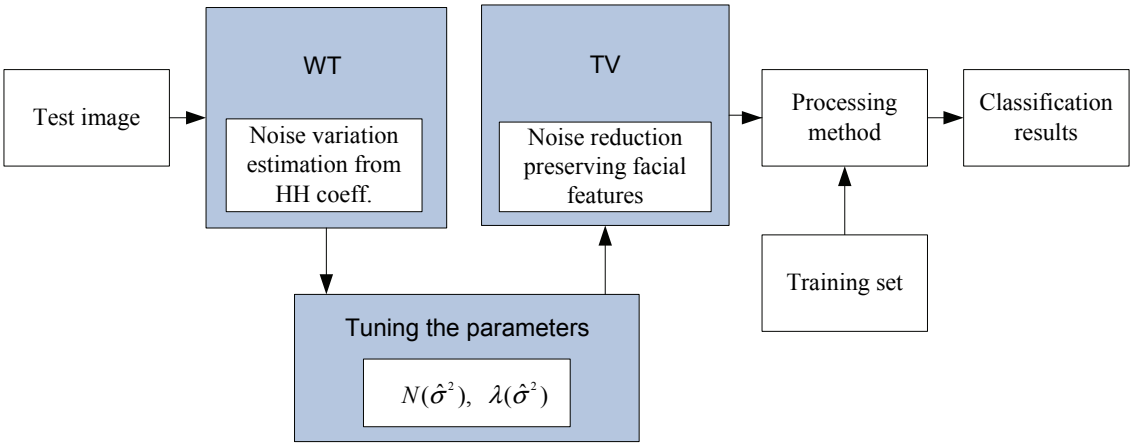


Figure 3.14. A block diagram of WT – TV pre-processing module.

To summarize the proposed de-noising procedure, Figure 3.14 illustrates a block diagram of WT – TV de-noising algorithm presented above. As a first step, WT is applied to the testing set to estimate the noise variance of each test image. Based on this, the input parameters for TV algorithm are adjusted and the de-noising process is accomplished. Subsequently, a face recognition technique is applied to the training set and de-noised test image, and the classification results are acquired.

3.6 Dealing with Face Occlusions

Partial face occlusion introduces a challenging problem in face recognition. Face recognition systems often confront occluded faces in real world applications, as the systems need to deal with people wearing common accessories, such as scarves or sunglasses, covering their face with their hands, or carrying different objects which may obstruct their face. Face occlusion may also occur when external objects partially occlude the camera view [29]. Therefore, a face recognition system has to be robust to occlusion in order to guarantee reliable results.

Several studies have been conducted in order to address this problem. In [68], face images are analyzed locally in order to handle partial face occlusion, where the face image is divided into local regions and for each region an eigenspace is constructed. In this way, if a region is occluded, it will automatically be detected. Moreover, weighting of local regions are also proposed in order to provide robustness against expression variations. A similar approach is presented in [94], where a self-organizing map is used to model the subspace instead of Gaussians or mixtures of Gaussians as in [68]. In [34], robustness against occlusion is provided by combining subspace methods that aim at best reconstruction (such as principal component analysis), with subspace methods that aim at discrimination (such as linear discriminant analysis). In [36] a generative model based approach is proposed, where the model assumes that image pixels are generated by an inlier process which is responsible for producing the majority of the data, or an outlier process which generates pixels not adhering to the inlier model. Partitioning into inlier and outlier regions is made by employing a hidden binary map which is modeled as a Markov random field, and inference is made tractable by a mean field EM-algorithm. A

very recent approach, different from those mentioned above, consists of utilizing sparse signal representation to analyze partially occluded face images [97].

Below we present a simple yet effective approach to deal with two types of facial occlusions which may be encountered in a face recognition process, namely the eyes occlusions and chin occlusions.

Given a test image, the potential occlusion problem is handled in the pre-processing step. First the test image is examined to see if occlusion of eyes or chin (i.e. if the person wears sunglasses or has beard/scarf, respectively) exists. This is done by applying a binary mask to the original test image to select the regions of interest r_1 and r_2, as illustrated in Figure 3.13. Subsequently, a decision is made based on the mean values of regions of interest. Let thr_1 be the threshold light-intensity value below which the regions of interest are considered occluded. In addition, let thr_2 be the maximum allowed mean difference between the two regions of interest. As the two mean values mean(r_1) and mean(r_2) should be similar when occlusion is present, occlusion is claimed only when mean(r_1) $< thr_1$, mean(r_2) $< thr_1$ and |mean(r_1) - mean(r_2)| $< thr_2$. If there is an eye or chin occlusion, then a binary occlusion mask is artificially created and applied to the test image and entire training set. In this way, negative effect of occlusion on recognition performance is largely eliminated, and equal conditions are created among the training set and test image which are subsequently fed into processing face recognition module and classification results are attained. For eye occlusion, an occlusion mask covers approximately one quarter of original image, while for chin occlusion it covers roughly one third of it.

A block diagram of this occlusion-resolving algorithm, shortly referred as OCCL, is given in Figure 3.15.

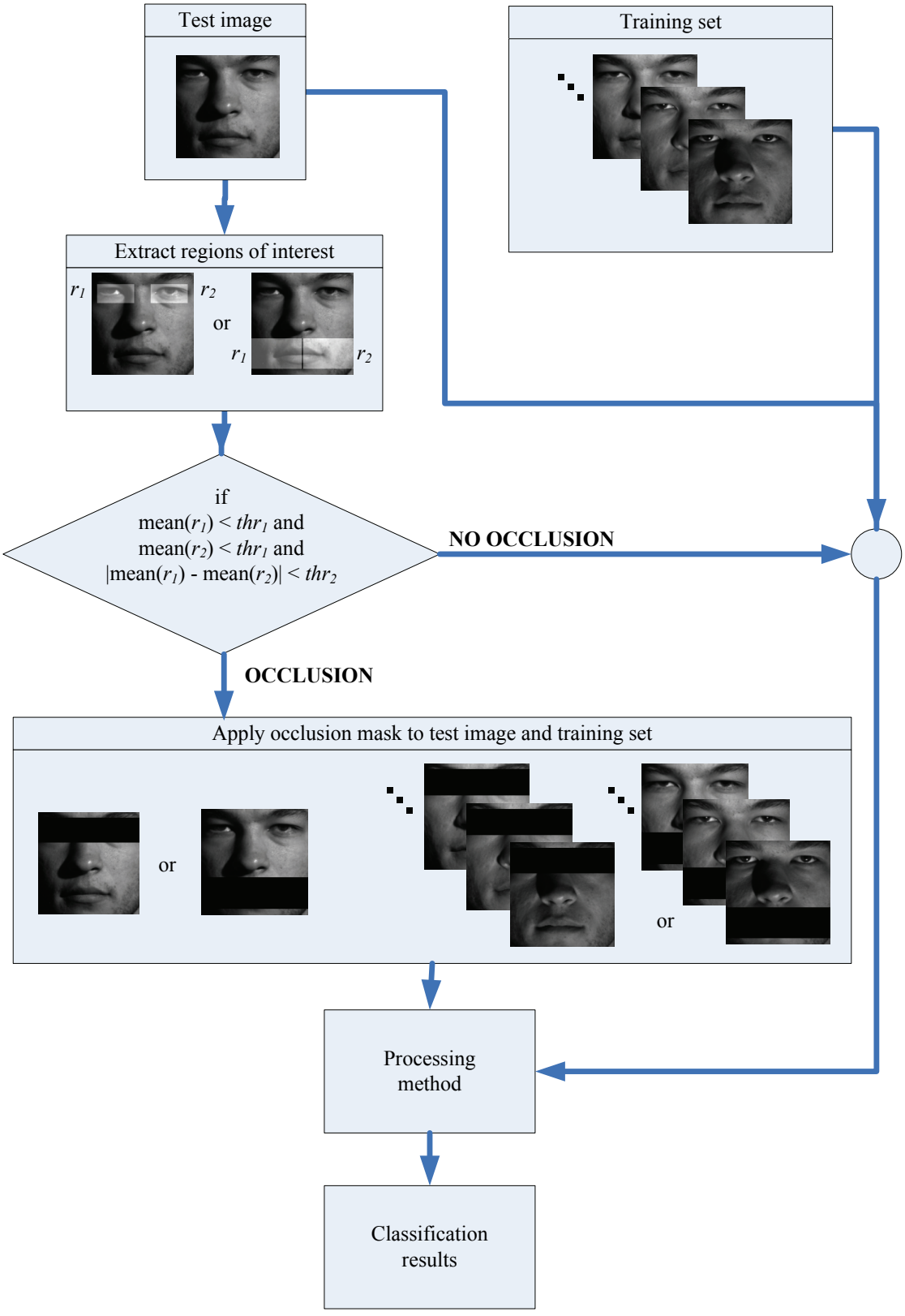


Figure 3.15. A block diagram of OCCL algorithm.

3.7 Experimental Results

As a general pre-processing algorithm, PHM may be incorporated into practically any face identification method to enhance its robustness to various facial expressions and illumination-related discrepancies. In this section, this point is illustrated by considering a case study where the PHM algorithm is applied to both the input image Γ and data set $\mathcal{D}$ prior the application of the conventional PCA algorithm. Throughout, the proposed method will be referred to as the perfect histogram matching PCA (PHM-PCA). Our study aims to evaluate the performance of PHM-PCA and compare it with WPCA [61], DCT-PCA [79], WII-PCA [38] and PCA [96] algorithms. The proposed algorithm is also compared with a PCA-based algorithm which incorporates a HE pre-processing step. The Yale Face Database and extended Yale Face Database B have been chosen as image databases as they include more frontal images per class (subject) than several other test data sets (such as FERET) and the images do not need to be rescaled.

3.7.1 Results for the Yale Face Database

A. Choosing Parameters for Gaussian Histogram

In all simulations, the Gausssian histogram in (3.8) was utilized as the reference histogram where the parameters were set to be $G = 256$, $r_0 = 0$, and $r_{255} = 255$. For a balanced histogram with midtones, *b* was set to 127.5. In order to determine the value of

Table 3.2. Ten cases from Yale Face Database

Case	Poses for training set (for all 15 members)	Poses for testing set (for all 15 members)
1	‘a’, ‘c’, ‘d’, ‘e’, ‘g’, ‘h’, ‘i’	‘f’ – normal pose
2	‘a’, ‘d’, ‘e’, ‘g’	‘f’ – normal pose
3	‘a’, ‘e’	‘f’ – normal pose
4	‘a’	‘f’ – normal pose
5	‘a’, ‘d’, ‘e’, ‘g’	‘c’ – happy pose
6	‘a’, ‘d’, ‘e’, ‘g’	‘h’ – sad pose
7	‘a’, ‘d’, ‘e’, ‘g’	‘i’ – sleepy pose
8	‘a’, ‘d’, ‘e’, ‘g’	‘j’ – surprised pose
9	‘a’, ‘d’, ‘e’, ‘g’	‘k’ – wink pose
10	‘a’, ‘d’, ‘e’, ‘g’	‘b’ – with glasses pose

c, an exhaustive search was conducted in that the PHM-PCA was applied to a total of ten cases, each with specific training set and testing set. As shown in Table 3.2, these ten cases considered various facial expressions (Cases 1 - 9) and face obstruction (Case 10). For additional details on these ten cases the reader is referred to Appendix 1. In each case, the training set $\mathcal{D}$ consisted of several selected poses of all 15 members while the testing set consisted of all 15 members with one selected pose. For example, in Case 2, $\mathcal{D}$ contained a total of $M = 60$ face images involving $K = 15$ members, each with $L = 4$ poses 'a', 'd', 'e', and 'g'; and the testing set contained 15 images from 15 members, each with a normal pose 'f'. In the search, the PHM-PCA was applied to all ten cases with *c* varying in the range of [1, 1100] and face-space dimension *p* (see Section 2.2.3) varying in the range of [4, 15]. The value of *c* that achieved the best overall recognition rate was found to be $c = 100$. The above parameter values were employed throughout our simulations with the Yale Face Database and extended Yale Face Database B.

B. Choosing Parameters for Other Test Algorithms

For all PCA-based algorithms evaluated in this section, the number of eigenfaces (*p*) was consistently chosen to be in range of [4, 15], which represents 5% up to 50% of the available number of eigenfaces. For WPCA, the parameters of interest were chosen as in [61], i.e. $f_c = 0.6 f_{\max}$, $n = 5$ and $\alpha_\omega = 2.5$. While HE-PCA does not require any input parameter, in DCT-PCA algorithm 30% of DCT coefficients were employed for further processing with PCA [79].

C. Face/Non-Face and Member/Non-Member Discrimination

The training set $\mathcal{D}$ used in this case consisted of a total of $M = 48$ images from Yale Face Database with *K* = 12 individuals out of the 15 available subjects from the database, each with *L* = 4 poses, 'a', 'd', 'e' and 'g'. As a result, there were available $M = 48$ eigenfaces from which a subset of $p = 12$ was chosen to represent the face images.

The testing set used to evaluate the discrimination performance of PCA, WPCA, HE-PCA, DCT-PCA and PHM-PCA between face/non-face and member/non-member images consisted of pose 'f' of the 12 individuals from the training set, who represented the member images, same pose 'f' of the remaining three individuals from the database, who represented the non-member images, labeled as *img_8f*, *img_11f* and *img_15f,* plus

Figure 3.16. The three non-face images *airplane_1*, *boats_1* and *goldhill_1*, obtained from cropping the original images.

three non-face images, labeled as *airplane_1, boats_1* and *goldhill_1*, obtained by cropping images *airplane, boats* and *goldhill* to size 128×128, see Figure 3.16.

In what follows, we use $\mathrm{C}^{(fm)}$ to denote the set of face/member images, and $\mathrm{C}^{(nf)}$ and $\mathrm{C}^{(nm)}$ to denote the sets of non-face images and non-member images, respectively.

To evaluate the discrimination performance of the five methods, we introduce a measure called gap$^{(f)}$, which quantifies the distance between the class of non-face images $\mathrm{C}^{(nf)}$ and the class of face images $\mathrm{C}^{(fm)}$ with respect to face space $\tilde{U}$ (see Section 2.2.3). This measure is defined by

$$\mathrm{gap}^{(f)} = \frac{\min(d_0^{(nf)}) - \max(d_0^{(f)})}{\min(d_0^{(nf)})} \cdot 100 \ \ (\%) \tag{3.15}$$

where $\min(d_0^{(nf)})$ is the smallest d_0 defined by (2.19) among all non-face images in $\mathrm{C}^{(nf)}$, and $\max(d_0^{(f)})$ denotes the largest d_0 among all face images in $\mathrm{C}^{(fm)}$.

In addition, a similar measure gap$^{(m)}$ is defined for quantifying the distance between the class of non-member images $\mathrm{C}^{(nm)}$ and the class of member images $\mathrm{C}^{(fm)}$ with respect to face space $\tilde{U}$ as

$$\mathrm{gap}^{(m)} = \frac{\min(d_{\min}^{(nm)}) - \max(d_{\min}^{(m)})}{\min(d_{\min}^{(nm)})} \cdot 100 \ \ (\%) \tag{3.16}$$

where $\min(d_{\min}^{(nm)})$ is the smallest $d_{\min}$ defined by (2.20) among all non-member images in $\mathrm{C}^{(nm)}$, and $\max(d_{\min}^{(m)})$ denotes the largest $d_{\min}$ among all member images in $\mathrm{C}^{(fm)}$.

Table 3.3. Face/non-face and member/non-member gaps [85]

PCA		WPCA		HE-PCA		DCT-PCA		PHM-PCA	
$gap^{(f)}$	$gap^{(m)}$	$gap^{(f)}$	$gap^{(m)}$	$gap^{(f)}$	$gap^{(m)}$	$gap^{(f)}$	$gap^{(m)}$	$gap^{(f)}$	$gap^{(m)}$
43.09	-	21.42	-	47.96	4.64	43.23	-	50.93	6.91

From the definition in (3.15), it follows that a bigger positive $gap^{(f)}$ indicates easier face/non-face discrimination, while a negative value of $gap^{(f)}$ indicates that no discrimination can be made, as the classes $\complement^{(nf)}$ and $\complement^{(fm)}$ overlap with each other. A similar claim can be made for member/non-member discrimination based on definition (3.16). The evaluation results are summarized in Table 3.3 from which it is observed that PCA and DCT-PCA produced a high gap for face/non-face discrimination, but failed to discriminate members from non-members; WPCA provided only a small gap for face/non-face discrimination and failed in member/non-member discrimination; HE-PCA succeeded in discriminating both cases; and the highest gaps for face/non-face discrimination and member/non-member discrimination was offered by PHM-PCA.

Finding the gaps can be considered as a training phase for face/non-face and member/non-member discrimination. Using the information involved in calculating the gap measures in (3.15) and (3.16), the face/non-face discrimination threshold can be localized in the middle of the distance between the class of non-face images $\complement^{(nf)}$ and the class of face images $\complement^{(fm)}$ (i.e. in the middle of $gap^{(f)}$), and obtained as

$$thr^{(f)} = \max(d_0^{(f)}) + \frac{\min(d_0^{(nf)}) - \max(d_0^{(f)})}{2} \tag{3.17}$$

while for member/non-member discrimination the threshold can be localized in the middle of the distance between the class of non-member images $\complement^{(nm)}$ and the class of member images $\complement^{(fm)}$ (i.e. in the middle of $gap^{(m)}$), and obtained as

$$thr^{(m)} = \max(d_{\min}^{(m)}) + \frac{\min(d_{\min}^{(nm)}) - \max(d_{\min}^{(m)})}{2} \tag{3.18}$$

In the light of definitions in (3.17) and (3.18), given a test image Γ, first the Euclidean distance d_0 between the input image Γ and the face space (see (2.19)) is compared with the threshold $thr^{(f)}$ in (3.17). If distance d_0 is found to be below the value of $thr^{(f)}$, the input image Γ is classified as a face image, otherwise it is considered

a non-face image. Next, if Γ turns out to be a face image, it can be classified as a class member or non-member face, by comparing the distance $d_{\min}$ (see (2.20)) with the threshold $thr^{(m)}$ in (3.18). If $d_{\min} < thr^{(m)}$, then the input image Γ is identified as a member of a class, else it is considered a non-member.

To illustrate the discrimination process, we examined the same training set with a total of $M = 48$ images from Yale Face Database and p = 12 eigenvectors, and a new set of test images consisted of pose 'c' of the 12 individuals from the training set, who represented the member images, same pose 'c' of the remaining three individuals from the database, who represented the non-member images, labeled as *img_8c*, *img_11c* and *img_15c,* plus three non-face images, labeled as *airplane_2, boats_2* and *goldhill_2*, obtained by cropping images *airplane, boats* and *goldhill* to size 128×128, see Figure 3.17.

We hierarchically performed face/non-face and member/non-member discrimination where the threshold values were set in accordance with (3.17) and (3.18). The face classification results for the five algorithms under testing are given in Table 3.4 in terms of TP, TN, FP, FN and TPR (see Sections 2.3.1 and 2.3.2). We remark that a (member or non-member) face image correctly identified as belonging to face class contributes one unit to TP, while a non-face image correctly identified as belonging to non-face class contributes one unit to TN. The performance measure FPR was not included in Table 3.4, as the total number of testing images that were not faces or not members was rather small, hence the ratio based on them was not representative in a statistical sense. At a glance, one may see that, in fact, a zero value of FPR was attended by all PCA-based algorithms, with only one exception for WPCA.

Figure 3.17. Another three non-face images, *airplane_2*, *boats_2* and *goldhill_2*, obtained from cropping the original images.

Table 3.4. Face classification results for the five PCA-based algorithms

	PCA	WPCA	HE-PCA	DCT-PCA	PHM-PCA
TP	13	10	14	13	14
TN	3	2	3	3	3
FP	0	1	0	0	0
FN	2	5	1	2	1
TPR	0.87	0.67	0.93	0.87	0.93

Table 3.5. Member classification results for HE-PCA and PHM-PCA algorithms

	HE-PCA	PHM-PCA
TP	10	11
TN	1	1
FP	1	1
FN	2	1
TPR	0.83	0.92

Next, for the HE-PCA and PHM-PCA algorithms who provided a positive value for $\text{gap}^{(m)}$ (see Table 3.3), the TP face-identified instances from Table 3.4 were further analyzed for member/non-member discrimination. Table 3.5 illustrates the performance of the five algorithms in classification of member images by means of TP, TN, FP, FN and TPR.

Finally, on comparing the results in Table 3.3 with those in Tables 3.4 and 3.5, we note a positive correlation between the gaps and TPR, which justifies the definition of the gaps in (3.15) and (3.16). In addition, although this section is not concerned with the face identification task, we note that the PHM-PCA algorithm provided a face recognition rate of 81.8% as opposed to 70.0% of the HE-PCA algorithm, when the TP instances for member discrimination process were considered (see Table 3.5).

D. Face Identification

As a first step of our simulations for face identification, all ten cases in Table 3.2 were examined. Figure 3.18 illustrates the comparison results of the five methods in terms of recognition rate versus number (p) of eigenfaces employed. The plots in Figure 3.18 show how the recognition rate was improved by utilizing the PHM-PCA algorithm as long as more than 3 eigenvectors were employed for image representation. It is also observed that the performance of PHM-PCA algorithm was quite robust versus the number of images used in the training set. As a matter of fact, even with a reduced

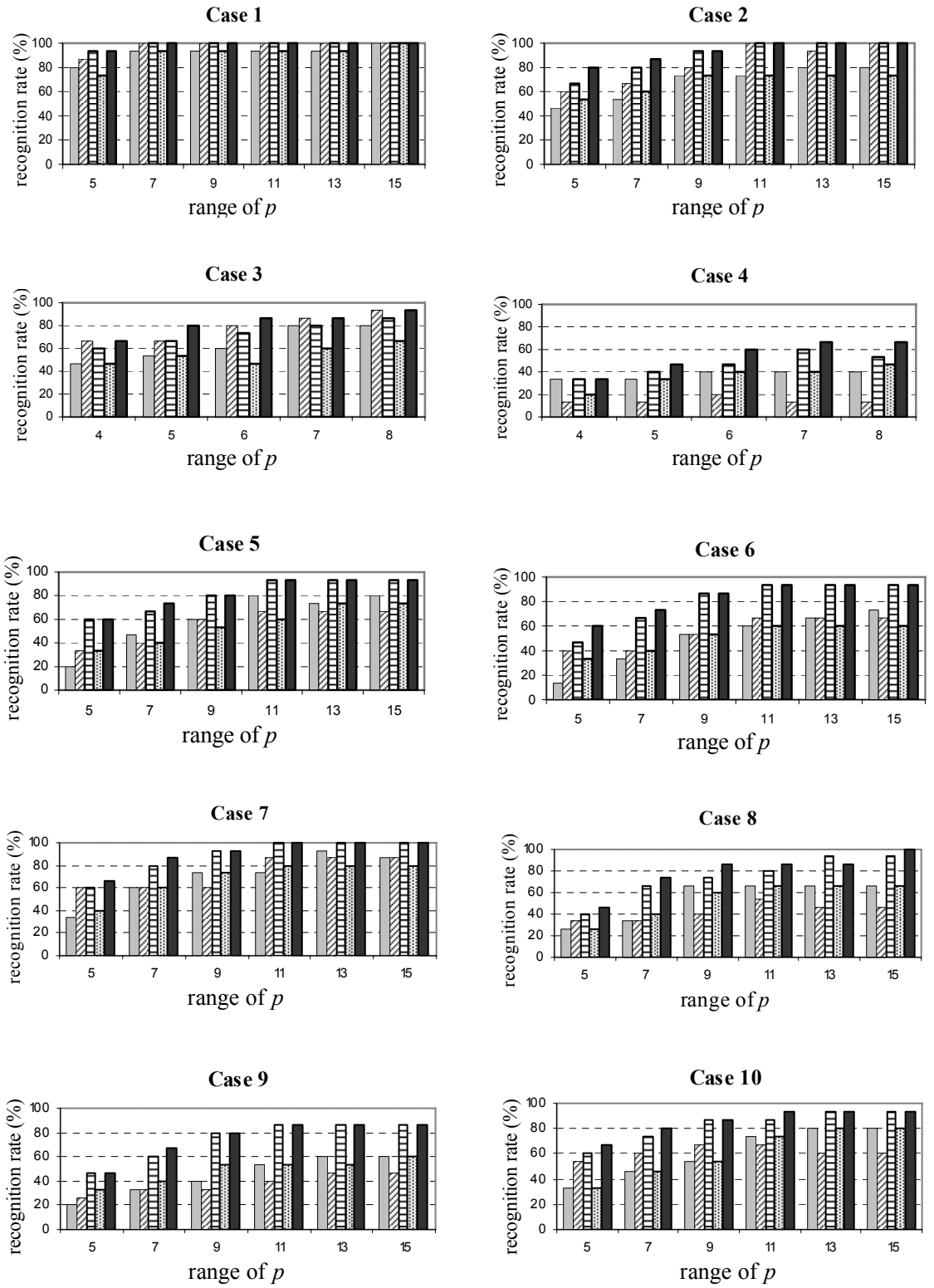


Figure 3.18. Comparison results for PCA (solid grey bar), WPCA (diagonal stripped bar), HE-PCA (horizontal stripped bar), DCT-PCA (dotted bar) and PHM-PCA (solid black bar) using the Yale Face Database.

training set such as in Cases 3 and 4, the PHM-PCA method outperformed the other four algorithms.

In terms of identification robustness to changes in facial expression (Cases 1 – 9), Figure 3.18 shows that PHM-PCA demonstrated satisfactory performance, with one exception when the training set was very small (Case 4). For slightly obstructed facial images (Case 10), PHM-PCA also offered the best performance among the five algorithms tested.

3.7.2 Results for the Extended Yale Face Database B

Face identification under various lighting conditions was also examined in our simulations. For this we employed the extended Yale Face Database B with a selection of 560 images representing 20 individuals with 28 poses per person. Each image was further manually cropped to a size of 168×168. The training set contained a total of 400 images representing 20 individuals with 20 poses per individual (see Figure 3.19). For each individual, eight new poses with various illumination conditions were considered for testing. This yielded eight testing sets, each containing 20 facial images. Figure 3.20 illustrates as an example eight poses of an individual, each of which belongs to a testing set.

The effect of adopting various number (p) of eigenvalues and lighting conditions on face recognition rate by the five algorithms is illustrated in Figure 3.21. It is observed that as long as the lighting condition was such that did not generate large shadowed areas on the face (Cases 1 – 4), PHM-PCA, HE-PCA, and WPCA exhibited comparable and satisfactory performance. Under more extreme lighting conditions (Cases 5 – 8), PHM-PCA was found to outperform the other four algorithms with one exception in Case 8 with $p = 20$.

Finally, the complexity of the algorithms was examined in terms of normalized elapsed time. Elapsed time rather than the amount of arithmetic computations was chosen as a complexity measure because some of the algorithms under comparison, including HE-PCA and PHM-PCA, involve considerable non-arithmetic operations. Here the elapsed time was normalized so that the elapsed time taken by the conventional PCA to perform a face identification task was set to unity.

Figure 3.19. Training set containing 20 individuals (top four rows) with 20 poses per individual (bottom four rows, exemplification for first individual) from the extended Yale Face Database B.

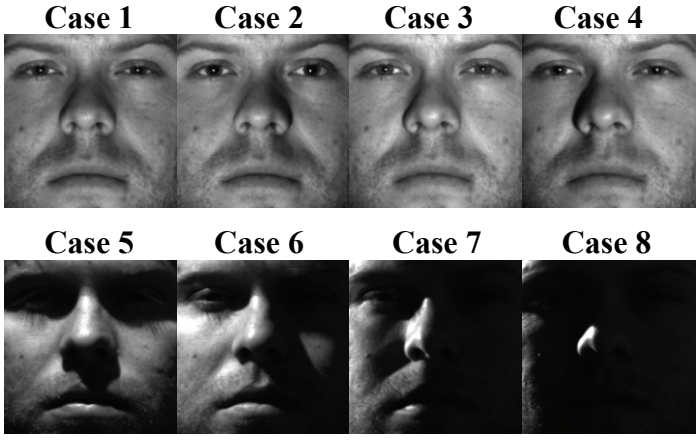


Figure 3.20. Eight illumination conditions considered for eight testing sets.

Table 3.6. Normalized elapsed time for the five algorithms

PCA	WPCA	HE-PCA	DCT-PCA	PHM-PCA
1	1.32	1.10	0.54	1.24

Table 3.6 summarizes the average normalized elapsed time over 100 trials of the five algorithms. As expected, DCT-PCA was found to have least complexity. The elapsed time required by PHM-PCA was found slightly higher but comparable with those of HE-PCA and PCA, and less than that of WPCA.

3.7.3 Results Employing PHM – WII PCA Algorithm

The PHM – WII pre-processing module in conjunction with standard PCA was tested on both the Yale and extended Yale face databases.

Figure 3.22 presents the recognition rates achieved by WII PCA versus those obtained by PHM – WII PCA, when the ten cases from Yale Face Database were considered. In most cases, the performance of the PHM – WII PCA method in terms of recognition rate was found to be superior to that of WII PCA. Noticeable exceptions were encountered in Cases 3 and 4 as a result of employing a reduced number of eigenfaces.

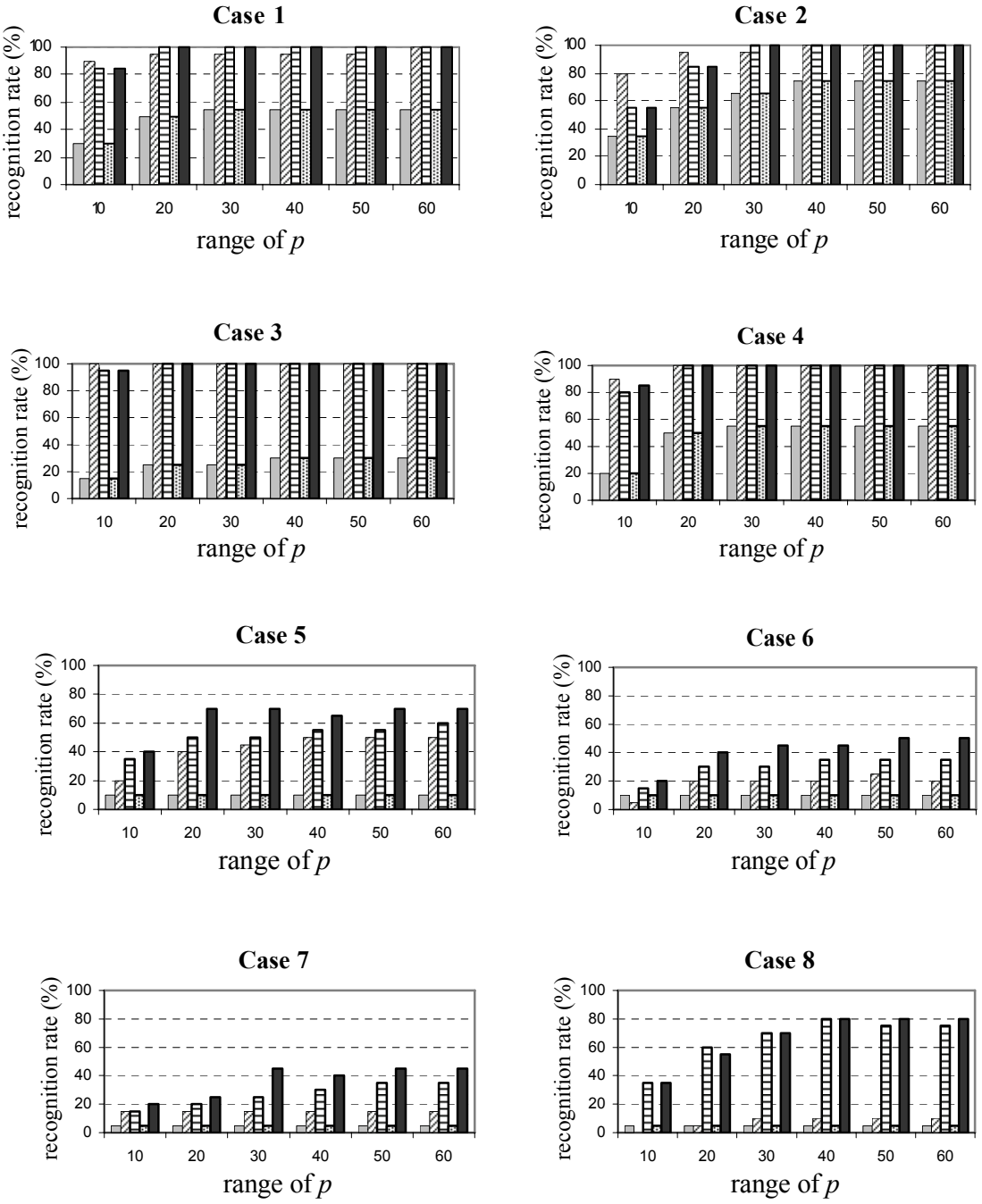


Figure 3.21. Comparison results for PCA (solid grey bar), WPCA (diagonal stripped bar), HE-PCA (horizontal stripped bar), DCT-PCA (dotted bar) and PHM-PCA (solid black bar) using the extended Yale Face Database B.

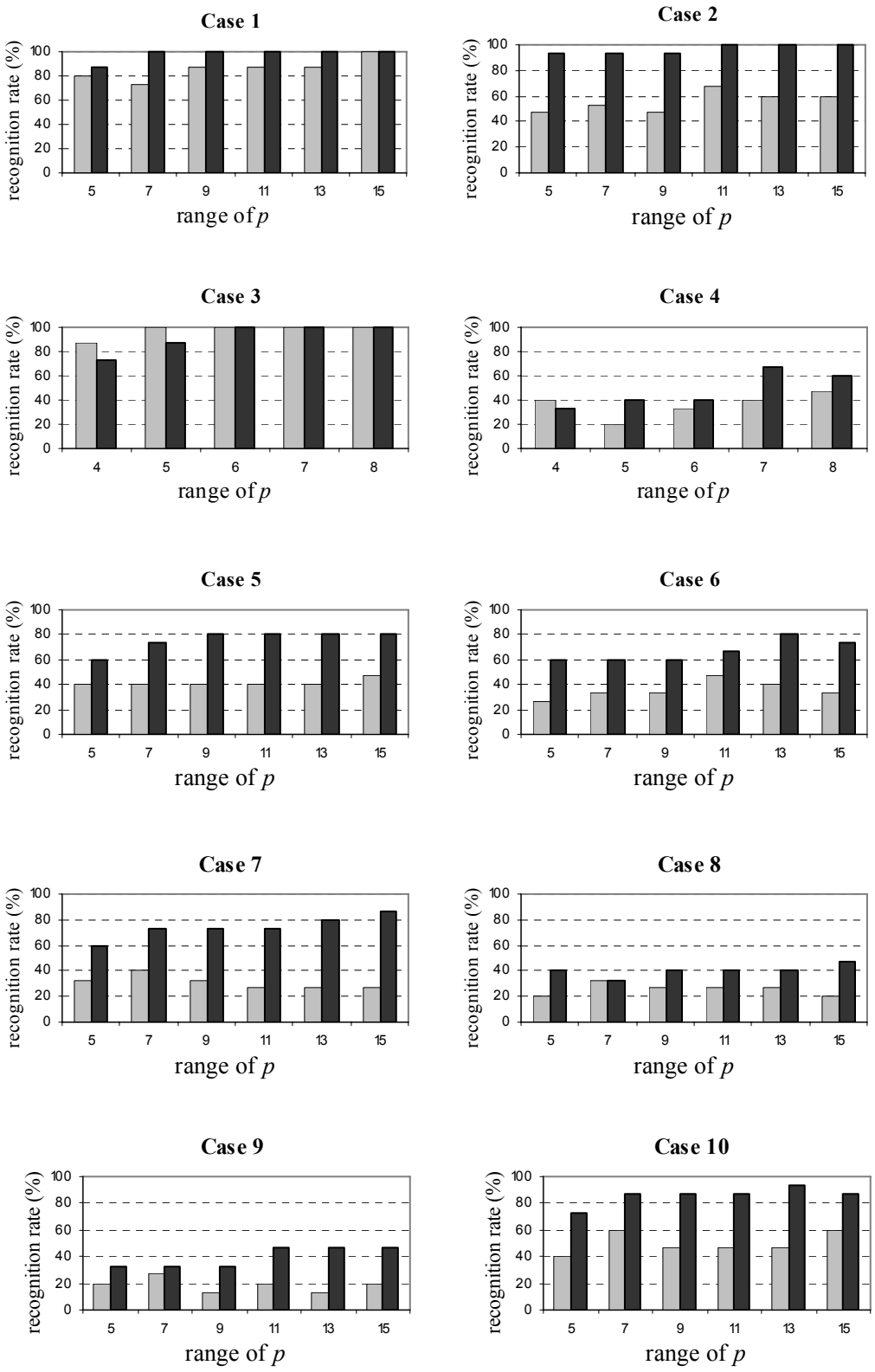


Figure 3.22. Comparison results for WII PCA (solid grey bar) and PHM-WII PCA (solid black bar) using the Yale Face Database.

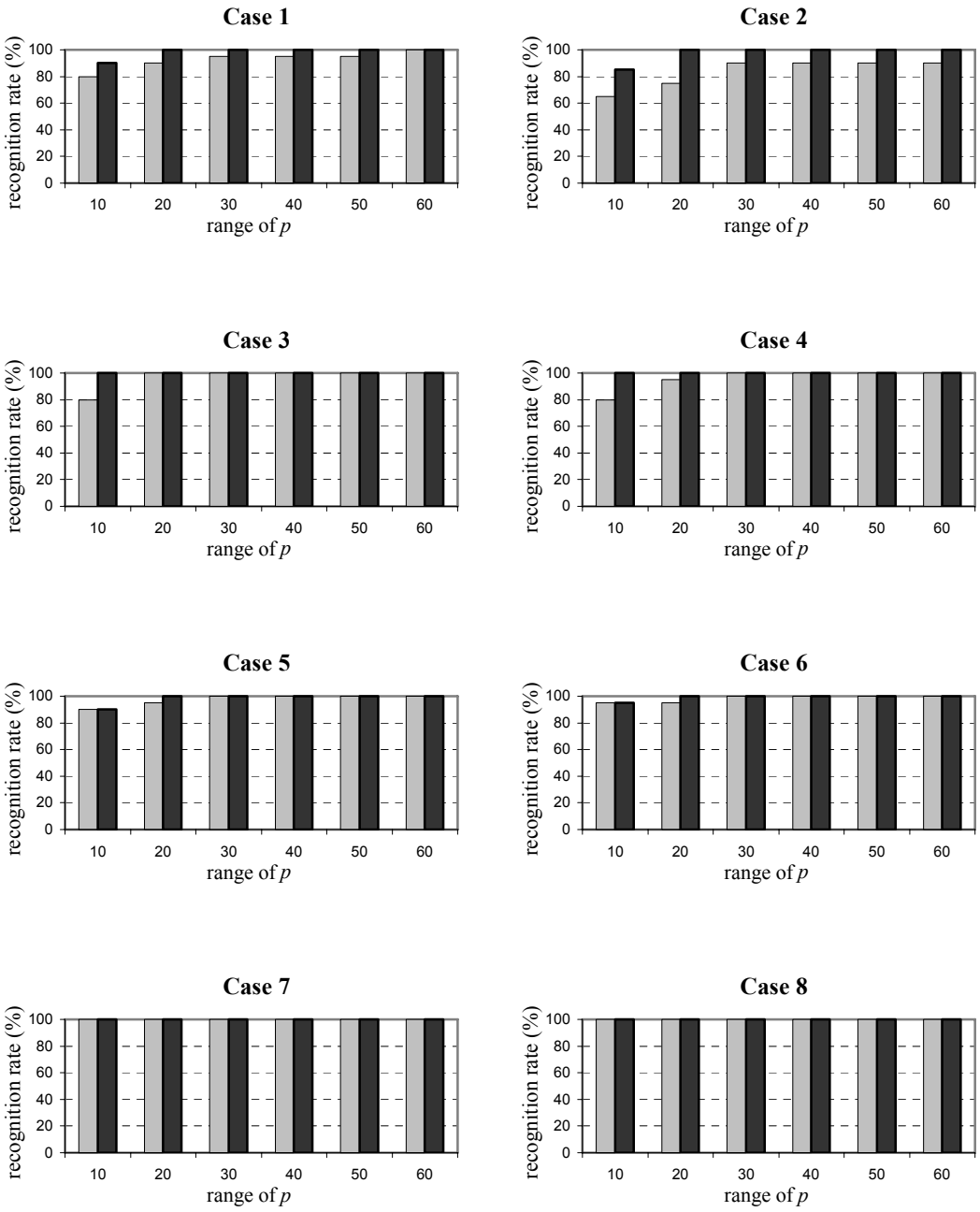


Figure 3.23. Comparison results for WII PCA (solid grey bar) and PHM-WII PCA (solid black bar) using the extended Yale Face Database B.

For the eight cases using the extended Yale Face Database B, Figure 3.23 illustrates the results attained by PHM – WII PCA compared with those obtained by WII PCA.

We can see that for facial images with no pose variations, when more that 20 eigenfaces were used (representing a total of 5% of the available number of eigenfaces),

the combined PHM – WII PCA algorithm achieved 100% recognition rate for all eight cases. Surprisingly, for Cases 7 and 8 with more drastic illumination conditions, both algorithms under comparison obtained 100% recognition rates even when 10 eigenfaces were employed.

We also note that for images with illumination variations (i.e. images from extended Yale face database B) the performance of both algorithms was substantially better than for images with slight pose variations (i.e. images from Yale face database), which was expected as the WII module is an illumination invariant pre-processing step specifically designed to deal with various lighting conditions. The PHM module was found to bring further improvement when applied as a pre-processing step prior to the application of the WII PCA algorithm.

3.7.4 Robustness to Noise and Face Occlusions

In order to evaluate the performance of WT – TV and OCCL pre-processing modules in conjunction with PHM PCA algorithm, we chose the extended Yale Face Database B as the dataset, as its images present a large variety of illumination conditions that imposed additional challenges for the de-noising and face occlusion-resolving algorithms compared to the moderate pose variations for the images from Yale Face Database.

A. Results for Noise-Contaminated Images

In order to evaluate the de-noising capability of WT – TV algorithm we considered the eight cases from the extended Yale Face Database B. The PHM PCA method was chosen as a point of reference and it was tested using noise-contaminated test images. The test images were obtained by adding white Gaussian noise with various variance values (see Section 3.7.5 for further details on noise amount).

Figure 3.24 illustrates the performance of PHM PCA under three scenarios: (a) no noise was applied to the test images; (b) noise was applied to test images, but no de-noising was performed; and (c) noise was applied to test images followed by WT – TV de-noising.

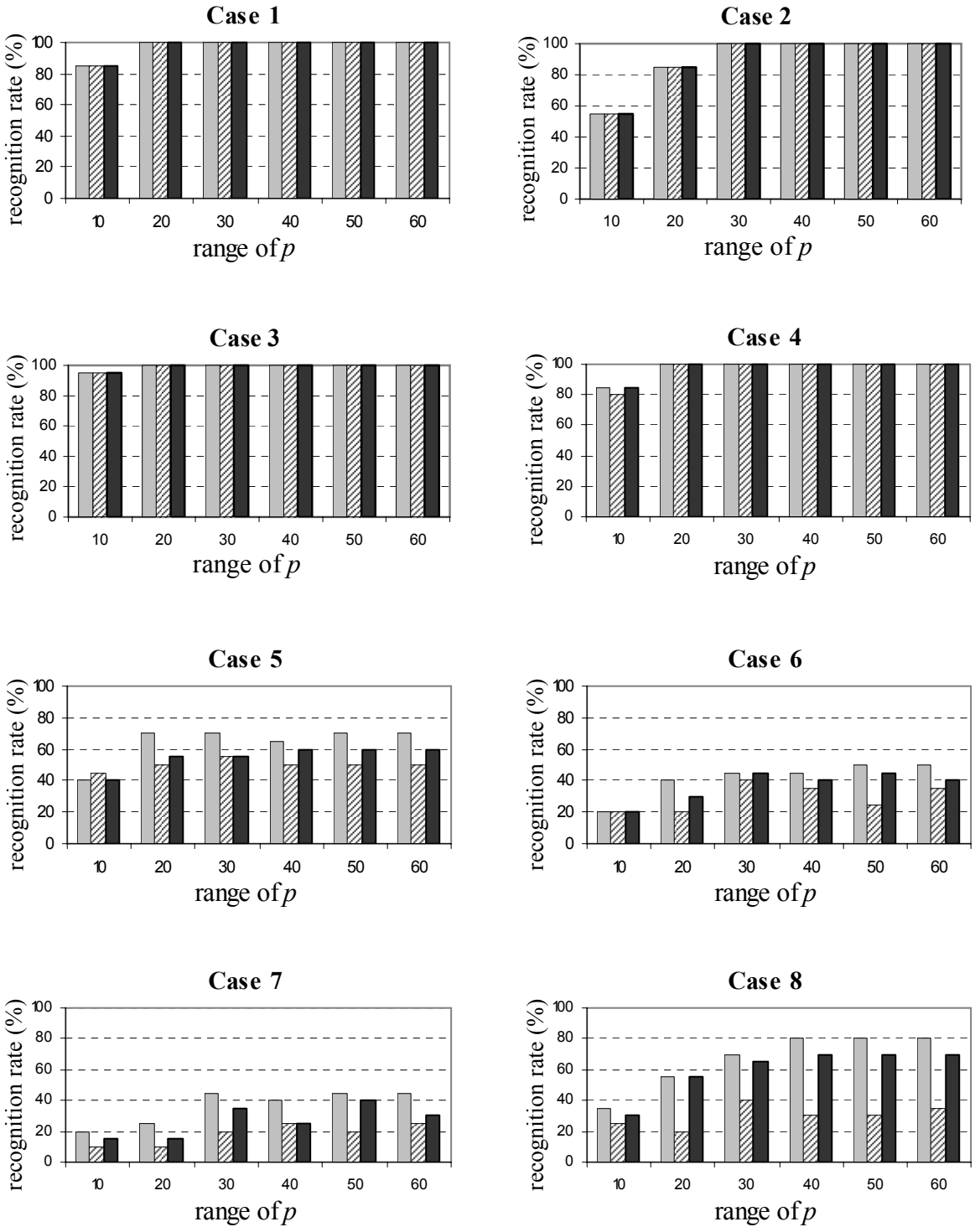


Figure 3.24. Comparison results for PHM PCA (solid grey bar), PHM PCA with noisy test images and no de-noising (diagonal stripped bar), and PHM PCA with noisy test images and WT – TV de-noising (solid black bar) using the extended Yale Face Database B.

As expected, when noise was added to the test images and no de-noising was performed, the recognition rates were reduced, especially for more challenging Cases 5 to 8, with a drastic decrease of recognition rate in Case 8. The WT – TV de-noising algorithm yields good results in the last four cases depicted in Figure 3.24, increasing the recognition rate up to 40% (see Case 8).

B. Results for Images with Face Occlusion

The ability of the algorithms to deal with facial occlusion was examined for the eight cases of extended Yale face database B. The PHM PCA algorithm was used as a point of reference. The assumption that all testing images have occlusions was made. Results of combining the OCCL pre-processing module with PHM PCA algorithm are presented in Figure 3.25. We remark that eyes occlusion is found to have a greater impact than chin occlusion on recognition capability of OCCL PHM PCA algorithm for more challenging Cases 5 – 8.

3.7.5 Implementation Issues

The implementation of proposed and tested algorithms presented in this thesis was carried out using MATLAB. Below we discuss some of the issues encountered during the implementation of the above pre-processing algorithms.

A. WII Algorithm

- As a first step of the algorithm, the logarithm function was applied to original gray scale image which has pixel intensity values between 0 and 255. However, for zero intensity values, the logarithm function is not well defined. The solution was to add a small positive constant ε to each image pixel intensity value before taking the logarithm of image, and to subtract the same amount ε after the exponential function was performed. A proper value of ε with satisfactory performance was found to be 1.

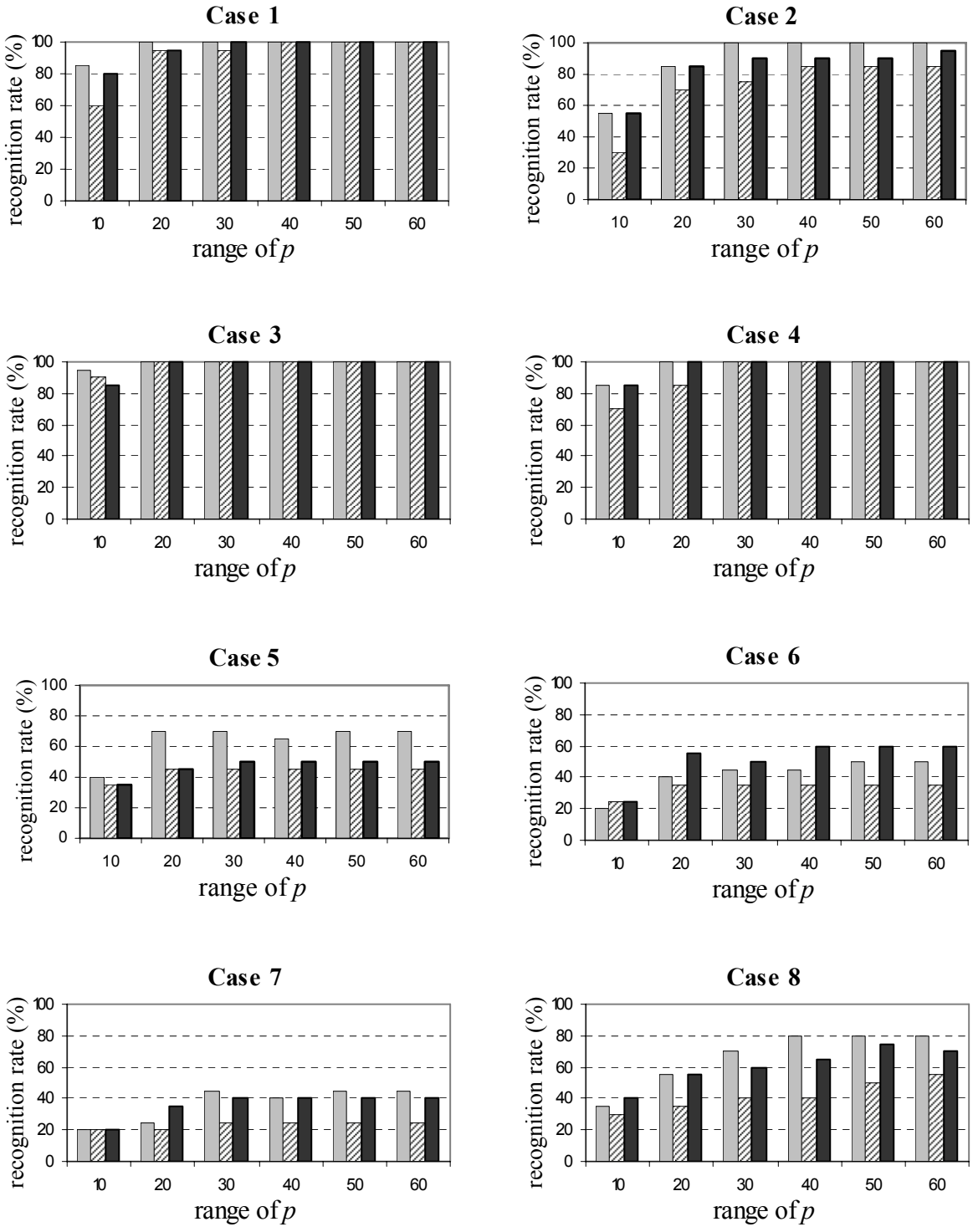


Figure 3.25. Comparison results for PHM PCA (solid grey bar), OCCL PHM PCA with eyes occlusion (diagonal stripped bar), and OCCL PHM PCA with chin occlusion (solid black bar) using the extended Yale Face Database B.

- Wavelet transform was applied to an image employing the MATLAB functions "wavedec2" for the wavelet decomposition (analysis) step and "waverec2" for the wavelet reconstruction (synthesis) step [110]. These functions are available in the Wavelet Toolbox. Alternatively, wavelet transform for images was carried out with identical results using commands "wt2d" and "iwt2d" available through the freeware Uni_Wave Toolbox [111].

B. HE Algorithm

- Histogram equalization function was implemented using the MATLAB function "histeq". The MATLAB command maps roughly equal number of pixels to each gray level producing a new histogram, which is approximately flat [110].

C. WT – TV De-Noising Algorithm

- Throughout the entire set of simulations related to the WT – TV de-noising procedure, the noise contaminated test images were constructed as follows. For each individual image, a specific seed was assign to generate the random noise. The maximum noise amount was varied, from image to image up to a value of 10% of the maximum pixel light intensity. As a result, various variance values were obtained for each image, up to a maximum value of 0.0041, when a normalized range of pixel light intensities (i.e. [0, 1]) was considered.
- To implementing the TV minimization procedure in Section 3.5.2, uniform grid points were used, with grid spacing $\Delta x = \Delta y = h = 1$ [65].

3.8 Summary

The PHM method proposed in this chapter is conceptually simple, easy to apply and computationally efficient, and it can be used as a pre-processing module in combination with face recognition processing techniques, for instance with conventional PCA. Our experimental results have shown its efficiency in improving face recognition rates in comparison with several existing pre-processing techniques, e.g. WPCA [61], DCT-PCA [79] and HE-PCA. In addition, it may be used in combination with other pre-processing steps, such as WII technique [38], WT-TV de-noising or OCCL occlusion-resolving algorithm.

Chapter 4

An Extended Two-Dimensional Principal Component Analysis Technique

4.1 Introduction

Numerous PCA-based methods have been developed to improve PCA performance since the original work on PCA [96]. As argued in [3], a considerable amount of information that is useful for face recognition tasks may be contained in the higher-order relations among pixels. On the other hand, the PCA considers only second-order statistical information, taking into account the global structure of the image. In [19], [41], [46] and [93] the original PCA is enhanced by taking into account local features by sub-pattern techniques, partitioning the original face images into several smaller sub-patterns. Moreover, the method in [46] combines an image partition technique with vertically centered PCA and whitened horizontally centered PCA to obtain a novel hybrid approach with recognition performance superior to the traditional methods.

A two-dimensional (2-D) approach was proposed by Yang et al. in [100]. It consists of an image projection technique in which images are treated as matrices instead of vectors as in the original PCA [96]. This method leads not only to better recognition rates, but also to improved computationally efficiency.

In this chapter, we take a close look at the 2-D PCA algorithm in [100] and propose an extended 2-D PCA (E-2DPCA) algorithm that utilizes a *pair* of covariance matrices to

extract both row-related and column-related features of facial images. In conjunction with our extension is a new criterion for face classification. Finally, we examine the performance of the proposed algorithm and its robustness to noise and face occlusion.

This chapter is organized as follows. Section 2 introduces some background material related to the 2-D PCA technique. In Section 3, an extended 2-D PCA is proposed along with a new classification criterion. Performance evaluations of the proposed technique and comparisons with standard PCA and 2-D PCA methods are presented in Section 4, where experimental results illustrating the robustness of E-2DPCA to noise and face occlusion are also included. In addition, Section 4 addresses several implementation issues encountered during the evaluation tests.

4.2 An Overview of 2-D PCA Method

Herein we present an overview of the 2-D PCA method recently proposed by Yang et al. [100].

Unlike the conventional PCA algorithm (see Section 2.2.3), the 2-D PCA (2DPCA) algorithm developed in [100] treats an image of size $N \times N$ as an $N \times N$ matrix rather that a vector of length N^2. This is obviously a more natural way to deal with digital images, and it leads to a covariance matrix of much reduced size compared to its 1-D PCA counterpart. Given a set of M training face images $\{A_i, i = 1, 2, \ldots, M\}$, a covariance matrix is constructed as

$$G = \frac{1}{M}\sum_{i=1}^{M}\left(A_i - \overline{A}\right)^T\left(A_i - \overline{A}\right) \tag{4.1}$$

where $\overline{A}$ denotes the mean of the M training face images, i.e., $\overline{A} = \frac{1}{M}\sum_{k=1}^{M} A_k$. Let $\{x_1, x_2, \cdots, x_p\}$ be the orthonormal eigenvectors of G associated with the first largest p eigenvalues. Each image A_i in the training set corresponds to a feature matrix $B_i = \left[\, y_1^{(i)} \quad y_2^{(i)} \quad \cdots \quad y_p^{(i)} \,\right]$ with columns given by the linear transformation

$$y_k^{(i)} = A_i x_k \quad \text{for } k = 1, 2, \ldots, p. \tag{4.2}$$

Similarly, a test image A is associated with a feature matrix $B = [y_1\ y_2\ \cdots\ y_p]$ where $y_k = Ax_k$. In this way, a distance between a test image A and a training image A_i is induced in terms of the Euclidean distance between their feature matrices as

$$d(B, B_i) = \sum_{k=1}^{p} \left\| y_k - y_k^{(i)} \right\|_2 . \tag{4.3}$$

Based on the evaluation of these distances $d(B, B_i)$ for $i = 1, 2, ..., M$, a test image can be identified as whether or not it is a face image and, if it is, to which class it belongs, in a way similar to the 1-D PCA method of Turk and Pentland [96].

4.3 An Extended 2-D PCA Technique

In what follows we propose an extension of the existing 2DPCA algorithm, referred as E-2DPCA [86]. First, observations that motivate our study are made. This is followed by technical details of the algorithm and experimental results.

4.3.1 Motivation

Eqs. (4.1) and (4.2) describe two key components of the 2-D PCA algorithm [100]: (4.1) constructs a covariance matrix G of the training facial images and (4.2) builds a feature matrix of a facial image by orthogonal projection of the image onto the p most significant eigenvectors $\{x_k, k = 1, \ldots, p\}$ of G.

We note that the projections performed in (4.2) are for the *rows* of A_i. Consequently, matrices B and B_i defined in [100] (see Sec. 4.2) are row-related feature matrices. A question that naturally arises is why not to use also the column-related features of an image? And, subsequently, what should one do to generate a column-related feature matrix for a given image? Since the covariance matrix G in (4.1) is designed exclusively for row-related features, one needs a different covariance matrix for extracting column-related features. Presented below is an extended 2-D PCA technique in which we work with a pair of covariance matrices, associate them to respective feature matrices, and develop a new criterion for face classification.

4.3.2 The E-2DPCA Method

Given a set of M training face images A_i of size $m \times n$, which contains K classes (individuals) $\{\Omega_j, j = 1, \ldots, K\}$ of data, with each class having $L = M/K$ images denoted by $\{A_k^{(j)}, k = 1, 2, \cdots, L\}$. We define a pair of covariance matrices $\{G, H\}$ of respective sizes $n \times n$ and $m \times m$ as

$$G = \frac{1}{M}\sum_{i=1}^{M}\left(A_i - \overline{A}\right)^T\left(A_i - \overline{A}\right) \tag{4.4.a}$$

$$H = \frac{1}{M}\sum_{i=1}^{M}\left(A_i - \overline{A}\right)\left(A_i - \overline{A}\right)^T, \tag{4.4.b}$$

where $\overline{A} = \frac{1}{M}\sum_{k=1}^{M} A_i$. Note that $G \in R^{n\times n}$ in (4.4.a) is the same covariance matrix as defined in (4.1) which is employed in [100], while $H \in R^{m\times m}$ in (4.4.b) is a new covariance matrix designed for extracting column-related features (see below).

Let $\{x_k, k = 1, \ldots, p\}$ and $\{z_k, k = 1, \ldots, p\}$ be the p orthonormal eigenvectors of G and H that are associated with the first largest p eigenvalues, respectively. Given a test (facial) image A, we construct two feature matrices, denoted by R and C, by orthogonal projection of the rows of A onto $\{x_k, k = 1, \ldots, p\}$ and columns of A onto $\{z_k, k = 1, \ldots, p\}$, respectively. Namely,

$$R = A \cdot X = \begin{bmatrix} r_1 & r_2 & \cdots & r_p \end{bmatrix} \tag{4.5.a}$$

$$C = Z^T \cdot A = \begin{bmatrix} c_1 & c_2 & \cdots & c_p \end{bmatrix}^T, \tag{4.5.b}$$

where $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix}$ and $Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_p \end{bmatrix}$ are the optimal projection matrices of size $n \times p$ and $m \times p$, respectively. Note that $R \in R^{m\times p}$ is the same feature matrix as that used in [100], while $C \in R^{p\times n}$ gathers column-related features of the image.

In the same way, for each class Ω_j in the training set, a pair of class-average feature matrices $\{\overline{R}_j, \overline{C}_j\}$ are defined as the arithmetic means over the row-related and column-related feature matrices of the images in that class respectively, i.e.,

$$\overline{R}_j = \frac{1}{L}\sum_{k=1}^{L} R_k^{(j)} = \left[\frac{1}{L}\sum_{k=1}^{L} A_k^{(j)}\right]\cdot X = \overline{A}_j \cdot X \tag{4.6.a}$$

$$\overline{C}_j = \frac{1}{L}\sum_{k=1}^{L} C_k^{(j)} = Z^T \cdot \left[\frac{1}{L}\sum_{k=1}^{L} A_k^{(j)}\right] = Z^T \cdot \overline{A}_j, \tag{4.6.b}$$

where $\overline{A}_j = (\sum_{k=1}^{L} A_k^{(j)})/L$ is the arithmetic mean of the images in class Ω_j.

In this way, any test image and any class from training set has similar representations consisting in a pair of row-related and column-related feature matrices.

4.3.3 Classification Employing Nearest Neighbor Classifier

With the definitions and notation in (4.5) and (4.6), the Euclidean distances between a test facial image A and class Ω_j are defined as

$$\begin{cases} d_j^{(r)} = \sum_{k=1}^{p} \left\| r_k - r_k^{(j)} \right\|_2 \\ d_j^{(c)} = \sum_{k=1}^{p} \left\| c_k - c_k^{(j)} \right\|_2 \end{cases}, \tag{4.7}$$

where $r_k^{(j)}$ and $c_k^{(j)}$ are the k^{th} column and row of $\overline{R}_j$ and $\overline{C}_j$, respectively. Based on these, a new criterion for facial image classification can be formulated as follows:

(i) Let $d_r^* = \min_{1\le j\le K}(d_j^{(r)})$ that is reached at $j = j_r^*$.

(ii) Let $d_c^* = \min_{1\le j\le K}(d_j^{(c)})$ that is reached at $j = j_c^*$.

(iii) The facial image A is classified to belong to class Ω_{j^*} where

$$j^* = \begin{cases} j_r^* & \text{if } d_r^* \le d_c^* \\ j_c^* & \text{if } d_r^* > d_c^*. \end{cases} \tag{4.8}$$

We remark that by regarding the pair of Euclidean distances $\left[d_j^{(r)} \quad d_j^{(c)}\right]$ as a two-dimensional measure between a test image A and class Ω_j, the criterion proposed above is merely one of the many possibilities, which is found to offer satisfactory performance (see Section 4.4). Another possibility is to use a weighted l_1-norm to combine the two distances $d_j^{(r)}$ and $d_j^{(c)}$.

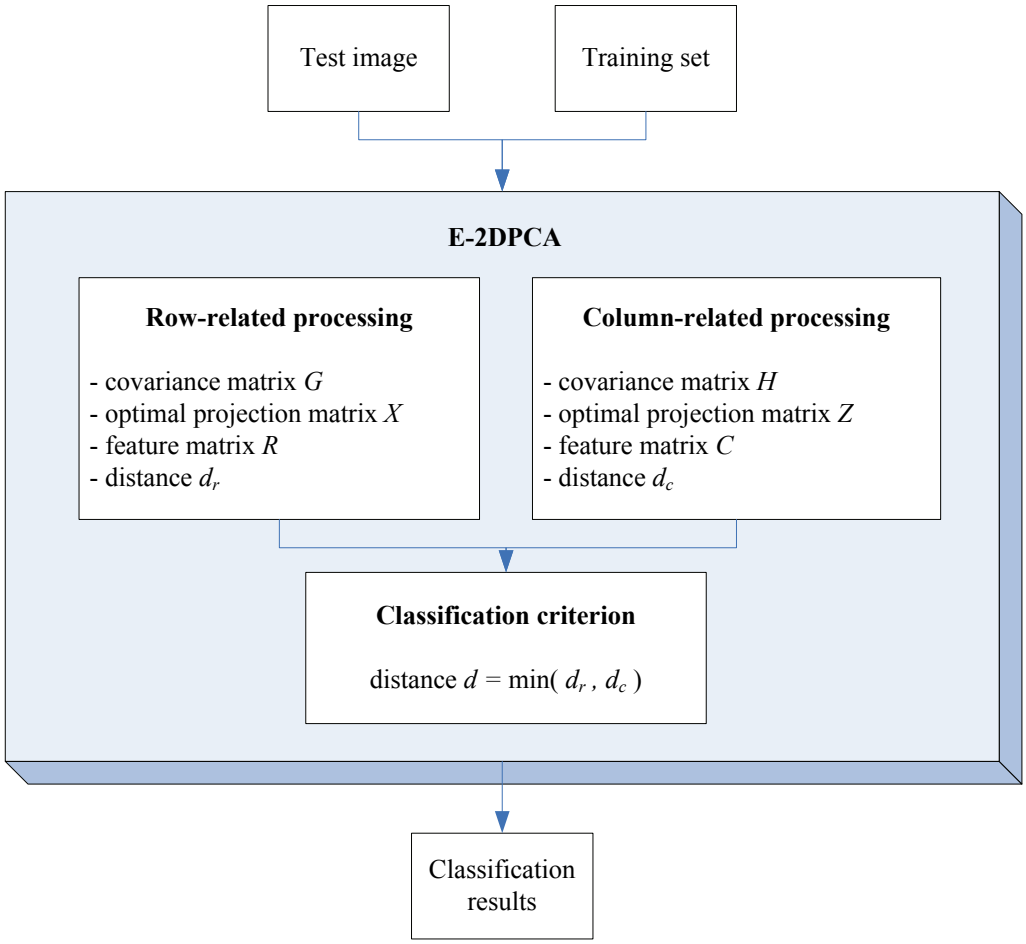


Figure 4.1. A block diagram of the E-2DPCA algorithm.

To summarize the proposed 2-D extension of existing 2DPCA method, the block diagram in Figure 4.1 illustrates the proposed E-2DPCA algorithm. It is expected that the dual processing (on rows and on columns) of a test face image will increase the likelihood of correct face identification.

4.4 Experimental Results

The performance of the E-2DPCA algorithm [86] was evaluated and compared with PCA [96] and 2DPCA [100]. The Yale Face Database and extended Yale Face Database B were used as image databases for performance evaluation as they include slight frontal pose variations and various illumination conditions, respectively.

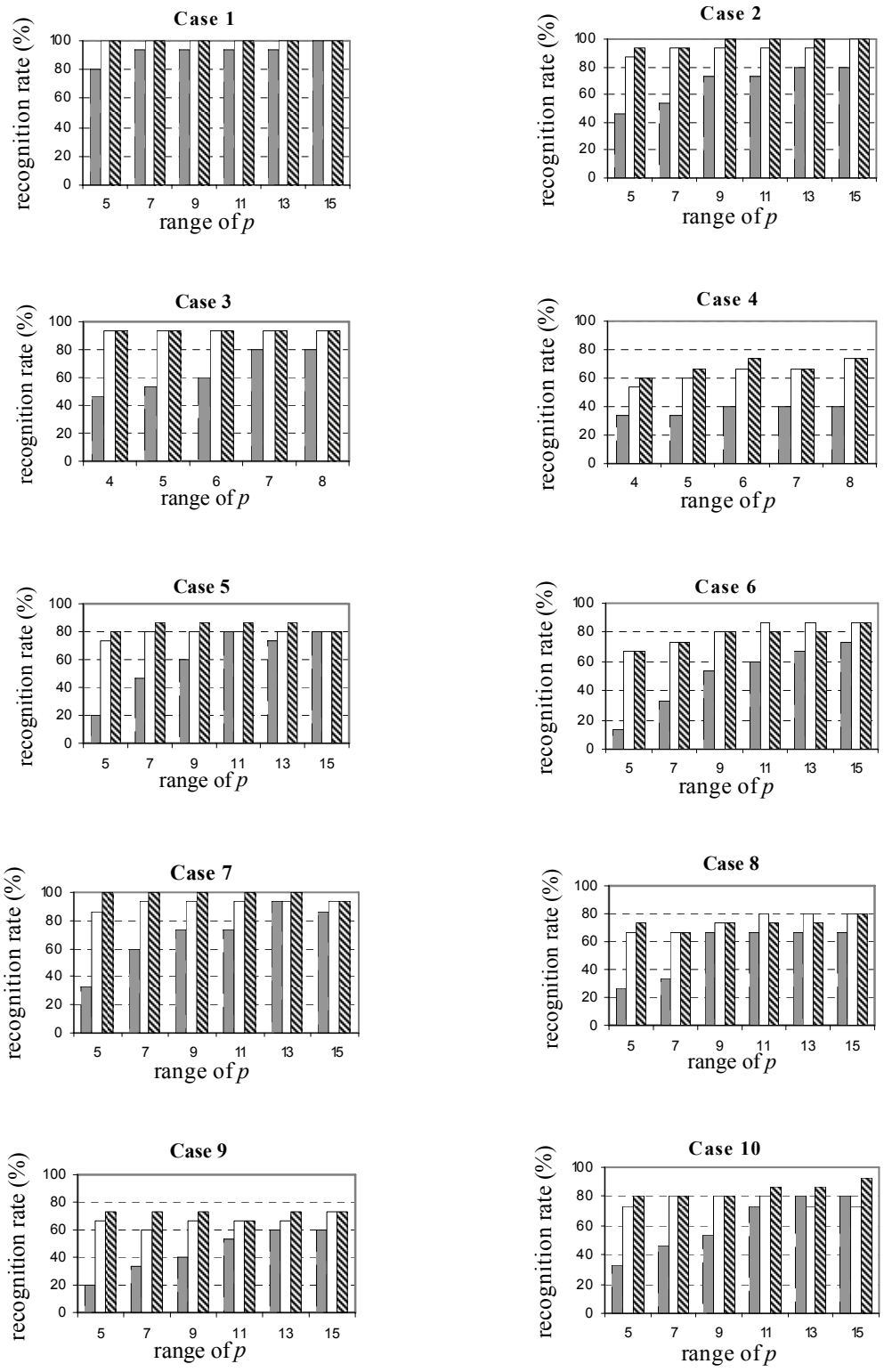


Figure 4.2. Comparison results for PCA (solid grey bar), 2DPCA (solid white bar) and E-2DPCA (diagonal stripped bar), for all ten cases from the Yale Face Database.

Figure 4.3. Six illumination conditions from the extended Yale Face Database B considered for six testing sets.

4.4.1 Performance Comparison

To investigate the algorithms' performance for facial images with various expressions and face obstruction, we examined the ten cases shown in Table 3.2 (see Appendix 1 for more details). Figure 4.2 illustrates the results of applying PCA, 2DPCA and E-2DPCA to these ten cases, in terms of recognition rate versus number of eigenfaces (p) employed. One can notice that the recognition rate was improved for Cases 2, 4, 5, 7, 9 and 10 when E-2DPCA was utilized. For Cases 1 and 3, the E-2DPCA has identical performance as the 2DPCA algorithm. The only two exceptions when the proposed extended algorithm has slightly worse performance that 2DPCA are encountered in Cases 6 and 8, for p = 11 and 13.

A separate set of experiments was conducted to examine the robustness of the proposed algorithm under illumination variations. The database employed was the extended Yale Face Database B with a selection of 520 images representing 20 persons with 26 poses per person. Each image was cropped to a size of 168×168. The training set contained 400 images representing 20 individuals, each with 20 poses (see Figure 3.19). The testing set was divided into two groups, named as group A and group B. Group A contained 60 images representing 20 individuals, each having 3 images with mild illumination variations, as shown in the first row in Figure 4.3. Group B also contained 60 images representing the same 20 individuals, each having 3 images with severe

illumination variations, see the second row in Figure 4.3. The images in both groups were different from those in the training set.

The results of applying the same three algorithms to the above testing set are depicted in Figure 4.4. The E-2DPCA algorithm was found to perform at least as good as the 2DPCA, with higher recognition rate for Cases 5 and 6.

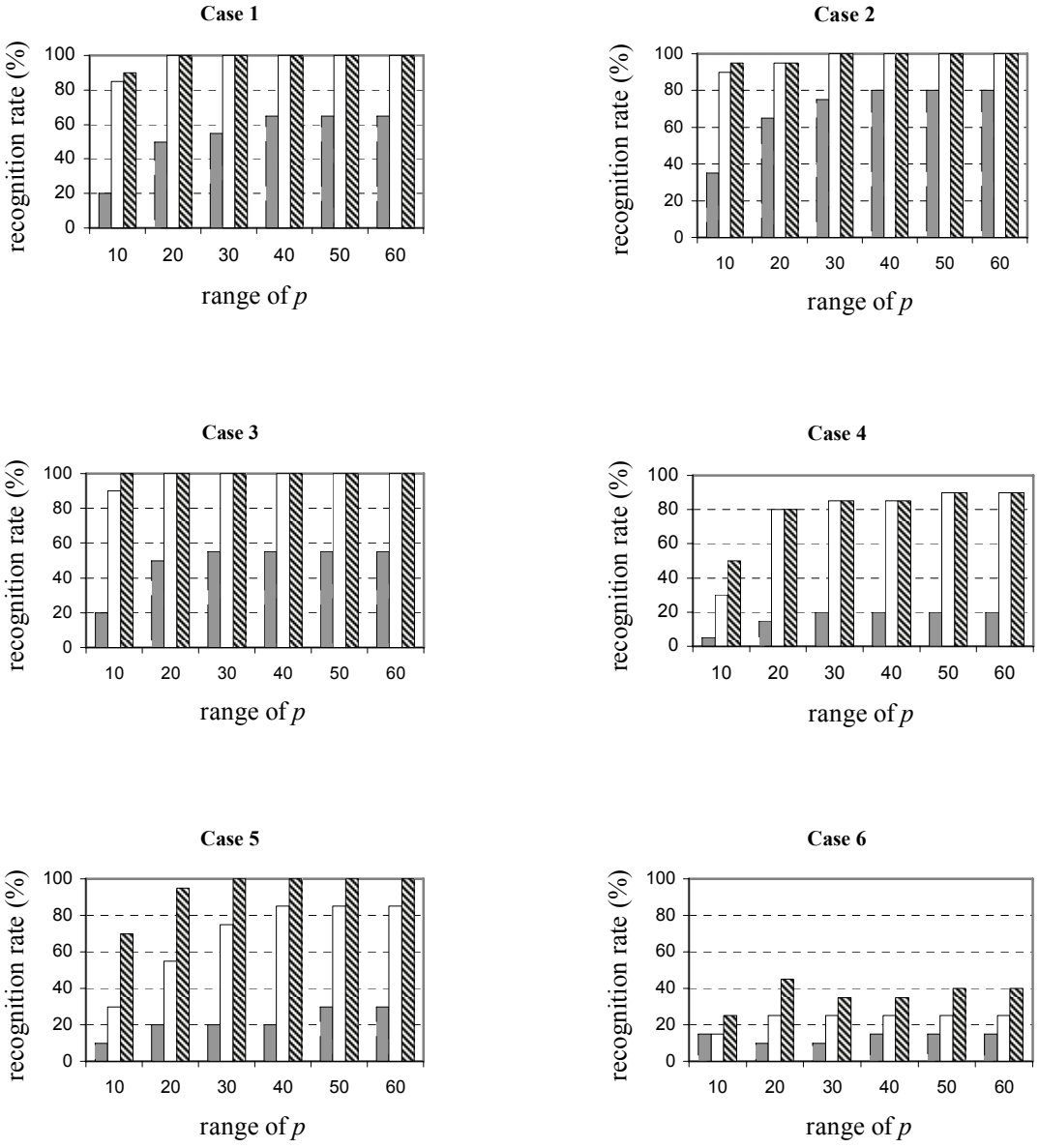


Figure 4.4. Comparison results for PCA (solid grey bar), 2DPCA (solid white bar) and E-2DPCA (diagonal stripped bar), for the six cases from the extended Yale Face Database B.

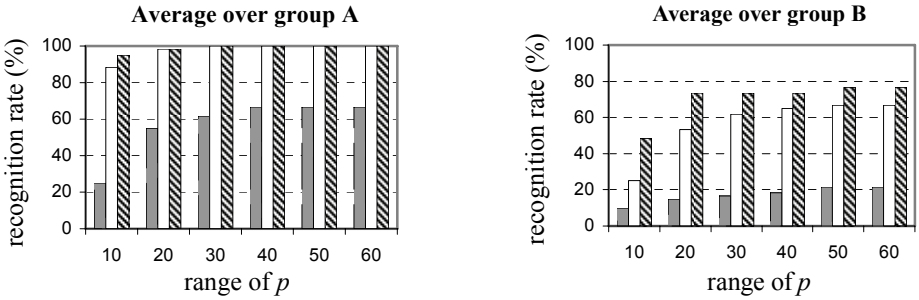


Figure 4.5. Averaged experimental results for the extended Yale Face Database B.

We note that the three algorithms evaluated did not work well for Case 6 because the test image in this case contained very large shadowed regions.

In Figure 4.5, the plot on the left shows the recognition rate averaged over group A, while the plot on the right shows the recognition rate averaged over group B. It is observed that the performance of E-2DPCA is more robust than that of conventional PCA and 2DPCA against large illumination variations.

Concerning the complexity of the PCA, 2DPCA and E-2DPCA algorithms, we note that, generally, a face recognition process may include two types of operations: offline operations and real time operations. If the training set is promptly updated with the newly identified test images, then all operations must be done in real time. The face identification task by itself consists of finding the image representation (which is given by the pattern vector for PCA, by the feature matrix for 2DPCA, and by the pair of feature matrices for E-2DPCA) and classifying it based on evaluation of the minimum distance between the test image representation and the K classes representations. Typically, the identification task does not include offline operations, such as computing the covariance matrix/matrices, finding the eigenvectors, and obtaining the class pattern vectors (for PCA)/class average feature matrices (for 2DPCA)/pair of class average feature matrices (for E-2DPCA).

The second column of Table 4.1 shows the number of multiplications required for computing all eigenvectors for a training set of M images of size $N \times N$ for the three algorithms being evaluated, while the third column of the table shows the number of multiplications required for finding the pattern vector (in the case of PCA), the feature matrix (in the case of 2DPCA) and the pair of feature matrices (in the case of E-2DPCA)

Table 4.1. Computational complexity in terms of the number of multiplications for the three algorithms

Algorithm	Evaluation of eigenvectors	Image representation using p eigenvectors
PCA	$\frac{M(M+1)}{2}N^2 + M^3 + N^2M + N^2$	$N^2 p$
2DPCA	$\frac{N^2(N+1)}{2}M + N^3$	$N^2 p$
E-2DPCA	$2\left(\frac{N^2(N+1)}{2}M + N^3\right)$	$2N^2 p$

Table 4.2. Normalized elapsed time for the three algorithms

Algorithm	Entire face recognition process	Face identification only
PCA	1	1
2DPCA	0.46	0.93
E-2DPCA	0.97	1.79

for a test image using p eigenvectors. We note that if the number M of images from training set is much larger than size N, and if all the operations are done in real time, then the 2DPCA and E-2DPCA algorithms are more efficient than the standard PCA.

An alternative and perhaps more realistic complexity measure is the normalized elapsed time. The elapsed time requires by a given algorithm was normalized to the elapsed time taken by the conventional PCA to perform the entire face recognition process and also for only face identification task. Table 4.2 summarizes the average normalized elapsed time over 100 trials of the three algorithms, where the six cases from the extended Yale Face Database B were tested, meaning that $M = 400$ and $N = 168$. The original 2DPCA algorithm was found to be the fastest when entire process of face recognition was evaluated, followed by the E-2DPCA and PCA – both required comparable execution times.

We remark that with $M = 400$, $N = 168$ and $p = 20$ which represents 5% of the available number of eigenvectors, the numerical values obtained using the formulas in Table 4.1 (after normalization) were found to be [1 0.41 0.82]' for the evaluation of

eigenvectors and [1 1 2]' for image representation. On comparing these values with those shown in Table 4.2, we see that the theoretical prediction of the computational complexity has a good match with the recorded time for the three algorithms.

4.4.2 Robustness to Noise and Face Occlusions

The extended Yale Face Database B was utilized as the data set for evaluating the robustness of E-2DPCA algorithm to noise and face occlusions.

A. Results for Noise-Contaminated Images

In order to evaluate the de-noising capabilities of the WT – TV pre-processing technique used in conjunction with E-2DPCA algorithm and the robustness of the E-2DPCA algorithm against noise, the test images from the six cases of the extended Yale Face Database B were modified by adding zero-mean Gaussian white noise with various amount of variance.

Figure 4.6 shows the performance of E-2DPCA in three scenarios: (a) no noise was applied to the test images; (b) noise was applied to test images, but no de-noising was performed; and (c) noise was applied to test images followed by WT – TV de-noising. As can be seen from the figure, to a large extent, the E-2DPCA algorithm was able to tolerate the noise. In effect, for the first three cases with mild light variations, the algorithm's performance remains unchanged in the presence of noise. For more difficult Cases 4, 5 and 6, there is 5% drop in recognition rates for a total of four instances.

The WT – TV de-noising pre-processing was found not as effective for the E-2DPCA algorithm, producing results that are practically the same as those obtained by the E-2DPCA algorithm without de-noising pre-processing.

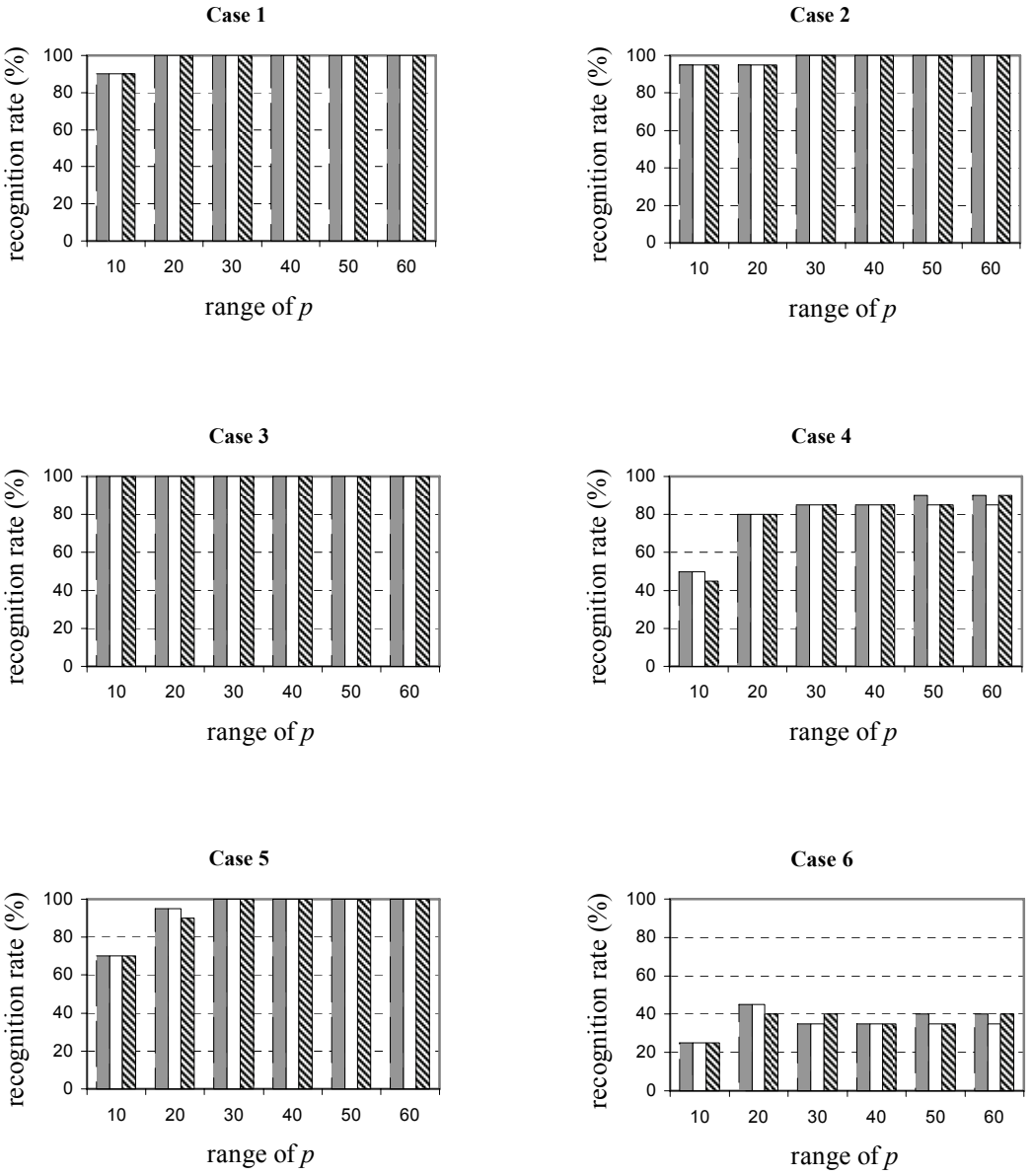


Figure 4.6. Comparison results for E-2DPCA (solid grey bar), E-2DPCA with noisy test images and no de-noising (solid white bar), and E-2DPCA with noisy test images and WT – TV de-noising (diagonal stripped bar) using the extended Yale Face Database B.

B. Results for Images with Face Occlusions

The ability of the E-2DPCA algorithm to deal with facial occlusion was examined for the same six cases of extended Yale face database B. The assumption that all testing images have occlusions was made. Results of combining the OCCL pre-processing module with E-2DPCA algorithm for eyes occlusion (solid white bar) and chin occlusion

(diagonal stripped bar) are presented in Figure 4.7. For more challenging Cases 4 – 6, we note that face occlusions have a great impact on recognition capability of OCCL E-2DPCA algorithm, drastically deteriorating its performance.

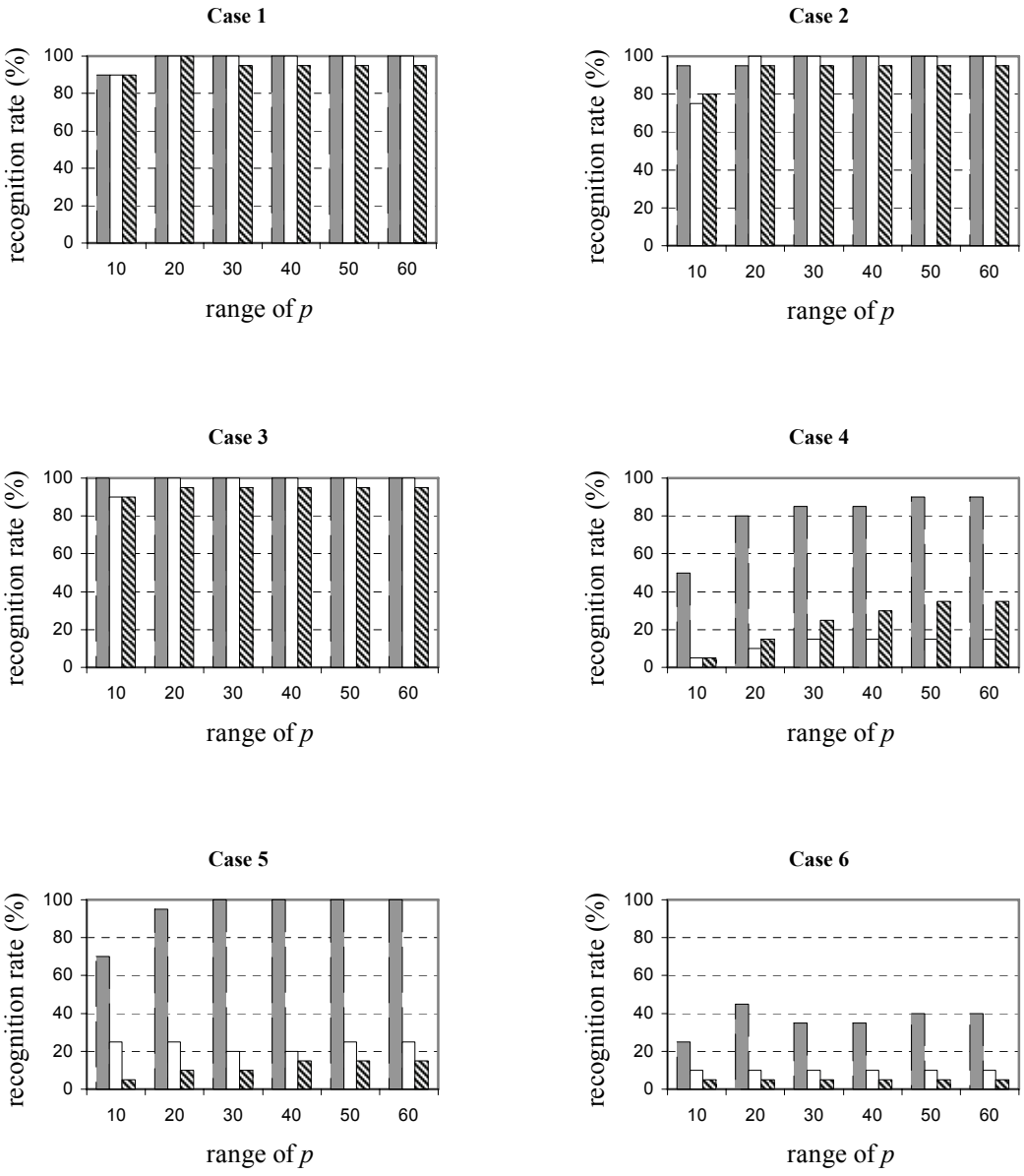


Figure 4.7. Comparison results for E-2DPCA (solid grey bar), OCCL E-2DPCA with eyes occlusion (solid white bar), and OCCL E-2DPCA with chin occlusion (diagonal stripped bar) using the extended Yale Face Database B.

4.4.3 Implementation Issues

- In E-2DPCA, the column-related and row-related processing may be sequentially performed. To reduce the computational time, a hardware optimization can be employed by using separate specialized processors to run the two operations in parallel.
- Contrary to the 1-D PCA algorithm whose implementation may encounter memory limitations if the dataset contains many large images, the 2DPCA and E-2DPCA algorithms can handle considerably larger images. The 32-bit version of MATLAB, for example, can process matrices in double precision with about $155 \cdot 10^6$ elements [109]. If the data set contains 1000 images of size 400×400, the data matrix A (see Section 2.2.3) in the 1-D PCA algorithm is of size 160000×1000 that exceeds the size limit and the software will fail to work yielding the error "out of memory". On the other hand, the covariance matrices encountered in the 2DPCA as well as E-2DPCA algorithms remain to be of size 400×400 (see (4.1) and (4.4)), a size far smaller than the limit.

4.5 Summary

Based on an analysis of the 2DPCA algorithm in [100], an extended 2DPCA technique has been proposed. Experimental results have been presented to demonstrate that the proposed algorithm can be used to improve face recognition rate with enhanced robustness against noise and illumination variations across testing images.

Chapter 5

Face Recognition Using Sparse Representation

5.1 Introduction

Sparse representations are representations that account for most or all information of a signal with a linear combination of a small number of elementary signals called atoms. These atoms are chosen from an over-complete dictionary which is a collection of atoms with the number of atoms exceeding the dimension of the signal space. As a result, any signal can be represented by more than one combination of different atoms [47].

Sparsity is one of the characteristics of popular transforms such as the DFT, DCT, DWT and the singular value decomposition (SVD) [47]. The aim of these transforms in this regard is to reveal certain structures of a signal and to represent these structures in a compact and sparse representation. In the past several years, sparse representations of signals have become increasingly popular as it is found useful in a wide variety of applications such as noise reduction, compression, feature extraction, pattern classification and blind source separation (see [31], [32], [59], [74] and [91]). Recent studies have shown that sparsity is also encountered at the level of human vision system (HVS). It was revealed that, in both low-level and mid-level human vision, many neurons in the visual pathway are selective for a variety of specific stimuli, such as color, texture,

orientation, scale, and even view-tuned object images. Considering these neurons as atoms that form an overcomplete dictionary at each visual stage, the response of the neurons with respect to a given input image is typically highly sparse [97].

In the statistical signal processing community, the problem of computing sparse linear representations with respect to an overcomplete dictionary of signal atoms has seen a recent surge of interest [97]. Much of this excitement centers on the discovery that whenever the optimal representation is sufficiently sparse, it can be efficiently computed by convex optimization [23].

This chapter is organized as follows. Section 2 gives an overview of two methods for sparse representations that are closely related to the development presented in subsequent sections. These two methods are the sparse representation technique proposed by Wright et al. in [97] and the sparse PCA algorithm proposed by Johnstone and Lu in [50]. In Section 3, we propose an enhanced sparse PCA (E-sparse PCA) and apply it to face recognition problems, and in Section 4 we present a combined E-sparse SRC technique. Performance evaluations of the proposed techniques, comparison results and several implementation issues are presented in Section 5. Section 6 summarizes the improvements of proposed algorithms over the existing methods.

5.2 An Overview of Sparse Representation Algorithms

In this section, we give an overview of two recent works on sparse representation of signals. The first one is a method by Wright et al. proposed in [97]. Their method is specifically designed to deal with robust face recognition, based on recently developed theory on sparse signal representation. The second one is a method called sparse PCA which is proposed in [50] being evaluated for feature extraction and representation of ECG data.

5.2.1 Face Recognition via Sparse Representation

The work in [97] addresses the problem of face recognition based on the sparse signal representation theory [22], [23]. The innovation of the approach in [97] lies in constructing the overcomplete dictionary especially for face recognition purpose. Unlike in a typical signal processing scenario where a generic dictionary composed of bases such

as DFT, DCT, or DWT is used [15], the dictionary employed in [97] uses the facial images of the training set as its atoms. Moreover, instead of using sparsity to identify a relevant model or relevant features to be used for classifying test samples, the method uses the sparse representation of each individual test sample directly for classification, adaptively selecting the training samples that give the most compact representation.

A basic problem in object recognition is to use labeled training samples from k distinct object classes to correctly determine the class to which a new test sample belongs. In the context of face recognition, in [97], a grayscale image of size $w \times h$ is transformed into a vector v of length $m = wh$ by stacking its columns. Similarly, the n_i training face images from the i^{th} class (subject) are transformed and arranged as columns of matrix $A_i = [v_{i,1}, v_{i,2}, ..., v_{i,n_i}]$ of size $m \times n_i$.

For the entire training set, a matrix A of size $m \times n$ with $n = \sum_{i=1}^{k} n_i$ is defined as the concatenation of the n training images of all k classes, namely

$$A = [A_1 \;\; A_2 \;\; ... \;\; A_k] = [v_{1,1} \;\; v_{1,2} \;\; ... \;\; v_{k,n_k}]. \tag{5.1}$$

The matrix A so constructed is called a dictionary because its atoms (columns) are used as basic "vocabulary" to express the "content" of a facial image.

Given a new test signal y, its linear representation within dictionary A can be written in terms of all training samples as

$$y = Ax \tag{5.2}$$

where x is a vector representing signal y. In the case of $n > m$, (5.2) is a linear system with the number of unknowns greater than the number of equations. Consequently, there are infinitely many solutions to (5.2) (here we have assumed rank(A) = m). Among these solutions, we are interested in a solution with fewest non-zero components. An n-dimensional vector x is said to be sparse if it has s non-zero components with $s \ll n$. To see the relevance of a sparse solution to the face recognition problem, let us consider the following scenarios. First, if image y happens to be one of the training images, say $v_{i,j}$ in the i^{th} class, then it is obvious that

$$x = [0 \;\; ... \;\; 0 \;\; 1 \;\; 0 \;\; ... \;\; 0]^T$$
$$\uparrow$$
$$\text{the } (n_1 + n_2 + ... + n_{i-1} + j)^{\text{th}} \text{ entry}$$

with only one non-zero component in the $(n_1 + n_2 + ... + n_{i-1} + j)^{\text{th}}$ position is the sparsest solution of (5.2). Second, if image *y* is not found in the training set but it is an image of the i^{th} subject (class) who has a total of n_i images in the training set, then *y* is expected to be expressed as a linear combination of atoms $\{v_{i,1}, v_{i,2}, \ldots, v_{i,n}\}$, which corresponds to a solution *x* of the form

$$x = [0 \ \ldots \ 0 \ \ \alpha_{i,1} \ \ \alpha_{i,2} \ \ \ldots \ \ \alpha_{i,n_i} \ \ 0 \ \ldots \ 0]^T$$

where $\alpha_{i,j}$ (for $j = 1, 2, ..., n_i$) is the $(n_1 + n_2 + ... + n_{i-1} + j)^{\text{th}}$ entry of *x*. Again, this is a sparse solution as it only has n_i non-zero components.

Following the theory of compressive sensing [16], [22], [23], the sparse representation of test image *y*, denoted by $\hat{x}$, can be obtained as the solution of the l_1-minimization problem

$$\underset{x}{\text{minimize}} \ \|x\|_1 \quad \text{subject to} \quad Ax = y \tag{5.3}$$

where $\|x\|_2 = \sum_{i=1}^{n} |x_i|$ denotes the l_1-norm of vector *x*.

Ideally, the nonzero entries in the estimate $\hat{x}$ will all be associated with the columns of *A* from a single class *i*, so that the test image *y* can be assigned to that class. However, noise and modeling error may lead to small nonzero entries associated with multiple classes.

To classify facial images, a characteristic function $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ is defined to select the coefficients associated with the i^{th} class as follows. For a vector *x* of length *n*, $\delta_i(x) \in \mathbb{R}^n$ is a vector whose only nonzero entries are the entries of *x* that are associated with class *i*. Using only the coefficients associated with the i^{th} class, one can approximate a given test image *y* as $\hat{y}_i = A\delta_i(\hat{x})$. The classification of a test image *y* can be carried out based on this approximation as follows: image *y* is classified to belong to class *i* if $\|y - A\delta_i(\hat{x})\|$ reaches the minimum among $\{\|y - A\delta_j(\hat{x})\|, \ j = 1, 2, ..., k\}$, namely,

$$r_i(y) = \|y - A\delta_i(\hat{x})\| = \min_{1 \le j \le k} \|y - A\delta_j(\hat{x})\|. \tag{5.4}$$

The method described above is called the sparse representation-based classification (SRC) algorithm which can be summarized as follows.

Input: A matrix of training images $A = [A_1 \ A_2 \ ... \ A_k]$ of size $m \times n$ for k classes, and a test image y of size $m \times 1$.

Step 1: Normalize each column of A to have an unit l_2-norm.

Step 2: Solve the l_1-minimization problem in (5.3) and denote its solution by $\hat{x}$.

Step 3: Compute the residuals $r_j = \left\| y - A\delta_j(\hat{x}) \right\|_2$ for $j = 1, 2, ..., k$.

Step 4: Output the class that y belongs to, which is identified as $\arg\min_{1 \le j \le k} r_j(y)$.

For raw face images, m is very large, hence typically A is a tall matrix. For instance, if the face images have a resolution of 640×480 pixels, the dimension m is in the order of 10^5. Although SRC algorithm relies on scalable computational methods such as linear programming, directly applying it to such images is still beyond the capability of regular computers. To reduce the dimensionality of problem (5.3), the authors of [97] have proposed two alternative pre-processing steps before applying the SRC algorithm. The first one consists of downsampling the original images such that the data matrix A and the test image y become of sizes $d \times n$ and $d \times 1$, respectively, with $d \ll m$ and $d < n$ (typically $d \ll n$ as well). The second one employs a random projection matrix R of size $d \times m$ that applies to both sides of (5.2) to yield

$$Ry = RAx \tag{5.5}$$

where the projection matrix R is obtained as a randomly generated matrix whose entries are independently sampled from a zero-mean normal distribution with each row normalized to unit length. Under these circumstances, the reduced l_1-minimization problem becomes

$$\underset{x}{\text{minimize}} \ \|x\|_1 \quad \text{subject to} \quad RAx = \tilde{y} \tag{5.6}$$

where $\tilde{y} = Ry$. We remark that the rows of R can be viewed as d random faces [97] in $\mathbb{R}^m$, consequently, the *random features* of the transformed matrix RA can be viewed as a less-structured counterpart to classical face features such as *eigenfaces* in the standard PCA.

5.2.2 Sparse PCA

As we have seen in Section 2.2.3, the PCA is a classical method for dimensionality reduction of data in the form of *n* observations of a vector with *m* variables. A typical data set has a structure with *m* much greater than *n*. In [50], Johnstone and Lu propose a method for a preliminary reduction in dimensionality before the PCA algorithm is applied to the data, called sparse PCA. It is shown that this preliminary reduction in dimensionality can be achieved by working with a basis in which signals have sparse representations.

It has been known that there exist orthonormal bases under which general signals are represented with sparse or near sparse coefficients. Well-known orthonormal bases of this nature include Daubechies' orthogonal wavelet transforms and discrete cosine transform (DCT). In what follows we call such a transform a *sparsifying transform*.

Let $A = [\ a_1\ \ a_2\ \ \ldots\ \ a_n]$ be a data matrix of size $m \times n$, where a_i denotes the i^{th} observation of a vector with *m* variables. The first step of the sparse PCA algorithm is to convert the data set A into an equivalent but sparse (or near sparse) data set $\tilde{A} = [\tilde{a}_1\ \ \tilde{a}_2\ \ \ldots\ \ \tilde{a}_n]$ where

$$\tilde{a}_j = \mathcal{T} a_j \qquad \text{for } j = 1, 2, \ldots, n \tag{5.7}$$

i.e., $\tilde{A} = \mathcal{T} A$ with $\mathcal{T}$ a sparsifying transform. The second step of the sparse PCA algorithm computes the variances of each row of data set $\tilde{A}$, namely

$$\sigma_i^2 = \text{variance of } \tilde{A}(i, 1:n) \quad \text{for } i = 1, 2, \ldots, m \tag{5.8}$$

and select those rows of $\tilde{A}$ with *d* largest variance to construct a reduced data set $\tilde{A}_d$ of size $d \times n$

$$\tilde{A}_d = \tilde{A}(\text{ind}(1:d), 1:n) \tag{5.9}$$

where 'ind' are the indices of the rows of $\tilde{A}$ whose variances are arranged in descending order.

Finally, the standard PCA is applied to the reduced data set $\tilde{A}_d$.

In [50], wavelet transforms are used as sparsifying transforms and the algorithm is applied to ECG data to extract features of interest.

5.3 An Extended Sparse PCA for Face Recognition

The dimensionality reduction in the sparse PCA algorithm [50] is performed based on the variance of each variable over its n observations which is related to the second-order moment of the variable. A question arising naturally is whether or not the other moments of the variables can also play a role here. In what follows we describe a sparse PCA algorithm which takes both the first-order and second-order moments of the variables into account. Let A and y be the set of training facial images and test image, respectively, where each image is represented by a column vector, hence $A \in \mathbb{R}^{m\times n}$ and $y \in \mathbb{R}^{m\times 1}$. As in [50], the evaluation of the moments is applied to data set $\tilde{A} = \mathcal{T}A$ and test image $\tilde{y} = \mathcal{T}y$, where $\mathcal{T}$ is a sparsifying transform of choice. The basic idea of the proposed method is that for a target dimension d, we would first pick more than d (say, $\tilde{d}$) rows of $\tilde{A}$ with greatest variances and then choose among them d rows with greatest mean values. In doing so, each of the $\tilde{d}$ rows of $\tilde{A}$ so chosen has both significant variance (hence capturing important changes across the observations) and significant mean values (hence it is a major component across the observations). The proposed method can be summarized as the steps given below.

Input: A set of training images $A = [A_1\ A_2\ \ldots\ A_k]$ of size $m\times n$ for k classes with $n = kL$; a test image y of size $m\times 1$; a target dimension d; a positive scalar γ typically in the range of [0, 3].

Step 1: Apply a sparsifying transform $\mathcal{T}$ to A and y (as shown in (5.7)) to get

$$\tilde{A} = \mathcal{T}A \quad \text{and} \quad \tilde{y} = \mathcal{T}y\,. \tag{5.10}$$

Step 2: Select $\tilde{d} = \text{round}((1+\gamma)d)$ rows of $\tilde{A}$ with greatest variances. Let 'ind_1' be the set of indices of such rows in $\tilde{A}$. Construct

$$\tilde{A}_{\tilde{d}} = \tilde{A}(\text{ind}_1, 1:n) \quad \text{and} \quad \tilde{y}_{\tilde{d}} = \tilde{y}(ind_1)\,. \tag{5.11}$$

Step 3: Select d rows of $\tilde{A}_{\tilde{d}}$ with greatest mean values. Let 'ind_2' be the set of indices of such rows in $\tilde{A}_{\tilde{d}}$. Construct

$$\hat{A}_d = \tilde{A}_{\tilde{d}}(\text{ind}_2, 1:n) \quad \text{and} \quad \hat{y}_d = \tilde{y}_{\tilde{d}}(ind_2)\,. \tag{5.12}$$

Step 4: Apply the standard PCA to data set $\{\hat{A}_d, \hat{y}_d\}$.

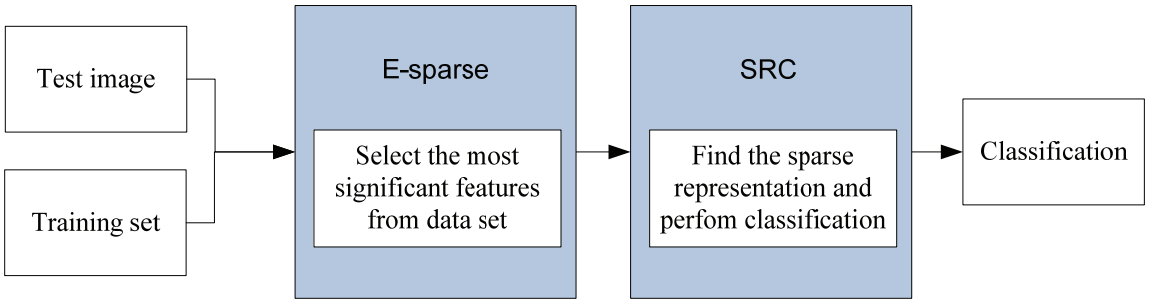


Figure 5.1. A block diagram of E-sparse SRC algorithm.

We remark that with parameter γ set to zero, the proposed algorithm becomes identical to the sparse PCA algorithm in [50]. In other words, the sparse PCA algorithm may be considered as a special case of the above method and, in this regard, we call the above an extended sparse (E-sparse) PCA algorithm.

5.4 E-Sparse SRC – a Combined Technique for Performance Enhancement

The sparse representation-based classification (SRC) algorithm reviewed in Section 5.2.1 was found to offer good performance [97]. However, the method encounters large-scale data even for facial images of moderate size. On the other hand, the E-sparse PCA (and consequently the sparse PCA) provides a preliminary dimensionality reduction as it reduces the data size from $m \times n$ to $d \times n$ with $d \ll m$ while retaining most significant features of the original data set. It is therefore natural to develop a face recognition system that combines the E-sparse pre-processing module for preliminary dimensionality reduction (see Steps 1 to 3 from algorithm illustrated in Section 5.3) with the SRC algorithm (see Section 5.2.1) for performance enhancement of face classification. The combined algorithm will be referred to as the E-sparse SRC algorithm and is illustrated by a block diagram in Figure 5.1.

5.5 Experimental Results

Several data sets have been utilized in the simulations described below. For PCA-based algorithms, such as sparse and E-sparse PCA, the ten cases from the Yale Face

Database sufficed (see Table 3.2 and Appendix 1). For simulations in a sparse representation framework, data sets of much larger scales are needed. To cope with different experimental scenarios, four new data sets have been constructed based on the extended Yale Face Database B (see Table 5.4 and Appendix 2).

5.5.1 Performance Comparisons

A. E-Sparse PCA

Employing the data for the ten cases from the Yale Face Database (see Table 3.2), the performance of E-sparse PCA algorithm with various choices of parameters d and γ and basis for sparse representation was examined. The standard PCA algorithm without a E-sparse preliminary dimensionality reduction module was used as a reference point.

As mentioned earlier, the sparse PCA applied to a face data set for face recognition purpose can be considered as a particular case of E-sparse PCA (with γ set to zero). The other parameters were set as follows:

- parameter d, which defined the dimensionality of the reduced data set, was set to $d =$ 50, 100, 150 and 200, respectively;
- the number of eigenfaces to be used was set to p = 5, 7, 9, 11, 13 and 15, respectively;
- for the E-sparse pre-processing, the 1-D DCT, 2-D DCT, 1-D DWT and 2-D DWT were employed. For the DWT cases, up to $L = 4$ levels of decomposition were simulated.

Table 5.1 illustrates the comparison results obtained for Case 1 using the standard PCA with and without preliminary sparse pre-processing.

The results included in Table 5.2 were obtained by selecting only the best results using E-sparse PCA and comparing them with the results obtained by the standardPCA, for all ten cases. It was observed that when 1-D DWT was employed for E-sparse PCA with an $\gamma \in (0,3]$, performance improvement over the original sparse PCA (i.e., with $\gamma = 0$) was achieved. Nevertheless, as illustrated in Table 5.2, the best performances were attained by using 2-D transforms, such as 2-D DWT and 2-D DCT.

Table 5.1. Comparison results for PCA and sparse PCA for Case 1 from the Yale Face Database

d	Eigenfaces / Sparse PCA	5	7	9	11	13	15
	Standard PCA	80.0	93.3	93.3	93.3	93.3	100.0
50	1-D DCT	60.0	86.7	93.3	93.3	93.3	100.0
	2-D DCT	73.3	80.0	93.3	93.3	100.0	100.0
	1-D DWT (L = 1)	26.7	33.3	33.3	33.3	33.3	33.3
	1-D DWT (L = 2)	40.0	40.0	40.0	40.0	40.0	40.0
	1-D DWT (L = 3)	53.3	53.3	53.3	53.3	53.3	53.3
	1-D DWT (L = 4)	66.7	66.7	66.7	66.7	66.7	66.7
	2-D DWT (L = 1)	53.3	53.3	53.3	53.3	53.3	53.3
	2-D DWT (L = 2)	66.7	66.7	66.7	73.3	73.3	73.3
	2-D DWT (L = 3)	86.7	86.7	86.7	93.3	86.7	93.3
	2-D DWT (L = 4)	73.3	73.3	86.7	86.7	86.7	86.7
100	1-D DCT	60.0	86.7	93.3	93.3	93.3	100.0
	2-D DCT	80.0	93.3	93.3	93.3	93.3	100.0
	1-D DWT (L = 1)	53.3	53.3	53.3	53.3	53.3	53.3
	1-D DWT (L = 2)	40.0	46.7	46.7	46.7	46.7	46.7
	1-D DWT (L = 3)	66.7	73.3	73.3	73.3	73.3	73.3
	1-D DWT (L = 4)	66.7	66.7	66.7	66.7	66.7	66.7
	2-D DWT (L = 1)	53.3	53.3	60.0	53.3	60.0	60.0
	2-D DWT (L = 2)	66.7	73.3	73.3	73.3	73.3	80.0
	2-D DWT (L = 3)	86.7	100.0	100.0	100.0	100.0	100.0
	2-D DWT (L = 4)	80.0	93.3	93.3	93.3	93.3	100.0
150	1-D DCT	66.7	93.3	93.3	93.3	93.3	100.0
	2-D DCT	80.0	93.3	93.3	93.3	100.0	100.0
	1-D DWT (L = 1)	53.3	53.3	53.3	53.3	53.3	53.3
	1-D DWT (L = 2)	60.0	66.7	66.7	66.7	66.7	66.7
	1-D DWT (L = 3)	66.7	73.3	73.3	73.3	73.3	73.3
	1-D DWT (L = 4)	73.3	73.3	73.3	80.0	80.0	80.0
	2-D DWT (L = 1)	60.0	66.7	66.7	60.0	66.7	60.0
	2-D DWT (L = 2)	73.3	80.0	86.7	93.3	86.7	93.3
	2-D DWT (L = 3)	86.7	93.3	100.0	100.0	100.0	100.0
	2-D DWT (L = 4)	80.0	93.3	93.3	93.3	93.3	100.0
200	1-D DCT	73.3	93.3	93.3	93.3	93.3	100.0
	2-D DCT	80.0	93.3	93.3	93.3	93.3	100.0
	1-D DWT (L = 1)	53.3	53.3	60.0	53.3	53.3	53.3
	1-D DWT (L = 2)	66.7	66.7	73.3	73.3	73.3	73.3
	1-D DWT (L = 3)	66.7	73.3	73.3	73.3	73.3	73.3
	1-D DWT (L = 4)	73.3	80.0	86.7	86.7	86.7	86.7
	2-D DWT (L = 1)	66.7	66.7	66.7	66.7	73.3	73.3
	2-D DWT (L = 2)	80.0	86.7	86.7	93.3	93.3	93.3
	2-D DWT (L = 3)	86.7	93.3	93.3	93.3	100.0	100.0
	2-D DWT (L = 4)	80.0	93.3	93.3	100.0	93.3	100.0

Table 5.2. Results for E-sparse PCA using the ten cases from the Yale Face Database

Case	Parameters (d, γ)	Eigenfaces / E-sparse PCA	5	7	9	11	13	15
1		Standard PCA	80.0	93.3	93.3	93.3	93.3	100.0
	(100, 0)	2-D DWT (L = 3)	86.7	100.0	100.0	100.0	100.0	100.0
2		Standard PCA	46.7	53.3	73.3	73.3	80.0	80.0
	(150, 0)	2-D DWT (L = 2)	60.0	53.3	80.0	80.0	80.0	80.0
3		Standard PCA	53.3	80.0	80.0	86.7	86.7	86.7
	(150, 0)	2-D DWT (L = 2)	73.3	86.7	86.7	86.7	93.3	93.3
4		Standard PCA	33.3	40.0	40.0	53.3	60.0	60.0
	(100, 0.5)	1-D DWT (L = 3)	33.3	40.0	53.3	53.3	60.0	60.0
5		Standard PCA	20.0	46.7	60.0	80.0	73.3	80.0
	(200, 0.3)	2-D DWT (L = 4)	26.7	46.7	66.7	80.0	80.0	80.0
6		Standard PCA	13.3	33.3	53.3	60.0	66.7	73.3
	(150, 0)	2-D DWT (L = 4)	13.3	33.3	53.3	60.0	66.7	66.7
7		Standard PCA	33.3	60.0	73.3	73.3	93.3	86.7
	(200, 0)	2-D DCT	33.3	60.0	66.7	73.3	80.0	80.0
8		Standard PCA	26.7	33.3	66.7	66.7	66.7	66.7
	(100, 0)	2-D DCT	26.7	33.3	66.7	66.7	66.7	66.7
9		Standard PCA	20.0	33.3	40.0	53.3	60.0	60.0
	(100, 0)	2-D DCT	20.0	33.3	46.7	46.7	53.3	53.3
10		Standard PCA	33.3	46.7	53.3	73.3	80.0	80.0
	(100, 0)	2-D DWT (L = 4)	33.3	40.0	60.0	73.3	80.0	80.0

The results included in Table 5.3, compare the standard PCA's performance with that of E-sparse PCA where 2-D DCT was used as the sparsifying transform and the target dimension was fixed to d = 100, which represents a dimensionality reduction ratio of 163.84 of the original size of face images from the ten cases of Yale Face Database. We remark that only the first two best sets of results for E-sparse PCA were included. Inspecting the results for all ten cases, it was noticed that E-sparse PCA with 2-D DCT, d = 100, r = 0, provides practically the same results as the standard PCA algorithm applied to the original data set. This demonstrates that the sparse and E-sparse module can be used as a preliminary dimensionality reduction step prior to the use of standard PCA without substantial performance degradation.

Finally, the complexity of E-sparse PCA algorithm was examined in terms of elapsed time. The results were averaged over the ten cases from Yale Face Database and six

values of eigenfaces considered for each case. It was found that, the time required by E-sparse PCA to produce the eigenfaces was 5.37 times less than that required by the standard PCA, while as the time required by the E-sparse PCA to classify the input image was a small fraction (1/107.86) of that required by the standard PCA.

Table 5.3. Comparison results for PCA and E-sparse PCA (with 2-D DCT and $d = 100$) for the ten cases from the Yale Face Database

Case	Eigenfaces / E-sparse PCA	5	7	9	11	13	15
1	Standard PCA	80.0	93.3	93.3	93.3	93.3	100.0
	$\gamma = 0$	80.0	93.3	93.3	93.3	93.3	100.0
	$\gamma = 0.3$	73.3	73.3	80.0	80.0	100.0	100.0
2	Standard PCA	46.7	53.3	73.3	73.3	80.0	80.0
	$\gamma = 0$	46.7	60.0	73.3	73.3	73.3	73.3
	$\gamma = 0.3$	33.3	53.3	60.0	66.7	66.7	66.7
3	Standard PCA	53.3	80.0	80.0	86.7	86.7	86.7
	$\gamma = 0$	60.0	66.7	80.0	86.7	86.7	86.7
	$\gamma = 0.5$	66.7	66.7	80.0	80.0	80.0	86.7
4	Standard PCA	33.3	40.0	40.0	53.3	60.0	60.0
	$\gamma = 0$	40.0	40.0	40.0	53.3	53.3	60.0
	$\gamma = 0.1$	33.3	33.3	40.0	53.3	60.0	53.3
5	Standard PCA	20.0	46.7	60.0	80.0	73.3	80.0
	$\gamma = 0$	26.7	46.7	60.0	73.3	73.3	73.3
	$\gamma = 0.3$	33.3	46.7	60.0	60.0	60.0	60.0
6	Standard PCA	13.3	33.3	53.3	60.0	66.7	73.3
	$\gamma = 0.5$	33.3	53.3	60.0	60.0	60.0	60.0
	$\gamma = 3.0$	40.0	53.3	60.0	53.3	60.0	60.0
7	Standard PCA	33.3	60.0	73.3	73.3	93.3	86.7
	$\gamma = 0$	33.3	60.0	73.3	73.3	73.3	73.3
	$\gamma = 0$	33.3	53.3	60.0	60.0	60.0	60.0
8	Standard PCA	26.7	33.3	66.7	66.7	66.7	66.7
	$\gamma = 0$	26.7	33.3	66.7	66.7	66.7	66.7
	$\gamma = 1.5$	46.7	46.7	53.3	53.3	53.3	53.3
9	Standard PCA	20.0	33.3	40.0	53.3	60.0	60.0
	$\gamma = 0$	20.0	33.3	46.7	46.7	53.3	53.3
	$\gamma = 0.5$	20.0	40.0	46.7	46.7	46.7	53.3
10	Standard PCA	33.3	46.7	53.3	73.3	80.0	80.0
	$\gamma = 0$	33.3	40.0	53.3	73.3	80.0	80.0
	$\gamma = 0.3$	26.7	53.3	60.0	66.7	66.7	66.7

B. E-sparse SRC

For this part of experimental studies, four new data sets were constructed using 1280 facial images of size 168×168, representing 20 individuals, each with 64 images in various light conditions, from the extended Yale Face Database B. A description of these data sets is given in Table 5.4. For more details about the images contained in the data sets, the reader is referred to Appendix 2.

The experimental results are given in terms of recognition rate for each class (i.e., individual).

Table 5.4. Four sets from the extended Yale Face Database B

Set	Training set	Testing set	Remark
1	All 20 individuals with 54 poses	All 20 individuals with remaining 10 poses	A data set considered as *easy* to deal with as its testing set consists of images with small shadowed areas, while the training set contains most of poses.
2	All 20 individuals with 54 poses	All 20 individuals with remaining 10 poses	A data set with *medium* level of difficulty. The testing set contains images with large shadowed regions, while training set contains most of poses.
3	All 20 individuals with 32 poses	All 20 individuals with remaining 32 poses	A data set with *medium* level of difficulty as its training set contains relatively less number of poses.
4	All 20 individuals with 44 poses	All 20 individuals with the remaining 20 poses	A data set considered as *hard* to deal with as its training set contains only images with small or medium shadowed regions, while the testing set contains images with large shadowed regions.

Table 5.5. Results for D-SRC for the four data sets of the extended Yale Face Database B

	D-SRC (d = 100)			
Individual	Set 1	Set 2	Set 3	Set 4
1	100.0	50.0	78.1	50.0
2	90.0	60.0	75.0	35.0
3	100.0	50.0	75.0	75.0
4	100.0	100.0	65.6	60.0
5	100.0	60.0	62.5	55.0
6	100.0	50.0	59.4	60.0
7	100.0	60.0	46.9	80.0
8	90.0	80.0	68.8	80.0
9	100.0	70.0	87.5	80.0
10	100.0	80.0	84.4	65.0
11	100.0	90.0	68.8	60.0
12	100.0	90.0	68.8	50.0
13	100.0	70.0	56.3	55.0
14	100.0	60.0	43.8	55.0
15	100.0	70.0	75.0	50.0
16	100.0	80.0	56.3	85.0
17	100.0	80.0	78.1	80.0
18	100.0	80.0	59.4	80.0
19	100.0	60.0	65.6	55.0
20	100.0	80.0	78.1	85.0
Average	**99.0**	**71.0**	**67.7**	**64.8**
Overall	**75.6**			

First, the performance of the SRC algorithm using downsampled images (referred as D-SRC) was evaluated. Face images were downsampled by 17 to size 10×10 so that the reduced dimension d was equal to 100. The results are given in Table 5.5. From the table, we notice that the average recognition rate per set decreases as the level of difficulty of data increases. An overall recognition rate of 75.6% was obtained for the four data sets considered.

Next, the SRC algorithm with a random projection pre-processing step (referred as R-SRC) was evaluated, and the effect of using different random matrices R (generated by different seeds) on its performance was examined. Table 5.6 presents the results obtained when d was set to 100 and different random matrices (generated by initial states 17 and 78, respectively) were employed. We note that the use of a different random projection matrix led to only 1.6% of variation in the overall performance, but larger variations occurred in the average recognition rates, up to 5.5% for Set 2.

Table 5.6. Results for R-SRC (with different random matrices *R*) for the four data sets of the extended Yale Face Database B

	R-SRC (d = 100, seed = 17)				R-SRC (d = 100, seed = 78)			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	100.0	80.0	75.0	75.0	100.0	60.0	65.6	70.0
2	100.0	90.0	59.4	75.0	100.0	70.0	75.0	65.0
3	100.0	70.0	59.4	65.0	100.0	80.0	62.5	65.0
4	100.0	90.0	75.0	80.0	100.0	90.0	71.9	65.0
5	100.0	60.0	68.8	50.0	100.0	40.0	62.5	40.0
6	100.0	70.0	53.1	65.0	100.0	70.0	56.3	65.0
7	100.0	90.0	59.4	80.0	100.0	60.0	71.9	70.0
8	100.0	80.0	46.9	70.0	100.0	70.0	56.3	80.0
9	100.0	80.0	78.1	80.0	100.0	90.0	81.3	80.0
10	100.0	90.0	71.9	60.0	90.0	90.0	90.6	50.0
11	100.0	90.0	90.6	80.0	100.0	90.0	81.3	65.0
12	100.0	80.0	75.0	65.0	100.0	80.0	71.9	60.0
13	100.0	70.0	62.5	70.0	100.0	40.0	62.5	65.0
14	100.0	60.0	46.9	60.0	100.0	50.0	53.1	50.0
15	100.0	70.0	56.3	60.0	100.0	60.0	53.1	50.0
16	100.0	60.0	68.8	60.0	100.0	70.0	78.1	75.0
17	90.0	60.0	56.3	65.0	100.0	70.0	50.0	70.0
18	100.0	100.0	75.0	90.0	100.0	100.0	78.1	85.0
19	100.0	100.0	93.8	70.0	100.0	80.0	78.1	70.0
20	100.0	70.0	75.0	60.0	100.0	90.0	87.5	85.0
Average	**99.5**	**78.0**	**67.3**	**69.0**	**99.5**	**72.5**	**69.4**	**66.3**
Overall	**78.5**				**76.9**			

Compared with the D-SRC, the R-SRC was found to provide higher recognition rates for all data sets with an exception for Set 3, where D-SRC was 0.4% better than R-SRC when the random seed was set to 17.

Finally, the E-sparse SRC algorithm proposed in Section 5.4 was evaluated, where 2-D DCT was employed as the sparsifying transform and parameters d and γ were set to $d = 100$ and $\gamma = 0$, respectively, as this set-up was found appropriate for most cases from Yale Face Database (see Section 5.5.1 A). The recognition results are given in Table 5.7. It is observed that the average recognition rates were improved over the D-SRC and R-SRC algorithms for all sets, with an overall recognition rate of 81.8%.

Table 5.7. Results for E-sparse SRC (with 2-D DCT, $d = 100$, $\gamma = 0$) for the four data sets of the extended Yale Face Database B

	E-sparse SRC (2-D DCT, $d = 100$, $\gamma = 0$)			
Individual	Set 1	Set 2	Set 3	Set 4
1	100.0	90.0	68.8	90.0
2	100.0	100.0	84.4	70.0
3	100.0	50.0	81.3	65.0
4	100.0	90.0	68.8	75.0
5	100.0	60.0	75.0	65.0
6	100.0	90.0	65.6	70.0
7	100.0	90.0	71.9	85.0
8	100.0	100.0	65.6	80.0
9	100.0	100.0	81.3	85.0
10	100.0	90.0	78.1	70.0
11	100.0	80.0	75.0	90.0
12	100.0	70.0	78.1	60.0
13	100.0	70.0	62.5	70.0
14	100.0	60.0	37.5	55.0
15	100.0	80.0	65.6	65.0
16	100.0	50.0	56.3	65.0
17	100.0	70.0	71.9	90.0
18	100.0	90.0	62.5	85.0
19	100.0	90.0	96.9	85.0
20	100.0	90.0	90.6	75.0
Average	**100.0**	**80.5**	**71.9**	**74.8**
Overall	**81.8**			

For comparison purposes, Tables 5.8, 5.9 and 5.10 present the recognition rates obtained when E-sparse SRC algorithm was employed with $d = 100$ and $\gamma = 0$, and the sparsifying transform used was 1-D DCT, 1-D and 2-D DWT with 3 levels of decomposition, respectively. It was found that the best performance was achieved by E-sparse pre-processing employed with 2-D DCT.

Table 5.8. Results for E-sparse SRC (with 1-D DCT, $d = 100$, $\gamma = 0$) for the four data sets of the extended Yale Face Database B

	E-sparse SRC (1-D DCT, $d = 100$, $\gamma = 0$)			
Individual	Set 1	Set 2	Set 3	Set 4
1	100.0	70.0	68.8	75.0
2	100.0	100.0	87.5	70.0
3	100.0	60.0	84.4	60.0
4	100.0	90.0	75.0	75.0
5	100.0	60.0	65.6	55.0
6	100.0	70.0	65.6	60.0
7	100.0	100.0	68.8	90.0
8	100.0	100.0	62.5	90.0
9	100.0	90.0	93.8	85.0
10	100.0	80.0	84.4	80.0
11	100.0	90.0	84.4	80.0
12	100.0	50.0	75.0	55.0
13	100.0	50.0	65.6	60.0
14	100.0	60.0	43.8	55.0
15	100.0	70.0	68.8	55.0
16	100.0	50.0	50.0	50.0
17	100.0	70.0	62.5	75.0
18	100.0	100.0	71.9	65.0
19	100.0	90.0	90.6	85.0
20	100.0	90.0	84.4	65.0
Average	**100.0**	**77.0**	**72.7**	**69.3**
Overall	**79.7**			

Table 5.9. Results for E-sparse SRC (with 1-D DWT, $L = 3$, $d = 100$, $\gamma = 0$) for the four data sets of the extended Yale Face Database B

	E-sparse SRC (1-D DWT, $L = 3$, $d = 100$, $\gamma = 0$)			
Individual	Set 1	Set 2	Set 3	Set 4
1	100.0	50.0	40.6	40.0
2	100.0	40.0	34.4	5.0
3	100.0	70.0	43.8	70.0
4	100.0	70.0	34.4	60.0
5	100.0	50.0	34.4	10.0
6	100.0	70.0	34.4	30.0
7	100.0	60.0	40.6	35.0
8	100.0	40.0	46.9	50.0
9	100.0	40.0	40.6	50.0
10	100.0	50.0	31.3	15.0
11	100.0	40.0	40.6	55.0
12	100.0	50.0	78.1	15.0
13	100.0	50.0	53.1	35.0
14	100.0	30.0	31.3	10.0
15	100.0	80.0	50.0	50.0
16	100.0	60.0	46.9	20.0
17	100.0	30.0	37.5	65.0
18	100.0	80.0	37.5	20.0
19	100.0	50.0	56.3	20.0
20	100.0	10.0	34.4	25.0
Average	**100.0**	**51.0**	**42.3**	**34.0**
Overall	**56.8**			

Table 5.10. Results for E-sparse SRC (with 2-D DWT, $L = 3$, $d = 100$, $\gamma = 0$) for the four data sets of the extended Yale Face Database B

	E-sparse SRC (2-D DWT, $L = 3$, $d = 100$, $\gamma = 0$)			
Individual	Set 1	Set 2	Set 3	Set 4
1	90.0	60.0	43.8	65.0
2	100.0	60.0	59.4	50.0
3	100.0	70.0	56.3	60.0
4	90.0	70.0	56.3	75.0
5	90.0	80.0	40.6	40.0
6	70.0	60.0	46.9	50.0
7	100.0	50.0	40.6	65.0
8	100.0	50.0	59.4	45.0
9	100.0	50.0	65.6	30.0
10	90.0	40.0	71.9	25.0
11	90.0	80.0	50.0	65.0
12	70.0	40.0	43.8	30.0
13	70.0	50.0	43.8	45.0
14	100.0	40.0	31.3	30.0
15	90.0	40.0	50.0	35.0
16	100.0	50.0	56.3	70.0
17	90.0	70.0	43.8	55.0
18	90.0	50.0	46.9	50.0
19	100.0	60.0	75.0	50.0
20	100.0	50.0	68.8	45.0
Average	**91.5**	**56.0**	**52.5**	**49.0**
Overall	**62.3**			

5.5.2 Robustness to Noise and Face Occlusion

In order to evaluate the robustness of E-sparse SRC algorithm against noise, the test images of the four data sets of the extended Yale Face Database B were modified by adding zero-mean Gaussian white noise with various amounts of variance.

Table 5.11 shows the performance of the E-sparse SRC algorithm when noise was applied to test images and no de-noising was performed. On comparing the results with those given in Table 5.7, the algorithm's performance is found robust to noise for all four data sets as its overall recognition rate of 81.3% is only 0.5% lower than achieved by a clean set of test images. As a result, no de-noising was necessary for the E-sparse SRC.

Table 5.11. Results for E-sparse SRC (with 2-D DCT, $d = 100$, $\gamma = 0$) applied to noise-contaminated data for the four data sets of the extended Yale Face Database B

	E-sparse SRC : Noisy, no de-noising			
Individual	Set 1	Set 2	Set 3	Set 4
1	100.0	90.0	68.8	90.0
2	100.0	100.0	87.5	60.0
3	100.0	50.0	78.1	60.0
4	100.0	80.0	68.8	70.0
5	100.0	60.0	75.0	60.0
6	100.0	70.0	65.6	65.0
7	100.0	90.0	71.9	80.0
8	100.0	100.0	65.6	90.0
9	100.0	100.0	87.5	85.0
10	100.0	90.0	78.1	65.0
11	100.0	80.0	81.3	80.0
12	100.0	70.0	78.1	70.0
13	100.0	70.0	62.5	70.0
14	100.0	70.0	37.5	55.0
15	100.0	80.0	62.5	70.0
16	100.0	60.0	59.4	70.0
17	100.0	70.0	71.9	90.0
18	100.0	90.0	62.5	85.0
19	100.0	90.0	96.9	80.0
20	100.0	80.0	87.5	75.0
Average	**100.0**	**79.5**	**72.3**	**73.5**
Overall	**81.3**			

In addition, the ability of the E-sparse SRC algorithm to deal with facial occlusion was examined using the same four data sets. It was assumed that all testing images have occlusions. Results obtained by combining an OCCL module (see Section 3.6) with E-sparse SRC for eyes occlusion (left hand-side) and chin occlusion (right hand-side) are presented in Table 5.12. From Table 5.12 it is observed that the algorithm was able to produce satisfactory results in terms of robustness to eyes and chin occlusion for Sets 1 and 3, but suffered 8.5% and 8.3% average recognition rate drop for eyes occlusion, and 30% and 19.3% drop for chin occlusion, for Sets 2 and 4, respectively (see Table 5.7 for comparison). It is also noted that E-sparse SRC was more sensitive to chin occlusion relative to eyes occlusion.

Table 5.12. Results for OCCL E-sparse SRC (with 2-D DCT, $d = 100$, $\gamma = 0$) for occluded facial images for the four data sets of the extended Yale Face Database B

	OCCL E-sparse SRC : Eyes occlusion				OCCL E-sparse SRC : Chin occlusion			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	100.0	80.0	65.6	70.0	100.0	70.0	62.5	75.0
2	100.0	90.0	90.6	65.0	100.0	50.0	68.8	50.0
3	100.0	60.0	68.8	60.0	90.0	50.0	68.8	60.0
4	100.0	80.0	78.1	85.0	100.0	40.0	62.5	40.0
5	100.0	70.0	62.5	75.0	100.0	30.0	59.4	15.0
6	100.0	50.0	68.8	50.0	100.0	30.0	53.1	30.0
7	100.0	80.0	56.3	75.0	100.0	60.0	59.4	75.0
8	100.0	80.0	65.6	65.0	100.0	70.0	71.9	70.0
9	100.0	70.0	90.6	90.0	100.0	40.0	84.4	60.0
10	100.0	70.0	87.5	70.0	90.0	70.0	62.5	55.0
11	100.0	100.0	87.5	85.0	100.0	90.0	96.9	90.0
12	100.0	70.0	78.1	75.0	100.0	60.0	75.0	75.0
13	100.0	50.0	65.6	60.0	90.0	30.0	50.0	50.0
14	100.0	40.0	46.9	40.0	100.0	10.0	37.5	25.0
15	100.0	70.0	56.3	55.0	100.0	30.0	56.3	35.0
16	100.0	70.0	78.1	65.0	100.0	20.0	53.1	50.0
17	100.0	50.0	56.3	55.0	100.0	20.0	56.3	65.0
18	100.0	70.0	78.1	50.0	100.0	50.0	59.4	35.0
19	100.0	100.0	100.0	70.0	100.0	100.0	84.4	90.0
20	100.0	90.0	96.9	70.0	100.0	90.0	93.8	65.0
Average	**100.0**	**72.0**	**73.9**	**66.5**	**98.5**	**50.5**	**65.8**	**55.5**
Overall	**78.1**				**67.6**			

5.5.3 Implementation Issues

Several issues encountered during the implementation of the above algorithms are discussed below.

A. E-Sparse Pre-Processing

- The orthonormal sparsifying transform $\mathcal{T}$ involved in the E-sparse pre-processing algorithm (see Section 5.3) was chosen as one of the following: 1-D DCT, 2-D DCT, 1-D DWT or 2-D DWT. These transforms can be implemented in MATLAB by using functions "dct", "dct2", "wt" and "wt2d", respectively, where the last two functions are available as long as the wavelet toolbox Uni_Wave [111] is installed.

B. SRC Algorithm

- SRC-related algorithms can be readily implemented as long as MATLAB toolbox CVX [105] is installed. An alternative MATLAB toolbox for implementing SRC algorithm is l_1-MAGIC [108]. It was found that the performance of these two toolboxes for face recognition purposes was quite similar to each other, while the elapsed computational time of solving a l_1-minimization problem using l_1-MAGIC was in average about 42% less that that required by CVX. The results concerning SRC-related algorithms reported in Chapters 5 and 6 were obtained using l_1-MAGIC.
- Using l_1-MAGIC to solve the optimization problem in (5.3) requires an initial point. In our simulations, the unitary vector was taken to be the required initial point.

5.6 Summary

Based on the work in [97] on sparse representation of facial images and [50] on sparse PCA for preliminary dimensionality reduction of large-scale data sets, an extended sparse PCA (E-sparse PCA) algorithm has been developed and an E-sparse SRC algorithm that combines two of the discussed algorithms for enhanced performance and efficient processing has been proposed. Experimental results were presented to support the proposed techniques.

Chapter 6

Face Recognition Systems – Integrating the Proposed Techniques

6.1 Introduction

Several methods for extension and performance enhancement of the standard PCA have been proposed in the preceding chapters. Some of these methods are for pre-processing the data involved, while other methods act as primary processing modules. It is therefore natural to construct face recognition systems that integrate the best modules in order to boost system performance in terms of recognition rate as well as robustness against variations in lighting conditions, measurement noise and facial occlusion. Based on our comprehensive experimental results, two face recognition systems are identified as the best among those studied in this thesis. These systems are described in Section 6.2. Performance evaluation of these systems is presented in Section 6.3.

6.2 Integration of the Best Modules

6.2.1 PHM E-2DPCA

The first system integration combines a PHM pre-processing module with an E-2DPCA processing module. In this system, a test image, as well as a training set, are first processed by modifying each of their image histograms to perfectly match a desired

histogram which is specified in the PHM module (see Section 3.3). The test image and training set so modified are then served as input data to which the E-2DPCA algorithm (see Section 4.3) is applied for face recognition. As long as the training set is not constantly updated with newly identified images, the calculations involved in pre-processing the training set and finding the required eigenvectors can be carried out offline, which greatly reduces the elapsed time in recognition process.

A block diagram is given in Figure 6.1 to illustrate the proposed PHM E-2DPCA based face recognition system.

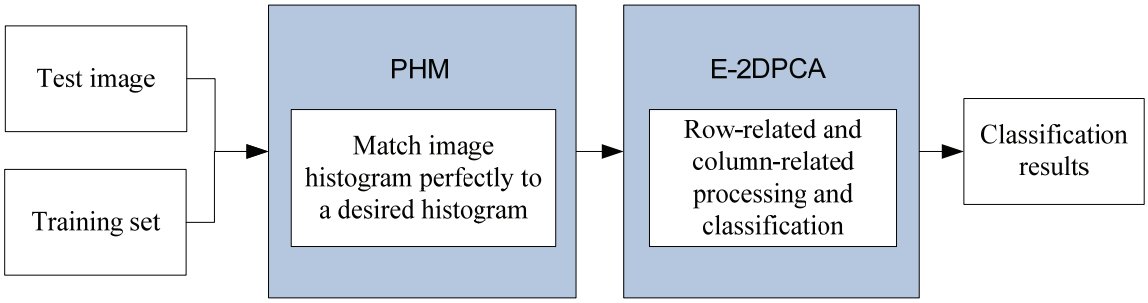


Figure 6.1. A block diagram of the PHM E-2DPCA based face recognition system.

6.2.2 PHM E-Sparse SRC

The second system integration combines a PHM module with an E-Sparse dimension reduction module and a processing module based on the SRC algorithm (see Section 5.4).

We remark that if the training set is not promptly updated with the newly identified test images, the pre-processing steps (i.e. perfectly matching the image histogram to the desired one and reducing the number of rows of data matrix A) can be carried out offline. However, all the processing operations necessary for face identification, such as finding the sparse representation of the test image in terms of the overcomplete dictionary constructed from the training set (see (5.3)), and using this sparse representation to classify the test image, must be done online. Solving the l_1-minimization problem in (5.3) is a time consuming process especially if large-scale data sets are involved, and this fact should not be neglected for real-time system implementation.

A block diagram is given in Figure 6.2 to illustrate the proposed PHM E-sparse SRD based face recognition system.

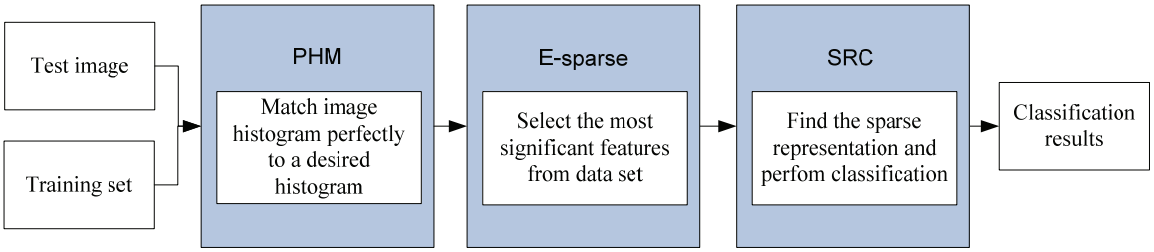


Figure 6.2. A block diagram of the PHM E-sparse SRC based face recognition system.

6.3 Experimental Results

The ten cases constructed from Yale Face Database (see Table 3.2 and Appendix 1) were used to examine the performance improvement brought by the PHM pre-processing module to the E-2DPCA algorithm. In addition, the six cases constructed from the extended Yale Face Database B (see Section 4.4.1) were re-examined in view of PHM E-2DPCA. The four data sets of the extended Yale Face Database B used earlier (see Table 5.4 and Appendix 2) were employed to evaluate and compare the PHM E-2DPCA with the PHM E-sparse SRC algorithms, as these data sets contain large training sets necessary in sparse representation framework.

6.3.1 Results for PHM E-2DPCA

In view of the fact that the PHM E-2DPCA algorithm can be applied to training sets of a variety of sizes, we first investigated its behavior when applied to the ten cases from Yale Face Database. As illustrated in Figure 6.3, for Cases 1, 2, 3 and 7, PHM E-2DPCA provided 100% recognition rates for all instances of p (the number of eigenvectors employed). For Cases 4, 5 and 6, a recognition rate of 93.3% was achieved. There were two instances in Cases 8 and 9 that had a lower recognition rate of 86.7%, while for the remaining instances of Cases 8 and 9, and for Case 10, recognition rates of 93.3% and 100% were attained. In comparison with PCA, 2DPCA and E-2DPCA methods, out of ten cases examined, there were seven cases (namely Cases 3, 4, 5, 6, 8, 9 and 10) where PHM E-2DPCA yielded improved performance with significant increase in recognition rate.

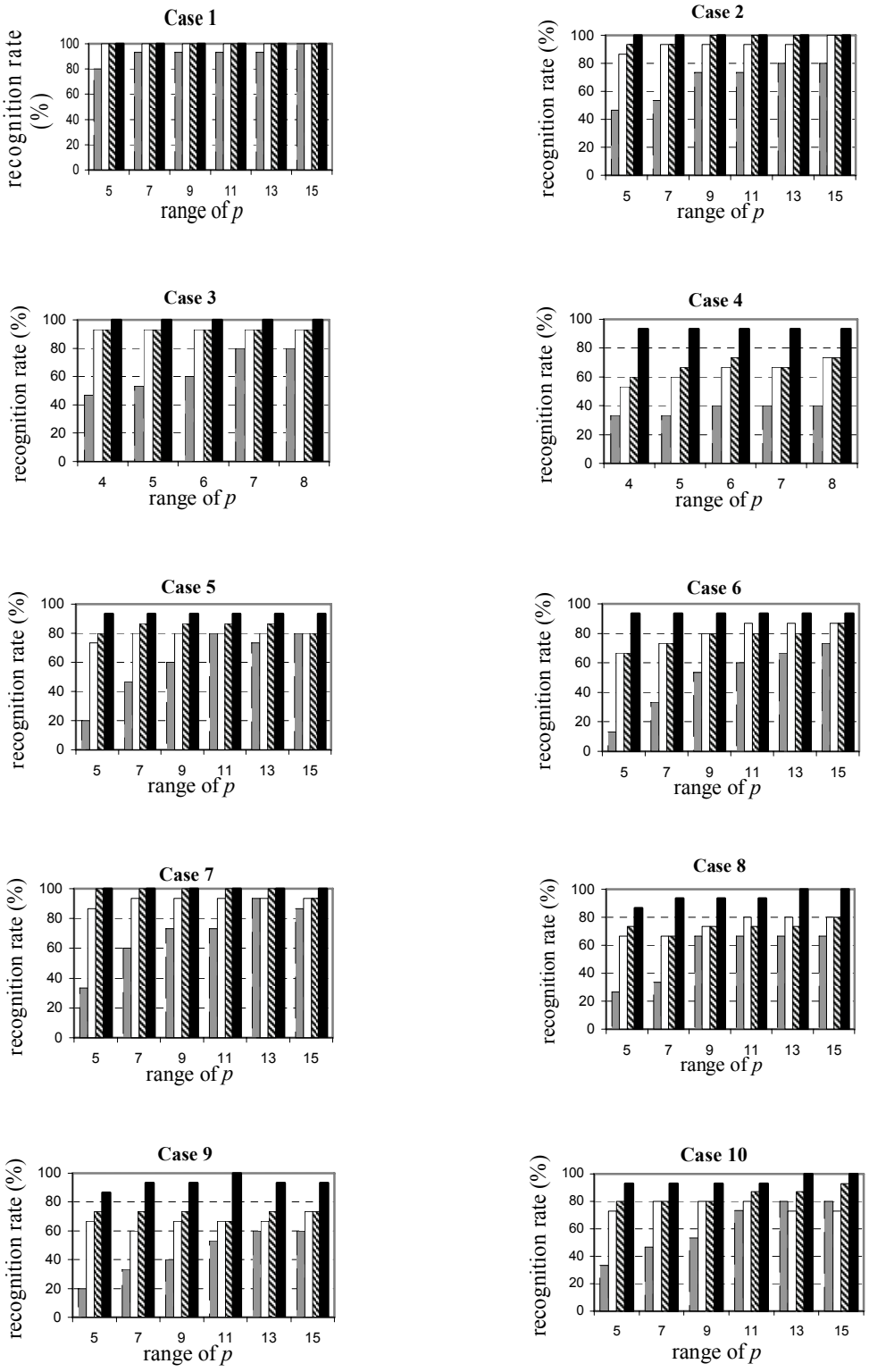


Figure 6.3. Comparison results for PCA (solid grey bar), 2DPCA (solid white bar), E-2DPCA (diagonal stripped bar), and PHM E-2DPCA (solid black bar), for all ten cases from the Yale Face Database.

When applied to the six cases form extended Yale Face Database B (see Section 4.4.1), the PHM E-2DPCA algorithm increased the recognition rates to 100% for Cases 1-5, for all instances of p with one exception (Case 4 with p = 10) where the rate was improved only to 85%. For Case 6 where other algorithms failed to perform face recognition adequately, the PHM E-2DPCA algorithm provided substantial improvement in recognition rate, as it can be seen from Figure 6.4.

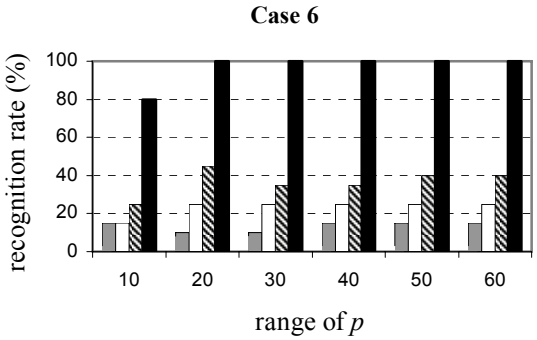


Figure 6.4. Comparison results for PCA (solid grey bar), 2DPCA (solid white bar), E-2DPCA (diagonal stripped bar), and PHM E-2DPCA (solid black bar), for Case 6 from the extended Yale Face Database B.

Next, the four data sets of the extended Yale Face Database B were utilized to evaluate the performance of the PHM E-2DPCA algorithm. As a first step in our evaluation process, the question of what number of eigenvectors, p, which should be employed in our simulations, was addressed. Table 6.1 shows the results obtained on Set 4, when 2.5%, 5%, 7.5% and 10% of the available eigenvectors were utilized. We see an increase of average recognition rate by 2.5% when the number of eigenvectors was increased from 2.5% to 5%. The use of 7.5% of the eigenvectors brings only a minor 0.5% improvement, while further increase of number of eigenvectors utilized does not lead to further performance improvement. As a result, in our subsequent simulations we have used 5% of the eigenvectors for the E-2DPCA algorithm, as it offered satisfactory performance with moderate computational complexity.

Table 6.1. Choosing the appropriate number of eigenvectors for PHM E-2DPCA – results for Set 4 from the extended Yale Face Database B

	PHM E-2DPCA			
Individual	p=22 (2.5%)	p=44 (5%)	p=66 (7.5%)	p=88 (10%)
1	95.0	100.0	100.0	100.0
2	85.0	90.0	90.0	90.0
3	100.0	100.0	100.0	100.0
4	100.0	100.0	100.0	100.0
5	90.0	95.0	100.0	100.0
6	80.0	95.0	100.0	100.0
7	90.0	90.0	90.0	90.0
8	100.0	100.0	100.0	100.0
9	100.0	100.0	100.0	100.0
10	100.0	100.0	100.0	100.0
11	95.0	95.0	95.0	95.0
12	95.0	100.0	100.0	100.0
13	100.0	100.0	100.0	100.0
14	90.0	95.0	95.0	95.0
15	95.0	100.0	100.0	100.0
16	100.0	100.0	100.0	100.0
17	100.0	100.0	100.0	100.0
18	100.0	100.0	100.0	100.0
19	90.0	95.0	95.0	95.0
20	100.0	100.0	100.0	100.0
Average	**95.3**	**97.8**	**98.3**	**98.3**

Depending on the number of images from the training set involved, the number of eigenvectors used in E-2DPCA was set to 54, 54, 32 and 44 for Set 1, 2, 3 and 4, respectively, representing 5% of the total number of available eigenvectors. Utilizing the four data sets described earlier, the performance of E-2DPCA was examined with and without PHM pre-processing. As illustrated in Table 6.2, when the PHM module was integrated into the system, significant performance improvement was obtained, especially for the more challenging Sets 2, 3, and 4, for which the E-2DPCA alone failed to work properly. An overall average recognition rate of 99.1% was attained by the PHM E-2DPCA based face recognition system.

Table 6.2. Comparison of E-2DPCA (left-hand side) with PHM E-2DPCA (right-hand side), for four data sets from the extended Yale Face Database B

	E-2DPCA				PHM E-2DPCA			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	90.0	0.0	43.8	0.0	100.0	100.0	93.8	100.0
2	80.0	30.0	53.1	10.0	100.0	100.0	100.0	90.0
3	80.0	0.0	40.6	0.0	100.0	100.0	100.0	100.0
4	90.0	40.0	50.0	20.0	100.0	100.0	100.0	100.0
5	90.0	0.0	31.3	0.0	100.0	100.0	100.0	95.0
6	90.0	10.0	34.4	5.0	100.0	100.0	96.9	95.0
7	80.0	100.0	100.0	100.0	100.0	100.0	96.9	90.0
8	80.0	30.0	40.6	5.0	100.0	100.0	100.0	100.0
9	90.0	30.0	40.6	20.0	100.0	100.0	96.9	100.0
10	90.0	20.0	34.4	0.0	100.0	100.0	100.0	100.0
11	90.0	20.0	50.0	0.0	100.0	100.0	100.0	95.0
12	90.0	20.0	43.8	5.0	100.0	100.0	100.0	100.0
13	90.0	0.0	40.6	0.0	100.0	100.0	100.0	100.0
14	90.0	0.0	34.4	0.0	100.0	100.0	100.0	95.0
15	90.0	30.0	43.8	5.0	100.0	100.0	100.0	100.0
16	80.0	10.0	28.1	10.0	100.0	100.0	93.8	100.0
17	90.0	10.0	28.1	5.0	100.0	100.0	96.9	100.0
18	90.0	0.0	31.3	0.0	100.0	100.0	100.0	100.0
19	90.0	0.0	40.6	0.0	100.0	100.0	100.0	95.0
20	90.0	0.0	31.3	0.0	100.0	100.0	100.0	100.0
Average	**87.5**	**17.5**	**42.0**	**9.3**	**100.0**	**100.0**	**98.8**	**97.8**
Overall	**39.1**				**99.1**			

Similar to previous simulations, the test face images were corrupted by additive zero-mean Gaussian white noise with various amounts of variance. Table 5.10 shows the performance of the PHM E-2DPCA algorithm in two scenarios: (a) noise was applied to test images and no de-noising was performed, and (b) noise was applied to test images followed by WT – TV de-noising. As can be observed, for Set 1, which includes test images with "easy" illumination conditions, the algorithm itself was found to be robust against noise. Nevertheless, the algorithm did not handle well the other three noise-corrupted data sets that contain test images with widely varying illumination conditions. The WT – TV de-noising algorithm applied as a pre-processing step to PHM E-2DPCA was able to offer a 19% overall performance improvement, with up to 33% recognition rate increase in average for Set 4.

Table 6.3. Results for PHM E-2DPCA with noisy test images and no de-noising (left-hand side) and noisy test images and WT – TV de-noising (right-hand side) for four data sets from the extended Yale Face Database B

	PHM E-2DPCA : Noisy, no de-noising				PHM E-2DPCA : Noisy, WT-TV de-noising			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	90.0	20.0	65.6	20.0	90.0	50.0	84.4	65.0
2	100.0	70.0	84.4	15.0	100.0	100.0	93.8	60.0
3	90.0	100.0	68.8	95.0	100.0	100.0	100.0	100.0
4	90.0	60.0	81.3	70.0	100.0	100.0	96.9	95.0
5	100.0	90.0	87.5	35.0	100.0	100.0	100.0	80.0
6	90.0	20.0	50.0	10.0	100.0	50.0	50.0	40.0
7	90.0	30.0	75.0	50.0	100.0	70.0	78.1	65.0
8	100.0	50.0	81.3	45.0	100.0	70.0	93.8	95.0
9	100.0	40.0	65.6	55.0	100.0	70.0	84.4	80.0
10	100.0	90.0	100.0	85.0	100.0	100.0	100.0	90.0
11	100.0	60.0	90.6	45.0	100.0	100.0	100.0	95.0
12	100.0	50.0	84.4	35.0	100.0	80.0	93.8	75.0
13	100.0	30.0	65.6	25.0	100.0	60.0	81.3	85.0
14	100.0	30.0	75.0	5.0	100.0	90.0	90.6	70.0
15	90.0	50.0	78.1	35.0	100.0	80.0	96.9	55.0
16	90.0	50.0	65.6	55.0	100.0	70.0	75.0	85.0
17	90.0	90.0	84.4	75.0	100.0	100.0	90.6	95.0
18	90.0	100.0	90.6	85.0	100.0	100.0	100.0	100.0
19	100.0	80.0	78.1	40.0	100.0	100.0	93.8	85.0
20	100.0	50.0	78.1	75.0	100.0	100.0	100.0	100.0
Average	**95.5**	**58.0**	**77.5**	**47.8**	**99.5**	**84.5**	**90.2**	**80.8**
Overall	**69.7**				**88.7**			

The robustness of the PHM E-2DPCA algorithm against facial occlusions was also examined for the same four data sets of the extended Yale face database B. The OCCL module (see Section 3.6) was employed as an initial pre-processing step in resolving the occlusion problem. Subsequently, the PHM algorithm was applied as a second pre-processing step. Table 6.4 shows the results obtained when eyes and chin occlusions occurred across the entire training sets. Similar results were obtained when the order of the two pre-processing modules was reversed.

We remark that the OCCL PHM E-2DPCA algorithm was found to be quite efficient in dealing with eyes and chin occlusions, achieving overall recognition rates of 96.2% and 98.8%, respectively.

Table 6.4. Results for OCCL PHM E-2DPCA applied to eyes-occluded (left-hand side) and chin-occluded (right-hand side) images for four data sets from the extended Yale Face Database B

	OCCL PHM E-2DPCA : Eyes occlusion				OCCL PHM E-2DPCA : Chin occlusion			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	90.0	100.0	87.5	95.0	100.0	100.0	90.6	100.0
2	100.0	100.0	96.9	80.0	100.0	100.0	100.0	80.0
3	100.0	100.0	100.0	100.0	100.0	100.0	96.9	100.0
4	90.0	100.0	93.8	100.0	100.0	100.0	96.9	100.0
5	90.0	100.0	96.9	95.0	100.0	100.0	100.0	90.0
6	90.0	100.0	71.9	80.0	100.0	100.0	87.5	95.0
7	100.0	100.0	87.5	95.0	100.0	100.0	100.0	95.0
8	100.0	100.0	90.6	95.0	100.0	100.0	100.0	100.0
9	100.0	90.0	93.8	90.0	100.0	100.0	100.0	100.0
10	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
11	100.0	100.0	100.0	95.0	100.0	100.0	100.0	95.0
12	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.0
13	100.0	100.0	90.6	95.0	100.0	100.0	100.0	100.0
14	100.0	90.0	84.4	80.0	100.0	100.0	96.9	95.0
15	100.0	100.0	96.9	100.0	100.0	100.0	96.9	100.0
16	100.0	100.0	100.0	95.0	100.0	100.0	96.9	100.0
17	100.0	100.0	93.8	100.0	100.0	100.0	100.0	100.0
18	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
19	100.0	100.0	96.9	80.0	100.0	100.0	100.0	100.0
20	100.0	100.0	96.9	100.0	100.0	100.0	100.0	100.0
Average	**98.0**	**99.0**	**93.9**	**93.8**	**100.0**	**100.0**	**98.1**	**97.3**
Overall	**96.2**				**98.8**			

6.3.2 Results for PHM E-Sparse SRC

The same four data sets of the extended Yale Face Database B have been also employed to evaluate the performance of PHM E-sparse SRC algorithm. Similar to the set-up in Chapter 5, the E-sparse module was implemented using 2-D DCT with $d = 100$ and $\gamma = 0$.

First, the performance of the E-sparse SRC algorithm was evaluated with and without the PHM pre-processing. As it can be seen from Table 6.5, a 6.5% increase in overall recognition rate was gained when PHM module was integrated into the E-sparse SRC algorithm.

Table 6.5. Comparison of E-sparse SRC (left-hand side) with PHM E-sparse SRC (right-hand side), for four data sets from the extended Yale Face Database B

	E-sparse SRC				PHM E-sparse SRC			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	100.0	90.0	68.8	90.0	100.0	100.0	53.1	95.0
2	100.0	100.0	84.4	70.0	100.0	100.0	68.8	85.0
3	100.0	50.0	81.3	65.0	100.0	100.0	71.9	85.0
4	100.0	90.0	68.8	75.0	100.0	100.0	59.4	100.0
5	100.0	60.0	75.0	65.0	100.0	60.0	68.8	60.0
6	100.0	90.0	65.6	70.0	100.0	60.0	46.9	80.0
7	100.0	90.0	71.9	85.0	100.0	100.0	65.6	90.0
8	100.0	100.0	65.6	80.0	100.0	100.0	87.5	100.0
9	100.0	100.0	81.3	85.0	100.0	90.0	68.8	100.0
10	100.0	90.0	78.1	70.0	100.0	100.0	93.8	85.0
11	100.0	80.0	75.0	90.0	100.0	100.0	100.0	85.0
12	100.0	70.0	78.1	60.0	100.0	90.0	75.0	80.0
13	100.0	70.0	62.5	70.0	100.0	100.0	53.1	90.0
14	100.0	60.0	37.5	55.0	100.0	100.0	90.6	90.0
15	100.0	80.0	65.6	65.0	100.0	100.0	59.4	80.0
16	100.0	50.0	56.3	65.0	100.0	80.0	53.1	70.0
17	100.0	70.0	71.9	90.0	100.0	100.0	84.4	100.0
18	100.0	90.0	62.5	85.0	100.0	100.0	59.4	95.0
19	100.0	90.0	96.9	85.0	100.0	100.0	96.9	100.0
20	100.0	90.0	90.6	75.0	100.0	90.0	68.8	95.0
Average	**100.0**	**80.5**	**71.9**	**74.8**	**100.0**	**93.5**	**71.3**	**88.3**
Overall	**81.8**				**88.3**			

Next, the PHM E-sparse SRC algorithm was applied to noise-contaminated test images with and without a de-noising step. The results obtained are shown in Table 6.6. It is observed that except for Set 1, the algorithm does not perform very well even with a WT – TV de-noising step. In addition, the WT – TV de-noising was not as efficient as for the PHM E-2DPCA based system, see Table 6.3.

On the other hand, the robustness of PHM E-sparse SRC algorithm in conjunction with OCCL pre-processing against facial occlusions was found to be quite impressive. From Table 6.7 we see that when eyes and chin occlusions occurred across all images of the training sets, high (94.3% and 95.4%, respectively) overall recognition rates were achieved.

Table 6.6. Results for PHM E-sparse SRC with noisy test images and no de-noising (left-hand side) and noisy test images and WT – TV de-noising (right-hand side) for four data sets from the extended Yale Face Database B

	PHM E-sparse SRC : Noisy, no de-noising				PHM E-sparse SRC : Noisy, WT-TV de-noising			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	100.0	60.0	50.0	70.0	100.0	90.0	53.1	55.0
2	100.0	90.0	81.3	80.0	100.0	90.0	78.1	75.0
3	90.0	70.0	43.8	40.0	100.0	90.0	56.3	65.0
4	100.0	80.0	56.3	80.0	100.0	90.0	53.1	75.0
5	100.0	50.0	62.5	40.0	100.0	30.0	56.3	50.0
6	100.0	30.0	43.8	35.0	100.0	20.0	40.6	55.0
7	100.0	90.0	78.1	85.0	100.0	70.0	62.5	75.0
8	100.0	80.0	71.9	100.0	100.0	90.0	81.3	95.0
9	100.0	90.0	68.8	100.0	100.0	80.0	68.8	95.0
10	100.0	70.0	75.0	70.0	90.0	90.0	78.1	75.0
11	100.0	80.0	100.0	60.0	100.0	90.0	100.0	80.0
12	100.0	50.0	71.9	45.0	100.0	50.0	81.3	50.0
13	100.0	80.0	46.9	80.0	100.0	90.0	43.8	90.0
14	100.0	70.0	68.8	45.0	100.0	80.0	75.0	85.0
15	100.0	80.0	46.9	60.0	100.0	80.0	46.9	70.0
16	100.0	40.0	50.0	60.0	100.0	60.0	50.0	65.0
17	100.0	40.0	59.4	70.0	90.0	50.0	65.6	90.0
18	100.0	90.0	46.9	75.0	100.0	80.0	50.0	85.0
19	100.0	90.0	90.6	80.0	100.0	100.0	96.9	95.0
20	100.0	80.0	68.8	85.0	100.0	80.0	68.8	90.0
Average	**99.5**	**70.5**	**64.1**	**68.0**	**99.0**	**75.0**	**65.3**	**75.8**
Overall	**75.5**				**78.8**			

6.3.3 Performance Comparison

On comparing the results from Sections 6.3.1 and 6.3.2, it is observed that PHM pre-processing in combination with E-2DPCA proved extremely efficient for face recognition when test images were noise free. Moreover, test images with eyes/chin occlusion were effectively handled when an OCCL module was incorporated into the PHM E-2DPCA algorithm. Robustness against noise was significantly enhanced when a WT- TV de-noising step was applied to noisy data before PHM E-2DPCA processing. The low point of WT – TV PHM combined pre-processing was that it did not help the E-sparse SRC algorithm in dealing with noise-contaminated test images.

Table 6.7. Results for OCCL PHM E-sparse SRC applied to eyes-occluded (left-hand side) and chin-occluded (right-hand side) images for four data sets from the extended Yale Face Database B

	OCCL PHM E-sparse SRC : Eyes occlusion				OCCL PHM E-sparse SRC : Chin occlusion			
Individual	Set 1	Set 2	Set 3	Set 4	Set 1	Set 2	Set 3	Set 4
1	100.0	100.0	78.1	95.0	100.0	100.0	96.9	100.0
2	100.0	90.0	93.8	80.0	100.0	100.0	90.6	95.0
3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.0
4	100.0	100.0	84.4	95.0	100.0	100.0	78.1	100.0
5	100.0	80.0	68.8	85.0	100.0	100.0	78.1	80.0
6	100.0	80.0	50.0	100.0	100.0	90.0	50.0	75.0
7	100.0	100.0	75.0	95.0	100.0	100.0	71.9	95.0
8	100.0	100.0	96.9	95.0	100.0	100.0	100.0	100.0
9	100.0	100.0	90.6	95.0	100.0	100.0	93.8	100.0
10	100.0	90.0	78.1	90.0	100.0	90.0	87.5	90.0
11	100.0	100.0	100.0	95.0	100.0	100.0	100.0	95.0
12	100.0	100.0	100.0	100.0	100.0	100.0	100.0	95.0
13	100.0	90.0	68.8	95.0	100.0	100.0	100.0	100.0
14	100.0	100.0	100.0	95.0	100.0	100.0	90.6	90.0
15	100.0	90.0	84.4	85.0	100.0	100.0	100.0	80.0
16	100.0	100.0	78.1	90.0	100.0	90.0	56.3	90.0
17	100.0	100.0	100.0	95.0	100.0	100.0	100.0	100.0
18	100.0	100.0	100.0	95.0	100.0	100.0	100.0	95.0
19	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
20	100.0	100.0	96.9	100.0	100.0	100.0	96.9	95.0
Average	**100.0**	**96.0**	**87.2**	**94.0**	**100.0**	**98.5**	**89.5**	**93.5**
Overall	**94.3**				**95.4**			

Comparing the two systems, namely the PHM E-2DPCA and PHM E-sparse SRC, PHM E-2DPCA appears to perform better especially for more challenging data sets such as Set 3 and 4. Concerning the elapsed time for a typical face identification task, in average over the four data sets, the PHM E-2DPCA algorithm was found to be 7.64 times faster than PHM E-sparse SRC. This significant difference is mainly due to the fact that the SRC processing module has to solve an l_1-minimization problem given in (5.3) for each test image.

Based on the above observations, we propose a face recognition system that is aimed at achieving the best performance in terms of face recognition rate and computational complexity. A block diagram of the system is given in Figure 6.5.

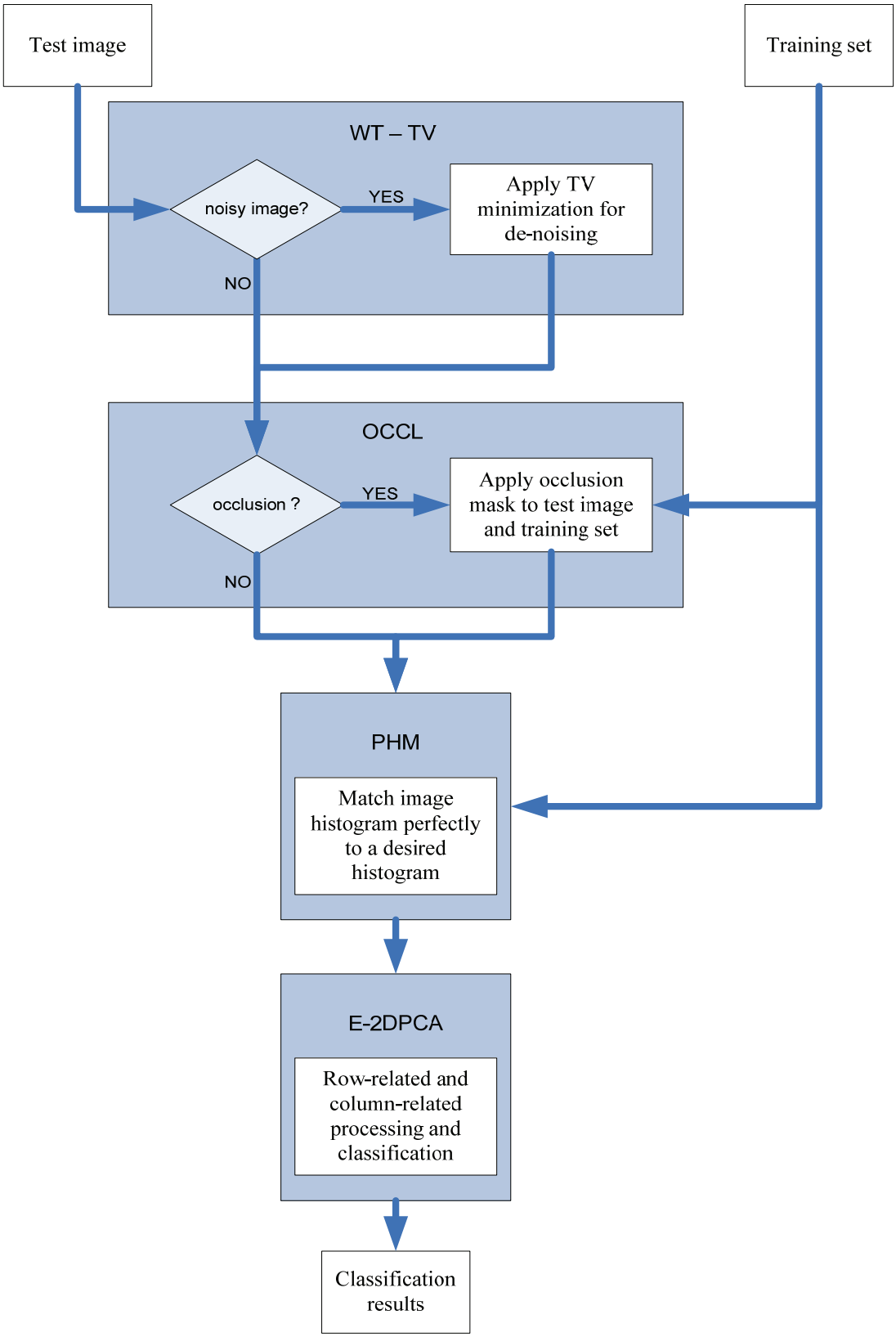


Figure 6.5. A block diagram of the proposed face recognition system.

6.3.4 Implementation Issues

Below are highlighted several general implementation issues encountered in the simulation study.

- The simulations using the four data sets from the extended Yale Face Database B were performed on a 64-bit PC system architecture with 4 GB of memory, using a 64-bit version of MATLAB.
- The AT&T Face Database, which is one of the most utilized face databases for testing algorithms for face occlusion, was unfortunately not available to the author. As a result, face occlusions in our simulations were created using images from the extended Yale Face Database B.
- Two face databases, namely the Yale Face Database [112] and the extended Yale Face Database B [106], [107], were used to construct several data sets with various facial expressions and/or illumination conditions. These data sets were utilized to evaluate and compare the algorithms' performance throughout the thesis.

6.4 Summary

We have described and compared two most promising face recognition systems based on the PHM E-2DPCA and PHM E-sparse SRC algorithms. Simulation results have demonstrated that the PHM E-2DPCA technique, in conjunction with WT – TV de-noising and OCCL occlusion-resolving modules, provides the highest performance in terms of recognition rate and elapsed computational time. Finally, a face recognition system integrating WT – TV, OCCL, PHM and E-2DPCA modules has been proposed.

Chapter 7

Conclusions and Future Research

In this chapter, we summarize the contributions made in the thesis and suggest several directions for future research in the field of face recognition.

7.1 Conclusions

In this thesis, we have investigated face recognition problems under variable lighting conditions and facial expressions, as well as possible face occlusions and noise contamination, and proposed several methods for pre-processing and processing for improved solutions. Although some techniques proposed in thesis are of general nature and may be incorporated into a variety of face recognition systems, the methods and algorithms developed in the thesis are primarily enhancement and extensions of the well-known principal component analysis (PCA).

In more specific terms, we have developed two sets of methods and algorithms – one for pre-processing and one for PCA-based processing.

The pre-processing methods proposed in the thesis include the PHM, WT – TV de-noising, an OCCL algorithm, and an E-sparse module for preliminary dimensionality reduction. The PHM can be used to obtain homogeneous tonal distribution for both training data set and test image. Through numerous experimental studies with convincing evidence, the PHM was found to be computationally inexpensive and outperforming many existing pre-processing techniques. We stress that the PHM as a generic pre-

processing tool has great potential to be integrated into other face recognition systems such as those based on independent component analysis (ICA) [3] for performance enhancement. Furthermore, the PHM module can be regarded as a simple yet effective radiometric calibration technique [69].

The WT – TV de-noising is a pre-processing module that can be used to help deal with test images that are contaminated with noise. A wavelet decomposition is employed for estimating noise variance and, based on this estimation, a TV minimization is carried out for noise reduction.

In the OCCL pre-processing module, eyes and chin occlusions have been considered and a simple yet effective algorithm to deal with these occlusions is proposed.

The E-sparse PCA is developed as an extension of the sparse PCA proposed by Johnstone and Lu [50]. It can be viewed as a pre-processing module because the algorithm can be used to reduce the data size from $m \times n$ to $d \times n$ with $d \ll m$ without sacrificing substantial performance degradation, and it is typically utilized before a standard PCA is applied. In particular, the E-sparse module for preliminary reduction is found effective when used in conjunction with the SRC algorithm, where the E-sparse helps reduce large-scale data that would be otherwise encountered in the SRC algorithm.

While the PHM is a major development for pre-processing the data, the proposal of the extended 2-D PCA (E-2DPCA) is a major development for performing a face recognition task. In effect, the E-2DPCA is a core module of the proposed face recognition system that replaces the standard PCA or the 2-D PCA proposed earlier in [100]. Considerable effort has been made to explain the difference between the 2-D PCA and E-2DPCA: the E-2DPCA handles facial images more naturally than what 2-D PCA does as it is able to extract features that are inherently associated with the rows as well as the columns of a facial image. The superior performance of the E-2DPCA is demonstrated by the experimental results. Especially when it is used in conjunction with PHM, the system excels among the face recognition systems constructed with other combinations of processing modules.

7.2 Suggestions for Future Research

There are several problems that are closely related to the work presented in this thesis and appear to be worthwhile for further investigation.

1. As a pre-processing approach for PCA-based algorithms, one may carry out an initial image partitioning step to identify the local features in facial images with improved accuracy, where a face image is first partitioned into several sub-images, each containing some facial features. As a result, the face identification problem in this case becomes a set of lower-dimensional feature identification problems. Accordingly, face classification is performed by weighting the contributions of different facial features.
2. Concerning the extended two-dimensional processing approach (E-2DPCA) proposed in Chapter 4, it appears to be beneficial to develop an algorithm with an improved classification method. For example, one can obtain an image representation by applying the optimal projection matrices X and Z from (4.5a) and (4.5b) simultaneously to image A, resulting a *single* feature matrix U of size $p \times p$:

$$U = Z^T \cdot A \cdot X ,$$

and develop an appropriate classification measure based on the above image representation.
3. Concerning the sparse representation approaches, for the E-sparse dimensionality reduction module, there are several issues associated with sparsifying transform $\mathcal{T}$ and its parameters that appear worthwhile to investigate. These include the interplay between parameters L, $\tilde{d}$ and γ when a 2-D DWT is employed.
4. Performance improvement over the standard SRC algorithm [97] may be achieved by optimizing the random projection matrix R in the SRC algorithm. Several methods for optimization of random projection matrix in compressed sensing framework are available. In [30], for example, an optimization method is developed where an average measure of the mutual coherence of the effective dictionary is used; reference [28] makes use of singular value decomposition-based techniques to perform the optimization of the sensing matrix for a given dictionary. These methods are found to be useful in improving reconstruction accuracy in a

compressed sensing framework. This certainly motivates to develop algorithms to optimize a random projection matrix that is tailored to a face recognition scenario.

5. Finally, we would like to point out an issue pertaining to the construction of a training set that is best suited for the purpose of face recognition in terms of both performance and efficiency. Especially with regard to the matter of computational complexity, we suggest to develop an extension of the E-sparse dimensionality reduction technique that will hopefully be able to reduce a data matrix of size $m \times n$ to $d \times n'$ with $d \ll m$ and $n' \ll n$ before applying the standard PCA.

Bibliography

[1] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: the problem of compensating for changes in illumination direction," *IEEE Trans. on PAMI*, vol. 19, pp. 721 – 732, 1997.

[2] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. On Signal Processing*, vol. 54, pp. 4311 – 4322, 2006.

[3] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. on Neural Networks*, vol. 13(6), pp. 1450–1464, 2002.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing Systems*, vol. 14, pp. 585–591, 2002.

[5] M. Belkin and P. Niyogi, "Towards a theoretical foundation for Laplacian-based manifold methods," *J. of Computer and System Sciences*, vol. 74, pp. 1289–1308, 2008.

[6] P.N. Beluhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. on PAMI*, vol. 19, pp. 711 – 720, 1997.

[7] P.N. Belhumeur and D.J. Kriegman, "What is the set of images of an object under all possible lighting conditions?," *Proc. on IEEE Conf. CVPR*, pp. 207-277, 1996.

[8] B. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. on PAMI*, vol. 25(9), pp. 1063-1074, 2003.

[9] W.W. Bledsoe, "The model method in facial recognition," Panoramic Research Inc., Palo Alto, CA, Rep. PRI:15, 1966.

[10] W.W. Bledsoe, "Man-machine facial recognition," Panoramic Research Inc., Palo Alto, CA, Rep. PRI:22, 1966.

[11] V. Bruce, *Recognizing faces*, Lawrence Erlbaum Associates, London, U.K., 1988.

[12] V. Bruce, M. Burton, and N. Dench, "What's distinctive about a distinctive face?," *Quart. J. Exp. Psych.*, vol. 47A, pp. 119–141, 1994.

[13] V. Bruce, P.J.B. Hancock, and A.M. Burton, "Human face perception and identification," in *Face Recognition: From Theory to Applications*, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulie, and T. S. Huang, Eds. Springer-Verlag, Berlin, Germany, pp. 51–72, 1998.

[14] D.D. Busch, *Mastering Digital SLR Photography*, Thomson Course Technology, 2005.

[15] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure and Applied Math.*, vol. 59(8), pp. 1207-1223, 2006.

[16] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?," *IEEE Trans. Information Theory*, vol. 52(12), pp. 5406-5425, 2006.

[17] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *IEEE Proc.*, vol. 83, pp. 705-741, 1995.

[18] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20(1), pp. 33–61, 1999.

[19] S. Chen and Y. Zhu, "Subpattern-based principle component analysis," *Pattern Recognition*, vol. 37, pp. 1081–1083, 2004.

[20] F. Chichizola, L. De Giusti, A. De Giusti, and M. Naiouf, "Face recognition: reduced image eigenfaces method," *ELMAR 47th Int. Symposium*, pp. 159–162, 2005.

[21] D.L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Information Theory*, vol. 41(3), pp. 613–627, 1995.

[22] D.L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, pp. 1289-1306, 2006.

[23] D.L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Comm. Pure and Applied Math.*, vol. 59(6), pp. 797- 829, 2006.

[24] D.L. Donoho and I.M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biomefrika*, vol. 81, pp. 425-455, 1994.

[25] D.L. Donoho, I.M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: asymptopia?," *J. R. Statist. Soc. B*., vol. 57, pp. 301-369, 1995.

[26] B.A. Draper, K. Baek, M.S. Bartlett, and J.R. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, pp. 115-137, 2003.

[27] B. Du, S. Shan, L. Qing, and W. Gao, "Empirical comparisons of several preprocessing methods for illumination insensitive face recognition," *IEEE Proc. ICASSP*, vol. 2, pp. ii/981 - ii/984, 2005.

[28] J.M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. on Image Processing*, vol. 18(7), pp. 1395-1408, 2009.

[29] H.K. Ekenel and R. Stiefelhagen, "Why is facial occlusion a challenging problem?," *Int. Conf. on Biometrics*, pp. 299-308, 2009.

[30] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. on Signal Processing*, vol. 55(12), pp. 5695-5702, 2007.

[31] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," *Proc. of CVPR*, vol. 1, pp. 895 – 900, 2006.

[32] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representation," *Proc. of CVPR*, vol. 2, pp. 1924 – 1931, 2006.

[33] K. Etemad and R. Chellappa, "Face recognition using discriminant eigenvectors," *IEEE Proc.*, *ICASSP*, vol. 4, pp. 2148–2151, 1996.

[34] S. Fidler, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Trans. on PAMI*, vol. 28(3), pp. 337-350, 2006.

[35] M.A. Fischler and R.A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. on Computers*, vol. C-22, pp. 67-92, 1973.

[36] R. Fransens, C. Strecha, and L. Van Gool, "Robust estimation in the presence of spatially coherent outliers," *Proc. on Conf. on CVPR Workshop*, pp. 102-102, 2006.

[37] I. Gauthier and N.K. Logothetis, "Is face recognition so unique after all?," *J. Cogn. Neuropsych.*, vol. 17, pp. 125–142, 2000.

[38] Y.Z. Goh, A.B.J. Teoh, and M.K.O. Goh, "Wavelet based illumination invariant preprocessing in face recognition," *Congress on Image and Signal Processing*, vol. 3, pp. 421-425, 2008.

[39] A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of human faces," *IEEE Proc*., vol. 59, pp. 748-760, 1971.

[40] R.C. Gonzales and R.E. Woods, *Digital Image Processing, Second Edition*, Prentice-Hall, New Jersey, 2002.

[41] R. Gottumukkal and V.K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, vol. 25, pp. 429–436, 2004.

[42] R. Gross and V. Brajovic, "An image preprocessing algorithm for illumination invariant face recognition," *4th Int. Conf. on AVBPA*, pp.10 – 18, 2003.

[43] P. Hallinan, "A low-dimensional representation of human faces for arbitrary lighting conditions," *Proc. IEEE CVPR*, pp. 995-999, 1994.

[44] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. on PAMI*, vol. 27(3), pp. 328–340, 2005.

[45] H. Hill, P.G. Schyns, and S. Akamatsu, "Information and viewpoint dependence in face recognition," *Cognition*, vol. 62, pp. 201-222, 1997.

[46] P.-C. Hsieh and P.-C. Tung, "A novel hybrid approach based on sub-pattern technique and whitened PCA for face recognition," *Pattern Recognition*, vol. 42(5), pp. 978–984, 2009.

[47] K. Huang and S. Aviyente, "Sparse representation for signal classification," *Advances in Neural Information Processing Systems*, vol. 19, pp. 609-616, 2007.

[48] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey, 1989.

[49] A. Johnston, H. Hill, and N. Carman, "Recognizing faces: effects of lighting direction, inversion and brightness reversal," *Cognition*, vol. 40, pp. 1–19, 1992.

[50] I.M. Johnstone and A.Y. Lu, "Sparse principal components analysis," unpublished manuscript, 2004.

[51] T. Kanade and A. Yamada, "Multi-subregion based probabilistic approach toward pose-invariant face recognition," *Proc., IEEE Computational Intelligence in Robotics Automation*, vol. 2, pp. 954-959, 2003.

[52] M. Kirby and L. Sirovich, "Application of Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. PAMI*, vol. 12, pp. 103-108, 1990.

[53] K. Kotani, F.F. Lee, Q. Chen, and T. Ohmi, "Face recognition based on the adjacent pixel intensity difference quantization histogram method," *Int. Sym. on Intelligent Signal Processing and Communication Systems*, D7-4, pp.877-880, 2003.

[54] K.-C. Kwak and W. Pedrycz, "Face recognition using an enhanced independent component analysis approach," *IEEE Trans. on Neural Networks*, vol. 18(2), pp. 530–541, 2007.

[55] M. Langford and E. Bilissi, *Langford's Advanced Photography*, *7th Edition*, Focal Press, Elsevier, 2008.

[56] F. Lee, K. Kotani, Q. Chen, and T. Ohmi, "Face recognition based on the combination of histogram features and rough location information of facial parts," *9th Int. Conf. on Signal Processing*, pp. 2065 – 2069, 2008.

[57] H.-S. Lee and D. Kim, "Pose invariant face recognition using linear pose transformation in feature space," *International Workshop on Human-Computer Interaction*, Springer, vol. 3058, pp. 200-210, 2004.

[58] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: the importance of good features," *CVPR*, pp. 53-60, 2004.

[59] Y. Li, A. Cichocki, and S. Amari, "Analysis of sparse representation and blind source separation," *Neural Computation*, vol. 16(6), pp. 1193–1234, 2004.

[60] Y. Li, S. Gong, and H. Liddell, "Recognizing the dynamics of faces across multiple views," *Proc., British Machine Vision Conf.*, pp. 242-251, 2000.

[61] L.-Z. Liao, S.-W. Luo, and M. Tian, "'Whitenedfaces' recognition with PCA and ICA," *IEEE Signal Processing Letters*, vol. 14(12), pp. 1008–1011, 2007.

[62] C. Liu, W.T. Freeman, R. Szeliski, and S.B. Kang, "Noise estimation from a single image," *IEEE Computer Society Conf. on CVPR*, vol. 1, pp. 901-908, 2006.

[63] X. Liu and T. Chen, "Video-based face recognition using adaptive hidden Markov models," *Proc., IEEE CVPR*, vol. 1, pp. 340-345, 2003.

[64] X. Liu and T. Chen, "Pose-robust face recognition using geometry assisted probabilistic modeling," *Proc., IEEE CVPR*, vol. 1, pp. 502-509, 2005.

[65] W.-S. Lu, *ELEC 639B*, *Selected Topics in Image Processing, Variational PDE Models and Applications in Image Processing*, Lecture Notes, Spring 2008.

[66] X. Lu, Y. Wang, and A.K. Jain, "Combining classifiers for face recognition," *ICME Proc*., vol. 3, pp. 13-16, 2003.

[67] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[68] A.M. Martinez, "Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class," *IEEE Trans. on PAMI*, vol. 24(6), pp. 748-763, 2002.

[69] H.B. Mitchell, *Multi-Sensor Data Fusion: An Introduction*, Springer, 2007.

[70] S.A. Nazeer, M. Khalid, N. Omar, and M.K. Awang, "Enhancement of neuro-eigenspace face recognition using photometric normalization," *Computer Graphics, Imaging and Visualisation*, pp. 370-376, 2007.

[71] A.V. Nefian and M.H. Hayes III, "Hidden Markov models for face recognition," *Proc., Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 2721–2724. 1998.

[72] B. Niu, Q. Yang, S.C.K. Shiu, and S.K. Pal, "Two-dimensional Laplacianfaces method for face recognition," *Pattern Recognition*, vol. 41(10), pp. 3237–3243, 2008.

[73] K. Okada and C. von der Malsburg, "Pose-invariant face recognition with parametric linear subspaces," *Proc., 5th Int. Conf. Automatic Face and Gesture Recognition*, pp. 64-69, 2002.

[74] B. Olshausen, P. Sallee, and M. Lewicki, "Learning sparse image codes using a wavelet pyramid architecture," *NIPS*, pp. 887–893, 2001.

[75] A. Pal, "Multicues face detection in complex background for frontal faces," *Int. Machine Vision and Image Processing Conf.*, pp. 57–62, 2008.

[76] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *27th Annual Asilomar Conf. on Signals, Systems, and Computers*, 1993.

[77] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on PAMI*, vol. 12(7), pp. 629–639, 1990.

[78] X. Qing and X. Wang, "Face recognition using Laplacian+OPRA-faces," *6th World Congress on Intelligent Control and Automation*, vol. 2, pp. 10013–10016, 2006.

[79] D. Ramasubramanian and Y.V. Venkatesh, "Encoding and recognition of faces based on the human visual model and DCT," *Pattern Recognition*, vol. 34(12), pp. 2447-2458, 2001.

[80] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290(5500), pp. 2323–2326, 2000.

[81] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D.*, vol. 60, pp. 259-268, 1992.

[82] L.K. Saul and S.T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *J. of Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[83] J. Sergent, "Microgenesis of face perception," in *Aspects of Face Processing*, H. D. Ellis, M. A. Jeeves, F. Newcombe, and A. Young, Eds. Nijhoff, Dordrecht, The Netherlands, 1986.

[84] A.-M. Sevcenco and W.-S. Lu, "Histogram-enhanced principal component analysis for face recognition," *IEEE PacRim Conf. on Communications, Computers and Signal Processing*, pp. 175-180, 2009.

[85] A.-M. Sevcenco and W.-S. Lu, "Perfect histogram matching PCA for face recognition," *J. Multidim. Syst. Sig. Process.* (DOI 10.1007/s11045-009-0099-y), 2010.

[86] A.-M. Sevcenco and W.-S. Lu, "An enhanced and extended 2-D PCA technique for face recognition", to be submitted to a journal, 2010.

[87] M. Sharkas and M.A. Elenien, "Eigenfaces vs. Fisherfaces vs. ICA for face recognition; a comparative study," *Proc. of ICSP*, pp. 914-919, 2008.

[88] A. Shashua, "On photometric issues in 3D visual recognition from a single 2D image," *Int. J. Computer Vision*, vol. 21(1/2), pp. 99-122, 1997.

[89] A. Shukla, J. Dhar, C. Prakash, D. Sharma, R.K. Anand, and S. Sharma, "Intelligent biometric system using PCA and R-LDA," *Global Congress on Intelligent Systems*, vol.1, pp. 267-272, 2009.

[90] L. Sirovich and M. Kirby, "Low-dimensional procedure for characterization of human faces," *J. Optical Soc. Am.*, vol. 4, pp. 519-524, 1987.

[91] J. Starck, M. Elad, and D. Donoho, "Image decomposition via the combination of sparse representation and a variational approach," *IEEE Trans. on Image Processing*, vol. 14(10), pp. 1570–1582, 2005.

[92] A. Stefano, P. White, and W. Collis, "Training methods for image noise level estimation on wavelet components," *EURASIP J. on Applied Signal Processing*, vol. 16, pp. 2400– 2407, 2004.

[93] K. Tan and S. Chen, "Adaptively weighted sub-pattern PCA for face recognition," *Neurocomputing*, vol. 64, pp. 505–511, 2005.

[94] X. Tan, S. Chen, Z.H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble," *IEEE Trans. on Neural Networks*, vol. 16(4), pp. 875-886, 2005.

[95] J.B. Tenenbaum, V. Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290(5500), pp. 2319–2323, 2000.

[96] M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," *IEEE Proc.*, *Computer Society Conf. on CVPR*, pp. 586–591, 1991.

[97] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on PAMI*, vol. 31, pp. 210-227, 2009.

[98] X. Xie and K.-M. Lam, "An efficient illumination compensation scheme for face recognition," *8th ICARCV,* vol. 2, pp. 1240-1243, 2004.

[99] S. Yan, H. Wang, X. Tang, and T. Huang, "Exploring feature descriptors for face recognition," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 1, pp. 629 – 632, 2007.

[100] J. Yang, D. Zhang, A.F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. on PAMI*, vol. 26(1), pp. 131–137, 2004.

[101] A.L. Yuille, D.S. Cohen, and P.W. Hallinan, "Feature extraction from faces using deformable templates," *CVPR Proc.*, *IEEE Computer Society Conf. on*, pp. 104-109, 1989.

[102] L. Zhao and Y.-H. Yang, "Theoretical analysis of illumination in PCA-based vision systems," *Pattern Recognition*, vol. 34(4), pp. 547-564, 1999.

[103] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face recognition: a literature survey," *ACM Computing Surveys*, pp. 399-458, 2003.

[104] Camera histograms: tones and contrast, http://www.cambridgeincolour.com/tutorials/histograms1.htm.

[105] CVX: MATLAB software for disciplined convex programming, version 1.21, Michael Grant and Stephen Boyd, 2010, http://cvxr.com/cvx/.

[106] *Extended Yale Face Database B* – A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. PAMI*, vol. 23(6), pp. 643-660, 2001.

[107] *Extended Yale Face Database B* – K.C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. PAMI*, vol. 27(5), pp. 684-698, 2005.

[108] l_1-MAGIC Toolbox: Recovery of sparse signals via convex programming, Emmanuel Candes and Justin Romberg, Caltech October 2005, http://www.acm.caltech.edu/l1magic/.

[109] MathWorks Product Support, Maximum matrix size by platform, http://www.mathworks.com/support/tech-notes/1100/1110.html.

[110] MATLAB (R2009b) Help, 2009.

[111] Uni_Wave Wavelet Toolbox, version 3.0 for MATLAB, Signal Theory Group, University of Vigo, http://www.gts.tsc.uvigo.es/~wavelets/uvi_wave.html.

[112] *Yale Face Database*, Yale University, CT, USA, http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

Appendix

Training and Testing Sets

A.1 The Yale Face Database [112]

Case 1

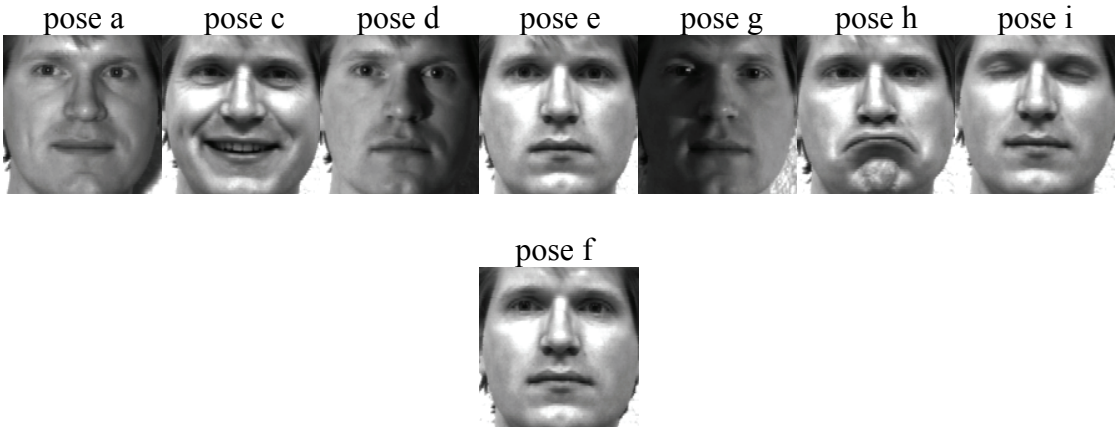


Figure A.1. The Yale Face Database – Case 1: seven poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 2

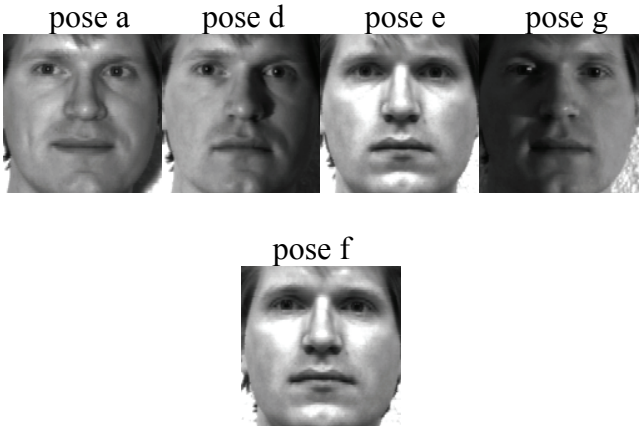


Figure A.2. The Yale Face Database – Case 2: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 3

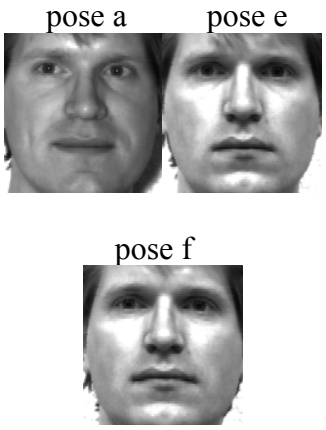


Figure A.3. The Yale Face Database – Case 3: two poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 4

Figure A.4. The Yale Face Database – Case 4: one pose of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 5

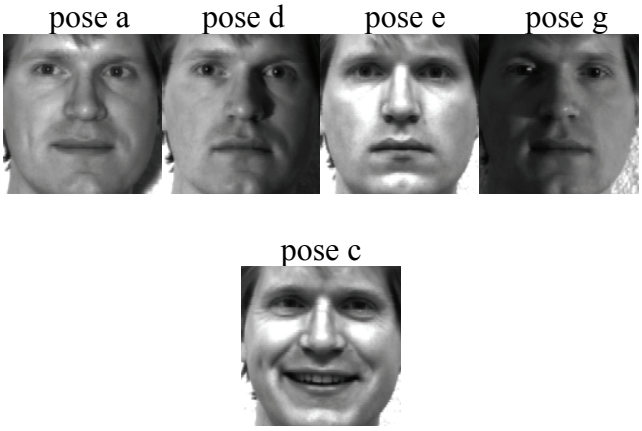


Figure A.5. The Yale Face Database – Case 5: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 6

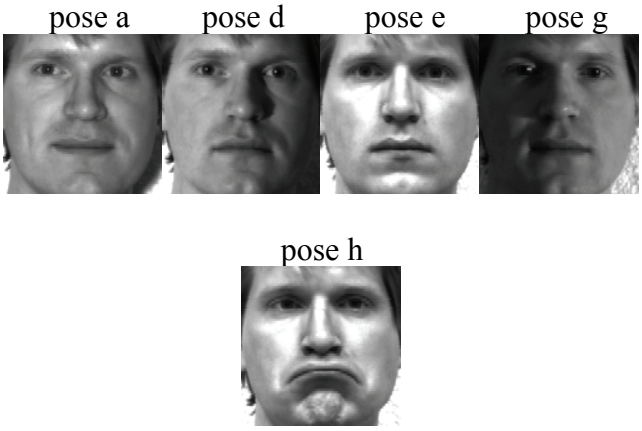


Figure A.6. The Yale Face Database – Case 6: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 7

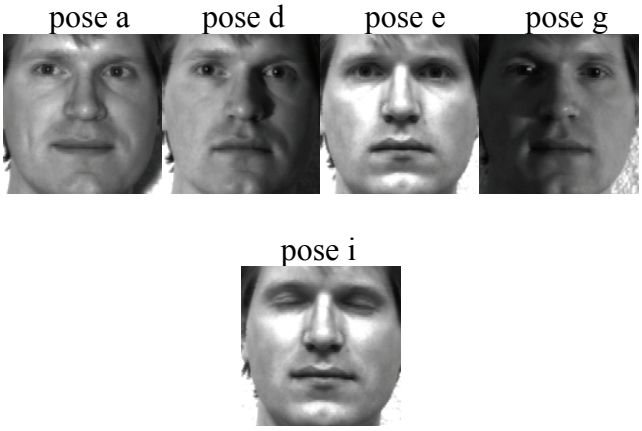


Figure A.7. The Yale Face Database – Case 7: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 8

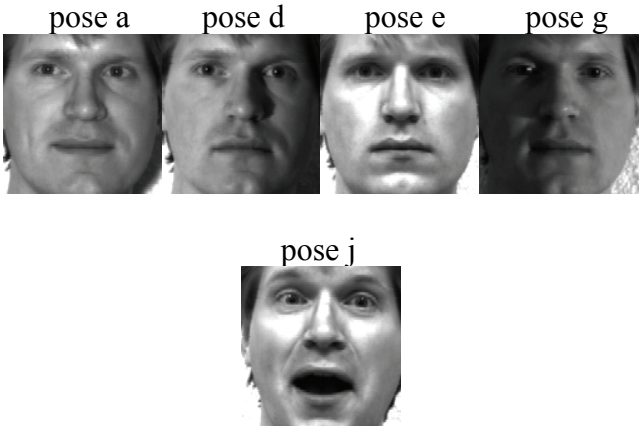


Figure A.8. The Yale Face Database – Case 8: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 9

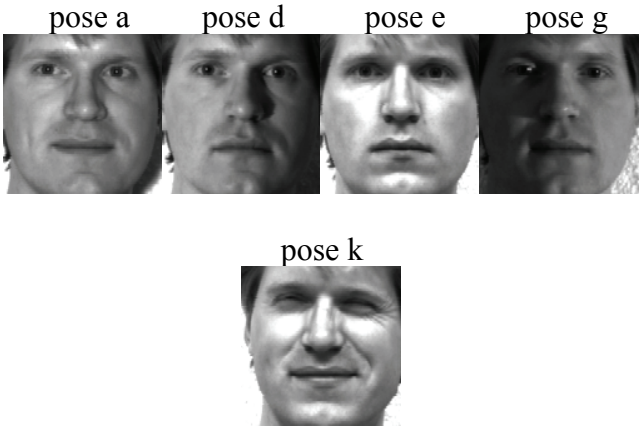


Figure A.9. The Yale Face Database – Case 9: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

Case 10

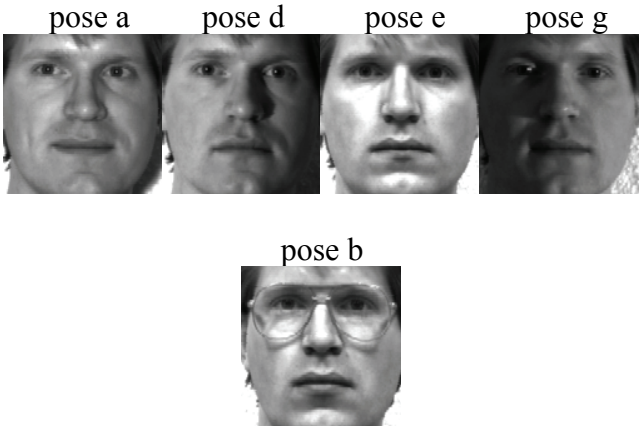


Figure A.10. The Yale Face Database – Case 10: four poses of one individual from training set (top row), one pose of the same individual from testing set (bottom row).

A.2 The Extended Yale Face Database B [106], [107]

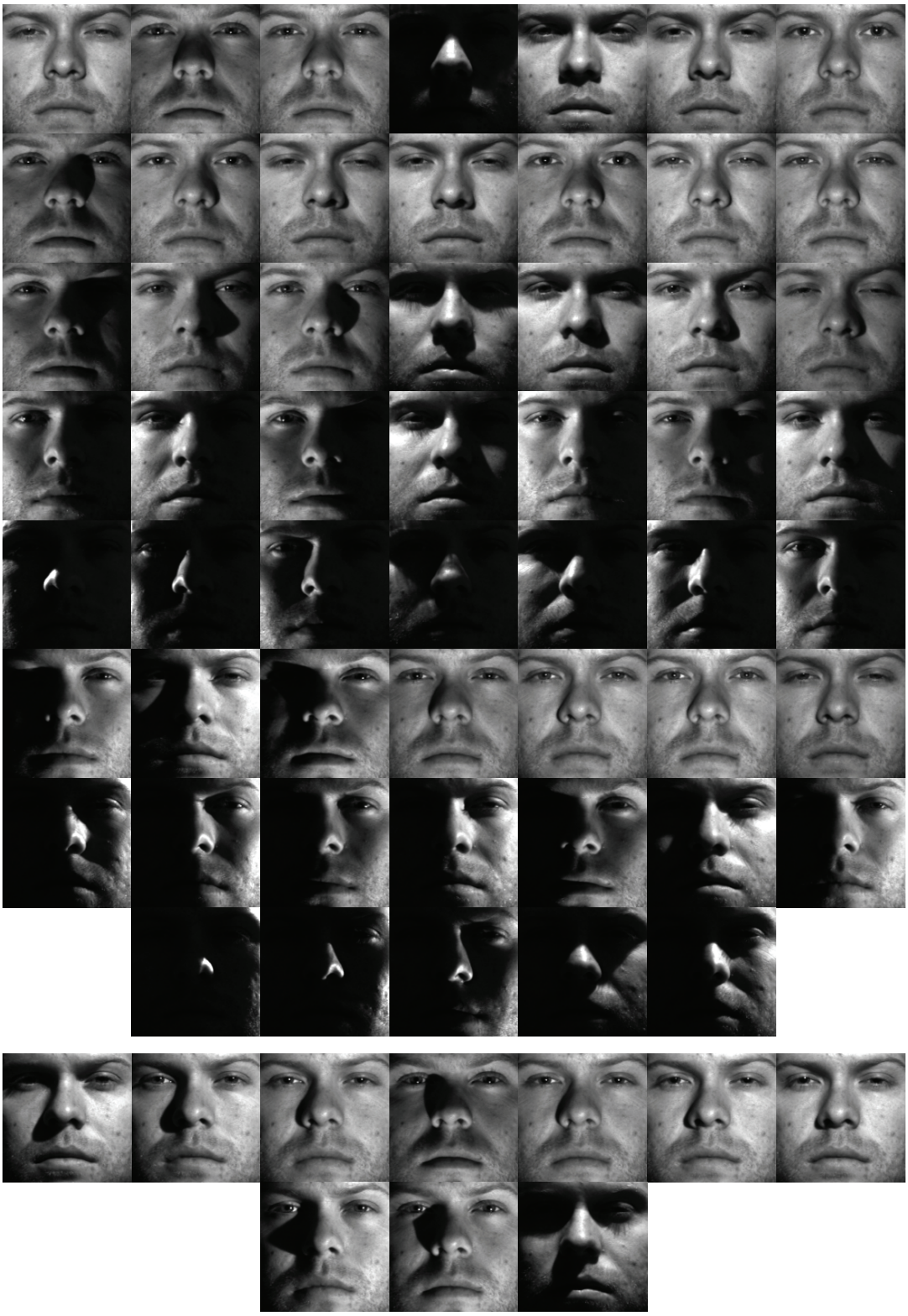


Figure A.11. The extended Yale Face Database B – Set 1: the poses included in training (top eight rows) and testing (bottom two rows) data sets.

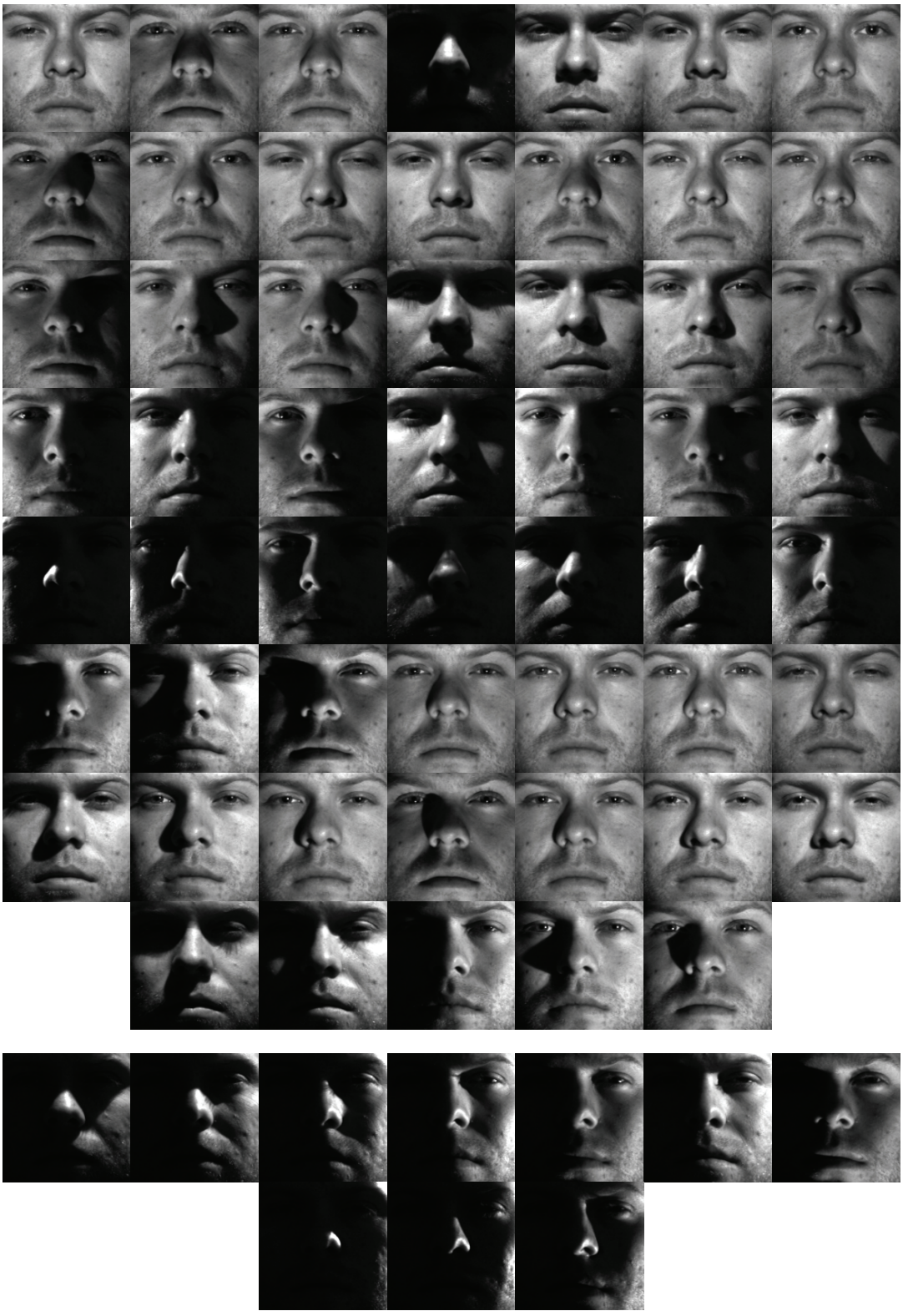


Figure A.12. The extended Yale Face Database B – Set 2: the poses included in training (top eight rows) and testing (bottom two rows) data sets.

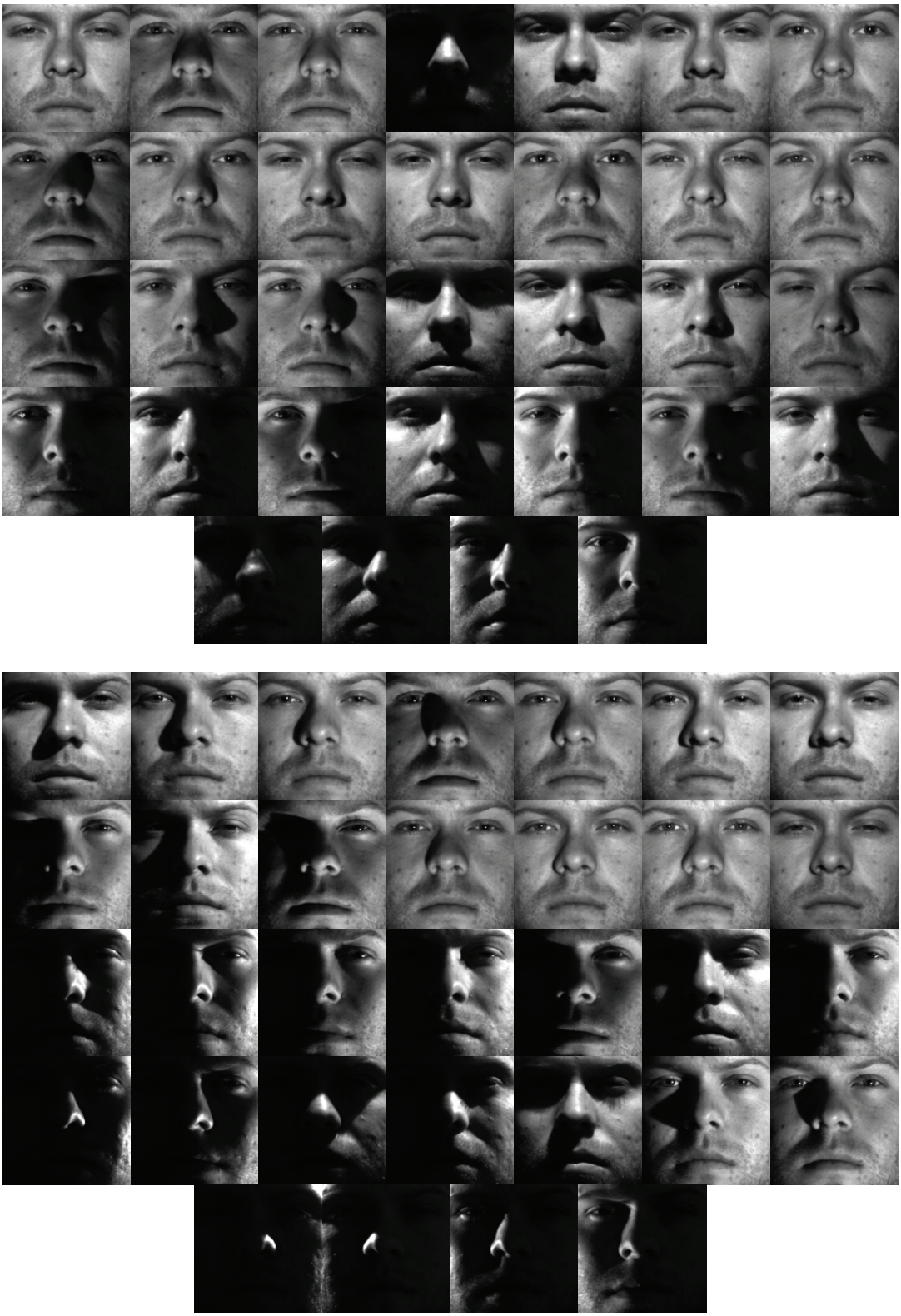


Figure A.13. The extended Yale Face Database B – Set 3: the poses included in training (top five rows) and testing (bottom five rows) data sets.

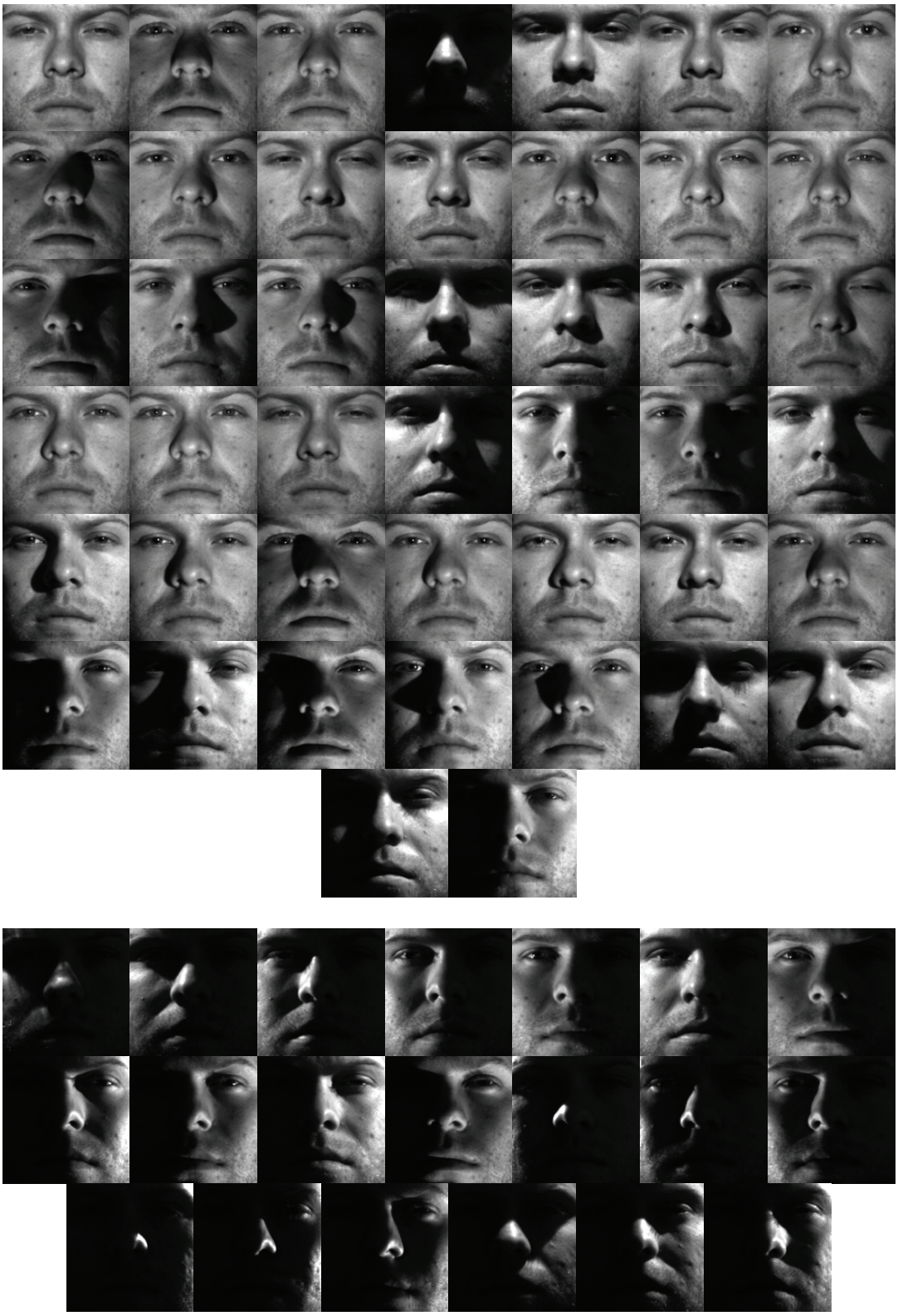


Figure A.14. The extended Yale Face Database B – Set 4: the poses included in training (top seven rows) and testing (bottom three rows) data sets.