

Face Recognition in Uncontrolled Environments

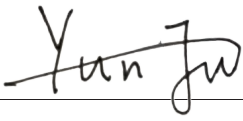
Yun Fu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
at
University College London.

Department of Computer Science
University College London

May 26, 2015

I, Yun Fu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.



Abstract

This thesis concerns face recognition in uncontrolled environments in which the images used for training and test are collected from the real world instead of laboratories. Compared with controlled environments, images from uncontrolled environments contain more variation in pose, lighting, expression, occlusion, background, image quality, scale, and makeup. Therefore, face recognition in uncontrolled environments is much more challenging than in controlled conditions. Moreover, many real world applications require good recognition performance in uncontrolled environments. Example applications include social networking, human-computer interaction and electronic entertainment. Therefore, researchers and companies have shifted their interest from controlled environments to uncontrolled environments over the past seven years.

In this thesis, we divide the history of face recognition into four stages and list the main problems and algorithms at each stage. We find that face recognition in unconstrained environments is still an unsolved problem although many face recognition algorithms have been proposed in the last decade. Existing approaches have two major limitations. First, many methods do not perform well when tested in uncontrolled databases even when all the faces are close to frontal. Second, most current algorithms cannot handle large pose variation, which has become a bottleneck for improving performance.

In this thesis, we investigate Bayesian models for face recognition. Our contributions extend Probabilistic Linear Discriminant Analysis (PLDA) [Prince and Elder 2007]. In PLDA, images are described as a sum of signal and noise components. Each component is a weighted combination of basis functions. We firstly investigate the effect of degree of the localization of these basis functions and find better performance is obtained when the signal is treated more locally and the noise more globally. We call this new algorithm multi-scale PLDA and our experiments show it can handle lighting variation better than PLDA but fails for pose variation.

We then analyze three existing Bayesian face recognition algorithms and combine the advantages of PLDA and the Joint Bayesian Face algorithm [Chen et al. 2012] to propose Joint PLDA. We find that our new algorithm improves performance compared to existing Bayesian face recognition algorithms. Finally, we propose Tied Joint Bayesian Face algorithm and Tied Joint PLDA to address large pose variations in the data, which drastically decreases performance in most existing face recognition algorithms. To provide sufficient training images with large pose difference, we introduce a new database called the UCL Multi-pose database. We demonstrate that our Bayesian models improve face recognition performance when the pose of the face images varies.

Acknowledgements

First and foremost, I deeply appreciate my supervisor Dr Simon Prince for his invaluable advice and continuous encouragement. It would have been impossible to finish this thesis without him. I will always remember his kindness. His patient and precise attitude to research has been invaluable to my endeavor. I am forever indebted to him for his help.

I would like to thank my parents Rensheng Fu and Huizhen Zeng, who supported me throughout my study. Without them I would not have the opportunity to study at University College London. I am forever in debt for their love.

Special thanks go to Professor Daniel Alexander for organizing the viva. He is the first academic staff I met at UCL and helped me obtain the opportunity to study the MSc in Computer Graphics, Vision and Imaging at UCL. I would like to thank my second supervisor Professor Simon Arridge for his suggestions.

Many thanks to all the staff and students in the Computer Science department at UCL for their great help, including Dr Gabriel Brostow, Professor Anthony Hunter, Professor Bernard Buxton, Dr Jan Kautz, Dr Simon Julier, Ms Melanie Johnson, Mr Ray Ojinnaka and Mr Bowei Chen. In particular, I would like to thank the members of the Prince Lab: Dr Alastair Moore, Dr Peng Li, Dr Jania Aghajanian, Mr Umar Mohammed and Dr Jonathan Warrell who read draft paper, give comments and provide support.

Finally, I would like to thank my examiners, Professor Roy Davies and Dr William Christmas, for their time and constructive feedback.

Contents

1	Introduction	9
1.1	What is Face Recognition?	9
1.1.1	Definition of Face Recognition	9
1.1.2	Typical Recognition Pipeline	9
1.1.3	Advantages of Face Recognition	9
1.1.4	Applications	10
1.2	Uncontrolled Environments	11
1.3	Challenges of Uncontrolled Environments	12
1.4	Problem Statement	13
1.5	Main Contributions	13
1.6	Report Structure	14
2	Literature Review	16
2.1	The General Model for Face Recognition	16
2.2	Overview of Existing Face Recognition Algorithms	17
2.2.1	Stage I (1964 - 1990)	18
2.2.2	Stage II (1991 - 1997)	18
2.2.3	Stage III (1998 - 2007)	23
2.2.4	Stage IV (2008 - present)	29
2.3	Face Databases and Performance Evaluation	37
2.3.1	Face Databases	37
2.3.2	Performance Evaluation	40
2.4	Conclusion	44
3	Investigating the Spatial Support of Signal and Noise in Face Recognition	46
3.1	Introduction	46
3.2	Related Works	48
3.2.1	Statistical Subspace Algorithms	48
3.2.2	Patch-based Face Representation Methods	50
3.3	Multi-scale PLDA	51
3.3.1	Learning	52

3.3.2	Inference	56
3.4	Experiments in Constrained Databases	57
3.4.1	Datasets and Preprocessing	57
3.4.2	Experiments for Frontal Lighting Data Set (XM2VTS)	58
3.4.3	Experiments for Illumination Variation Data Set (XM2VTS Lighting)	61
3.4.4	Experiments for Expression and Illumination Variation Data Set (Yale)	63
3.4.5	Experiments for Pose Variation Data Set (ORL)	65
3.5	Experiments in the Unconstrained Database	66
3.5.1	Dataset	66
3.5.2	Experiments Using Image Intensities	67
3.6	Conclusion	68
4	Joint Probabilistic Linear Discriminant Analysis for Face Recognition	70
4.1	Introduction	70
4.2	Bayesian Face Recognition Algorithms	73
4.2.1	The Bayesian Face Algorithm	73
4.2.2	PLDA	74
4.2.3	The Joint Bayesian Face Algorithm	76
4.2.4	Discussion	77
4.3	Empirical Comparison of the Joint Bayesian Face algorithm and PLDA	79
4.4	Joint PLDA	80
4.4.1	Motivation	80
4.4.2	Face Image Representation	81
4.4.3	Learning	82
4.4.4	Inference	82
4.5	Experiments in the Unconstrained Database	83
4.5.1	Preprocessing	83
4.5.2	Experiments Using Image Descriptors	83
4.5.3	Experiments Combining Multiple Image Descriptors	88
4.6	Conclusion	91
5	Tied Bayesian Face Recognition Algorithms for Pose Variation	93
5.1	Introduction	93
5.2	Performance of three Bayesian Face Recognition Algorithms under pose variation	94
5.3	Existing Face Recognition Algorithms Across Pose	97
5.3.1	Previous Work	97
5.3.2	Tied PLDA	98
5.4	Tied Bayesian Face Recognition Algorithms	99
5.4.1	The Tied Joint Bayesian Face Algorithm	99

5.4.2	Tied Joint PLDA	104
5.5	New Database	106
5.5.1	Motivation	107
5.5.2	The UCL Multi-Pose Face Database	108
5.6	Experiments	109
5.6.1	Data Preprocessing	109
5.6.2	Train and Test in the Multi-PIE Database	111
5.6.3	Train and Test in the UCL Multi-Pose Database	113
5.6.4	Train in the Multi-PIE Database and Test in the LFW Database	114
5.6.5	Train in the UCL Multi-Pose Database and Test in the LFW Database	116
5.6.6	Switching Mechanism	118
5.7	Conclusion	119
6	Conclusion	120
6.1	Summary and Contributions	120
6.2	Limitations and Future Work	122
	Glossary	124
	Bibliography	127

List of Figures

1.1	Face recognition pipeline	10
1.2	Images from controlled and uncontrolled environments	12
1.3	Main challenges for face recognition under uncontrolled Environments	13
2.1	Geometric parameters of the Kanade's face recognition algorithm	19
2.2	The face representation method of the Eigenfaces algorithm	20
2.3	Eigenfaces	20
2.4	Two subspaces of the Bayesian Face algorithm	21
2.5	Generation process of label graph in the EBGm algorithm	22
2.6	Two recognition approaches of the 3D morphable model based algorithm	25
2.7	Maximizing within-individual correlations improves pose invariance	26
2.8	Four illumination components that affect the appearance of face images	28
2.9	3D alignment method pipeline	30
2.10	Facial components	31
2.11	Calculation of similarity score	33
2.12	Attribute and simile classifiers	35
2.13	Frontalization of the appearance-prediction model	36
2.14	Visualization of the logistic discriminant base metric learning algorithm and the marginalized k-nearest-neighbour algorithm	37
2.15	An example of an ROC figure	42
2.16	An example of a CMC figure	43
3.1	Face generation process in the Multi-scale PLDA model	47
3.2	Signal exists locally and noise should be understand globally	48
3.3	Graphical model for multi-scale PLDA model	52
3.4	Structure of Multi-Scale PLDA model	53
3.5	Four face datasets	58
3.6	% Correct face identification for the XM2VTS frontal dataset	60
3.7	% Correct face identification for the XM2VTS frontal dataset	60
3.8	% Correct face identification for the XM2VTS lighting dataset	62
3.9	% Correct face identification for the XM2VTS lighting dataset	62

3.10	Two patch division methods	63
3.11	% Correct identification in the Yale database	64
3.12	% Correct identification in the ORL database	65
3.13	Several examples from the LFW database [65]	66
4.1	Comparing the training likelihood of the Joint Bayesian Face algorithm and PLDA at each iteration	79
4.2	Comparing the verification performance of the Joint Bayesian Face algorithm and PLDA at each iteration	81
4.3	Comparison of different combination approaches.	88
4.4	Performance comparison of four Bayesian face recognition algorithms when multiple image descriptors are combined	90
5.1	Several examples for pair groups in the LFW database	95
5.2	Several examples from the UCL Multi-Pose database	109
5.3	Structure of experiments	110
5.4	Several examples from the Multi-PIE database, the UCL Multi-Pose database, the LFW database.	111
5.5	Comparing three Tied Bayesian face recognition algorithms when trained and tested in the Multi-PIE database	112
5.6	Comparing three Tied Bayesian face recognition algorithms when trained and tested in the UCL Multi-Pose database	114

List of Tables

2.1	Summary of four development stages	18
2.2	Performance comparison among pose invariant face recognition algorithms.	27
2.3	Main papers in Stage IV	29
2.4	Main face databases at each development stage	38
2.5	Main evaluation methodology for each stage	43
3.1	% Correct results for the XM2VTS frontal data set	59
3.2	% Correct results for the XM2VTS lighting dataset	61
3.3	% Correct results for the Yale dataset	64
3.4	% Correct results for the ORL dataset	65
3.5	Face verification results in the LFW database using image intensities	68
4.1	Comparison of Bayesian face recognition algorithms	78
4.2	Performance of PLDA and Joint PLDA using LBP image descriptor	84
4.3	Performance of four Bayesian Face algorithms using LBP image descriptors	85
4.4	Performance of the three Bayesian Face algorithms for different image descriptors	87
4.5	Verification results using the SVM approach to combine four descriptors	90
4.6	Verification results using our own LBP descriptors	91
5.1	Performance of three Bayesian Face algorithms for different pose groups in the LFW database	96
5.2	Identity number and possible training pair number in each pose group of 10 training set of the LFW database.	107
5.3	Area under the ROC curve of figure 5.5	112
5.4	Area under the ROC curve of figure 5.6	114
5.5	Verification results when trained in the Multi-PIE database and tested in the LFW database	115
5.6	Verification results when trained in the UCL Multi-Pose database and tested in the LFW database	117
5.7	The effect of the switching mechanism	118

Publications

The following publications are related to this thesis:

Chapter 3

- Y. Fu and S. Prince. Investigating the spatial support of signal and noise in face recognition. In *ICCV workshop on subspace methods*, pages 131–138, 2009

Chapter 4

- P. Li, Y. Fu, U. Mohammed, J. Elder, and S. Prince. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, 2012

Chapter 1

Introduction

1.1 What is Face Recognition?

1.1.1 Definition of Face Recognition

The motivation of automatic face recognition is to give the computer the same capability as human beings to recognize faces. The general definition to face recognition is to estimate the identity of one or more people from static images or video sequences using a stored database of gallery faces. There are two types of face recognition: closed set and open set. In closed-set recognition, we can definitely find a gallery face matching the input image. In open-set recognition, we might not find any matched gallery image to the input face image.

1.1.2 Typical Recognition Pipeline

Face recognition is a visual pattern recognition problem. Figure 1.1 shows the basic pipeline of a face recognition algorithm. Generally the process for a computer to recognize faces can be divided into the following subtasks:

1. Face Detection. Detect the presence of faces and give the location, size and orientation of the faces in the image if faces exist. Normally the output is a bounding box around each face.
2. Face Alignment. Locate the facial landmarks, such as the eyes, nose, mouth, etc. and align the input face image to a pre-defined template to eliminate the size, location and orientation variation of face.
3. Feature Extraction. Describe a face image by a representation method.
4. Face Identification. Compare the similarity between the input image and all the faces in the gallery database and then estimate the identity of the input face image.

1.1.3 Advantages of Face Recognition

Face identification is a very common activity in everyday life. Everyone has to identify other people and prove their own identity to others. Examples include showing a passport to open a bank account and inputting a password to login on a computer. In most cases we rely on traditional identification methods which include identification cards, keys, passwords, etc. However, these methods are not necessarily

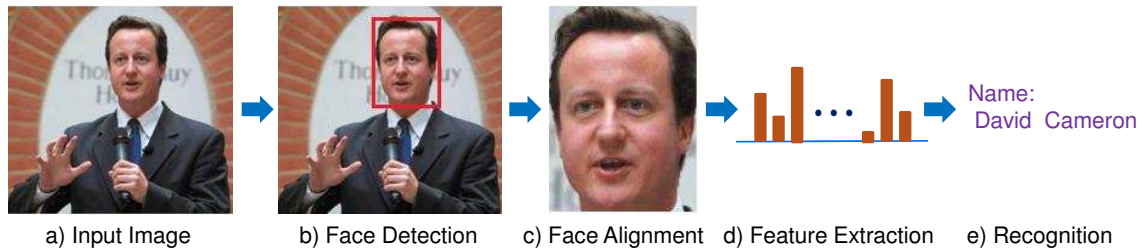


Figure 1.1: **Face recognition pipeline.** a) Given an input face image. b) We detect the presence of a face and put a bounding box around it. c) We crop out the face region and align it to a pre-defined template to compensate for size, location and orientation variation. d) We extract features. e) We recognize the identity of the face.

safe or convenient: identification cards and keys might be counterfeited; passwords might be forgotten and stolen; cards and keys are not easy to carry. Therefore, a more secure and convenient method is desirable. It is widely believed that biometrics are the ideal solution.

The term ‘biometric’ means to use one or more intrinsic physical or behavioral traits to recognize people. Because these biometric traits are unique and part of the individual, they are difficult to counterfeit or steal. Biometrics are believed to be reliable, practical and convenient. There are different kinds of biometric characteristics to identify people, for example iris, fingerprint, DNA, palm print, voice, gait, etc. Among these, the face is the most important characteristic to recognize people.

Compared with other biometrics, face recognition has the following advantages:

- Natural. The face is the most natural way to identify a human being. Compared with fingerprint and iris recognition, it is easier for normal users to get involved.
- Ideal for surveillance. Face recognition does not need the participant’s cooperation, so security cameras can be installed secretly. This is especially useful for investigating criminals. It is the biggest advantage of face recognition compared to other biometrics.
- Easy to be accepted. There is no direct contact when acquiring the face image, so normally it will be unobtrusive.
- Cheap and widely-distributed image acquisition equipment. Current CMOS cameras are very cheap. The webcam has already become a standard external device and CCTV cameras are installed in many companies and cities. Many people own digital cameras, camcorders and photo-scanners.

Because of the above advantages, face recognition has become a very popular research topic in the past twenty years.

1.1.4 Applications

Face recognition has great potential in numerous government and commercial applications. Generally these applications can be classified into the following categories:

- **Access Control:** computer login and building access. Face recognition can prevent misuse of stolen or lost passwords and keys effectively. The recognition accuracy in this type of application is quite high because the number of people is relatively small and input images are normally frontal face under indoor illumination. For example, Omron [131] provided a face recognition system to the University of Missouri-Rolla to secure a nuclear reactor.
- **Security.** Face recognition is often combined with a smart card to confirm a user's real identity. The organizers of Beijing Olympics installed a face recognition system developed by Authenmetric to make sure only the valid ticket holders can enter the sport venues in 2008 [1].
- **Surveillance.** Many airports have installed face recognition systems to identify known terrorists. However, false alarms are quite high for most current face recognition systems. For example, a face recognition system developed by Viisage was deployed in Fresno Yosemite International airport in California in 2001. However, they finally gave up the system because of frequent false alarms [87].
- **Human Computer Interaction and entertainment.** The human body is a natural input device to achieve user-friendly and efficient human computer interaction. The Xbox 360 Kinect developed by Microsoft can make users' avatars simulate their talking style during the game.
- **Law enforcement.** Face recognition could help investigators obtain the identity of a person from a face database quickly. For example, a face recognition system called Imigis helps California's police officers identify unknown bodies.
- **Labeling face images.** It has become more and more difficult to label images manually as the number of images increases. Face recognition can be used to label images automatically. Picasa developed by Google uses face recognition technology help its users manage their photos efficiently.
- **Video Management.** Human faces appear very frequently in news, films and home video. In order to generate summaries from these videos for video browsing, skimming and summarization, face recognition technology is often used. For example, a software developed by Ma and Zhang can collect a set of video segments from original video files by using face recognition technology [91].

1.2 Uncontrolled Environments

In the past decades people focused on developing fundamental face recognition algorithms [132] [10] based on controlled environments which have simple backgrounds and limited variation in pose and lighting. To compare the performance of face recognition algorithms, a number of standard face databases were published, for example FERET [106], XM2VTS [95] and Multi-PIE [56]. Images from controlled databases are illustrated in Figure 1.2a. After years of development many proposed face recognition algorithms have produced very impressive results in these controlled databases.



Figure 1.2: **Images from controlled and uncontrolled environments.** a) In the past, face recognition algorithms were evaluated on controlled face databases in which images have simple backgrounds and limited variation in pose and lighting. From left to right, example images are from XM2VTS [95], Yale[52], Multi-PIE [56] and FERET [106] face databases respectively. b) Recently research has shifted to face recognition in uncontrolled environments in which images have complex backgrounds, partial occlusions and large variations in pose, lighting and race. Example images are from the most famous uncontrolled face database: Labeled Faces in the Wild [65].

Recently, research has shifted toward face recognition in uncontrolled environments to encourage real-world applications. Images are collected from the internet and have complex backgrounds, partial occlusion and large variations in pose, lighting, image quality, race and expression. The most famous uncontrolled face database is the Labeled Faces in the Wild database with over 600 citations in the face recognition literature [126]. Examples from this database are shown in Figure 1.2b. An ideal face recognition algorithm should perform well in uncontrolled environments to satisfy the requirements of real-world applications. However, this still remains a big challenge for most current face recognition algorithms.

1.3 Challenges of Uncontrolled Environments

The three main challenges for face recognition in uncontrolled environments are large variation in pose, lighting and partial occlusion.

The first challenge is pose variation. A person appears very differently from different viewpoints (see Figure 1.3a). Pose variations make the feature matching between two face images under different pose very difficult. In general, non-matching frontal faces are more similar to each other in terms of pixel values than matching faces of different poses.

A second major obstacle is lighting variation (Figure 1.3b). It is hard to recognize the face under varying lighting. Even two images from the same person but under different lighting can appear dramatically different.



Figure 1.3: **Main challenges for face recognition under uncontrolled Environments.** a) Pose variation: the appearance of a face varies significantly as the position of the camera varies. b) Lighting variation: the face looks very different when lighting changes. c) It is hard to recognize the face when face expression varies. All example images are from the Labeled Faces in the Wild database [65].

Finally, expression is also an impeding factor (Figure 1.3c). Varying face expression can reduce recognition performance dramatically.

In this thesis we mainly focus on overcoming these challenges to improve the face recognition performance in uncontrolled environments.

1.4 Problem Statement

Face recognition in uncontrolled environments is a challenging task and many existing algorithms do not perform well. In this thesis we propose a series of robust generative probabilistic face algorithms which can handle the challenges of uncontrolled environments. To verify the performance of our algorithms, we test our algorithms in the well-known uncontrolled face database, Labeled Faces in the Wild [65].

1.5 Main Contributions

In this report we discuss how to overcome the main challenges for a reliable face recognition system under uncontrolled environments. The main contributions are:

1. We review existing face recognition algorithms. We review the history of face recognition research by dividing it into four development categories. We list the main problems and representative methods in each category. We also summarise the main publically accessible face databases and describe the evaluation methods and conclusions of famous Face Recognition Technology Test (FERET) and Face Recognition Vendor Test (FRVT).
2. We investigate the role of the spatial support of signal and noise for face recognition. We develop a model for face recognition that describes the image as a sum of signal and noise components. We describe each component as a weighted combination of basis functions. We investigate the effect of degree of localization of these basis functions: each might describe the whole image (describe global pixel covariance) or only a small part of the face (describe only local pixel covariance). We called this new algorithm Multi-Scale PLDA. Our experiments show that we can extract a more robust recognition signal from face images and produce better performance by treating the signal more locally and the noise more globally.
3. We analyze three existing Bayesian face recognition algorithms and propose a new algorithm: Joint PLDA. Probabilistic linear discriminant analysis (PLDA) [111] and the Joint Bayesian Face algorithm [30] are two state of the art face recognition algorithms. We compare the two algorithms to identify their similarities and differences. Then we combine the advantages of PLDA and the Joint Bayesian Face algorithm to propose Joint PLDA. We compare the performance of four Bayesian face recognition algorithms (The Bayesian Face algorithm, PLDA, the Joint Bayesian Face algorithm and Joint PLDA) when different image descriptors are used. Our experimental results demonstrate that Joint PLDA performs better than PLDA and the Joint Bayesian Face algorithm in the LFW database.
4. We identify the challenge in the LFW database and propose two new algorithms to overcome the challenge. We analyse the verification results of three Bayesian face recognition algorithms in the LFW database and find that large pose variability is the challenge for improving performance. Tied PLDA [82] is one possible solution to overcome this problem. However, there are insufficient LFW training images for Tied PLDA, especially where there is a large pose difference. To address this issue, we introduce a new database called the UCL Multi-pose database with more training images for large pose changes. We also describe tied version of the Joint Bayesian Face algorithm and Joint PLDA. We compare performance of three Tied Bayesian face recognition algorithms (Tied PLDA, Tied Joint Bayesian Face algorithm and Tied Joint PLDA) when different image descriptors are used. Our experiments show Tied Bayesian face recognition algorithms perform better than Bayesian face recognition algorithms (PLDA, the Joint Bayesian Face algorithm and Joint PLDA) when large pose variation exists.

1.6 Report Structure

In chapter 2 we describe previous related work. In chapter 3, face generation is divided into signal and noise components and we investigate the optimal spatial support for these two components. In

chapter 4, we compare the existing three Bayesian face recognition algorithms and propose Joint PLDA to combine the advantages of PLDA and the Joint Bayesian Face algorithm. In chapter 5, we propose Tied Joint Bayesian Face and Tied Joint PLDA to improve the performance for large pose variation. In the final chapter, we draw conclusions and describe future work.

Chapter 2

Literature Review

In this chapter we will analyse the generation process of face images and discuss the general model for face recognition. Then we divide the development of face recognition into four historical stages and introduce the main targeted problems and face recognition algorithms at each stage. Finally, we discuss publicly-accessible face databases and evaluation methods.

2.1 The General Model for Face Recognition

The generation of face images can be described as follows: the light interacts with the face through physical processes such as reflection and then the charge-coupled device (CCD) of the camera captures the reflected light to form the pixel intensity for each pixel location [138]. Therefore, the process includes three factors:

1. The intrinsic structure of the face. It includes the 3D shape of the face, the reflectance of the face surface (texture) and variations caused by expression.
2. External factors. These include the luminance and direction of the light source.
3. The parameters of the camera. These include the location, focus, shutter speed and aperture size of the camera.

From the face image generation process we know the 3D structure of a face and its reflectance characteristics are the intrinsic features of the face which can be used to identify people. Obviously, face expressions cannot be used to distinguish different people although they belong to the intrinsic feature of the face. Clearly, the external factors and the camera parameters cannot be used to discriminate between identities. Consequently, the intrinsic structure of the face is called ‘the signal’ and can be used to estimate the identity of people. The other factors are called ‘the noise’ and are not useful for face recognition.

The ideal face recognition algorithm can divide the face image into two parts: signal (stable intrinsic structure of the face) and noise (expression, external conditions and camera parameters). Then we identify faces based on the extracted signal. So the process to identify people from an input face image Υ is as follows

1. Image decomposition. We decompose a face image into stable intrinsic structure of face, light source and parameters of camera.
2. Feature extraction. We extract discriminant features \mathbf{s} from the stable intrinsic structure of the face.
3. Identification. We compare the features \mathbf{s} of the input face image Υ with the features $\{\mathbf{s}_j\}_{j=1}^J$ of all the J gallery face images to identify the input face image by the gallery image with the maximum similarity

$$\hat{\mathbf{h}} = \arg \max_{j \in N} (Sim(\mathbf{s}, \mathbf{s}_j)), \quad (2.1)$$

where the term $\hat{\mathbf{h}}$ denotes the identity of the face, the function Sim calculates the similarity score, and N is the number of images in gallery face database. Here we assume we can definitely find a gallery face matching the input image.

In fact the process to determine the 3D shape and the reflectance of face is a very difficult vision problem even when there is only a single point light source [150]. At the present time it is still an unsolved problem although researchers made some process by using different kinds of constraints and priors [152] [17] particularly in the case where there are multiple images under different illuminations [52] [147]. Therefore, most current face recognition algorithms do not decompose the face image to obtain 3D shape and the reflectance of face but extract the discriminant features from the image directly.

For local feature-based face recognition algorithms such as the Elastic Bunch Graph Matching algorithm [139], the feature \mathbf{s} comprises local statistics (geometric and appearance) extracted from facial landmarks, such as the eyebrows, eyes, nose, mouth, etc. For holistic subspace methods such as the Eigenfaces algorithm [132], the Fisherfaces algorithm [10] and Probabilistic Linear Discriminant Analysis (PLDA) [111], the feature \mathbf{s} is a point in a low dimensional subspace.

2.2 Overview of Existing Face Recognition Algorithms

Research in face recognition goes back to 1965 with the work of Chan and Bledsoe [28]. Since then, face recognition has become more and more popular, especially after the Eigenfaces algorithm [132] was published in 1990. It is likely that face recognition will become more widespread as potential applications have extended from traditional security applications to the areas of human-computer interaction, electronic entertainment and social networking.

After decades of development there is a huge literature concerning face recognition. To describe the development more clearly, we divide research history into four historical stages according to the targeted problems. Table 2.1 summaries the four stages. We now describe each stage in turn.

Stages	I	II	III	IV
Targeted Problems	Use geometry methods to do face recognition	Face recognition under controlled environments	Pose and lighting variation under controlled environments	Face recognition under uncontrolled environments
Main Achievement	The first face recognition paper and the first PhD thesis was proposed	Subspace algorithms were proposed and automatic face recognition become possible; The first commercial face recognition system was proposed	The performance of face recognition is improved when pose varies and lighting changes; Many commercial face recognition systems are developed and three FRVT tests are organized to compare these products.	There has been significant performance improvement for face recognition under uncontrolled environments; The commercial applications of face recognition have been expanded to social networks, electronic entertainment, and online search
Limitations	The algorithms cannot recognize people without human intervention	Recognition performance decreases significantly when pose and lighting variation exists	The algorithms perform badly under uncontrolled environments	A performance gap still exists when comparing to human accuracy
Main Algorithms	The first face recognition paper [73], The first recognition PhD thesis [71]	Eigenfaces [132], Bayesian Faces [97], Fisherfaces [10], EBGM [139]	3D Morphable model [17], Eigen Light Fields [54], Tied PLDA [82], Quotient image [122], Illumination cones [52]	Nowak similarity learning[101], Attribute and Simile Classifiers [76], Multi-shot [127], PLDA [111], Deepfaces [128]

Table 2.1: Summary of four development stages.

2.2.1 Stage I (1964 - 1990)

In this stage researchers focussed on extracting geometric features of different people to distinguish individuals. Most methods were purely geometric. For example, Kelly [73] used the width of the head, the distances between the eyes and from the eyes to the mouth to identify people in 1971. Two years later, Kanade [71] proposed a method which used distances and angles between the eye corners, the mouth extremal, the nostrils and the chin top (see Figure 2.1). Because these distances have to be extracted manually, automatic face recognition is not practical in this stage.

2.2.2 Stage II (1991 - 1997)

This stage is quite short but very important because a number of very important algorithms were proposed. Moreover, during this period, the Department of Defense of American government sponsored George Mason University to collect face images for the Face Recognition Technology (FERET) database and organized three famous tests [106] [115] [110]. The first commercial face recognition systems were also set up during this period (e.g. FaceIt).

In 1991 Turk and Pentland proposed the Eigenfaces algorithm [132] which is the most well-known algorithm in this stage. Many of the subsequent algorithms were variations of the Eigenfaces algorithm. Nowadays, the Eigenfaces algorithm has become the benchmark algorithm for face recognition evaluation.

The motivation behind the Eigenfaces algorithm is that natural images such as face images have significant statistical redundancy. Principal Component Analysis (PCA) can be applied to reduce the

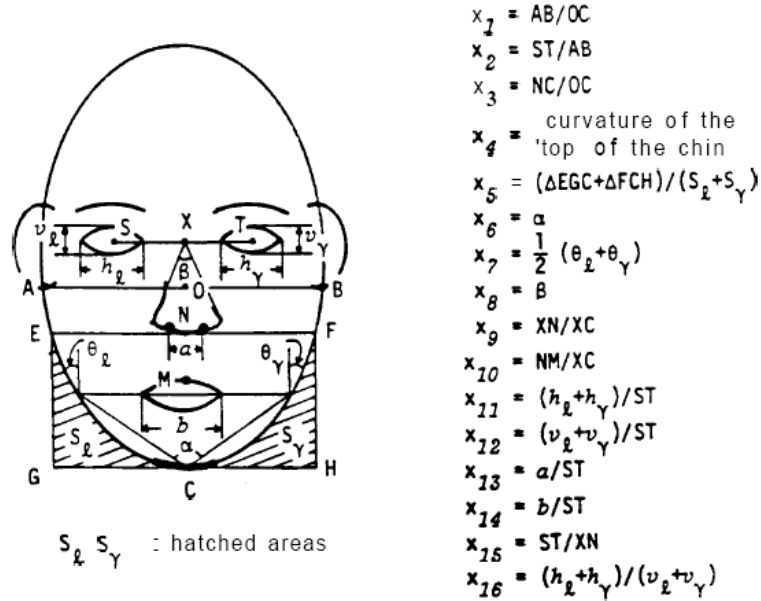


Figure 2.1: **Geometric parameters of the Kanade's face recognition algorithm [71]**. Kanade extracted 16 geometric parameters $x = \{x_1, \dots, x_{16}\}$ manually from each face and used them to identify people. (Adapted from Kanade [71])

dimensions to form a more compact representation to face images. Using this approach the signal-to-noise ratio can be increased.

In the Eigenfaces algorithm a face image x is represented by the following equation:

$$x \approx \mu + \Phi\omega, \quad (2.2)$$

where x is a pixel intensity vector obtained by concatenating the columns of pixels in the image Υ (shown in Figure 2.2), μ is the mean vector of all the training images, Φ contains the basis functions of the feature subspace in its columns, and ω is a coefficient vector.

In the training phase of the Eigenfaces algorithm, the goal is to learn the basis functions of the feature subspace. Firstly the mean image vector μ is subtracted from each of the training images. The resulting vectors are concatenated to form a $n \times m$ matrix \mathbf{B} , where n denotes the vector dimensions and m is the number of training data. Then Principal Component Analysis (PCA) is applied to the covariance matrix $\mathbf{B}\mathbf{B}^T$ to obtain m eigenvectors. However, to have a compact representation only p eigenvectors with the largest eigenvalues will be chosen from m eigenvectors. The subspace spanned by p eigenvectors is called feature space. The 4 eigenvectors with the largest eigenvalue are reshaped to form RGB images, which are shown in Figure 2.3. Each training image can be represented by a corresponding point in the feature space.

In the testing phase we assign identity to input images. Each input image is projected into feature space and the Euclidean distance is measured to all the training images in the feature space. If the distance is smaller than a certain threshold, the input image is assigned to the same identity as the closest training image in feature space.



Figure 2.2: **The face representation method of the Eigenfaces algorithm [132].** In the Eigenfaces algorithm the input face image Υ is represented by an intensity vector \mathbf{x} , which is obtained by concatenating the columns of image pixels.

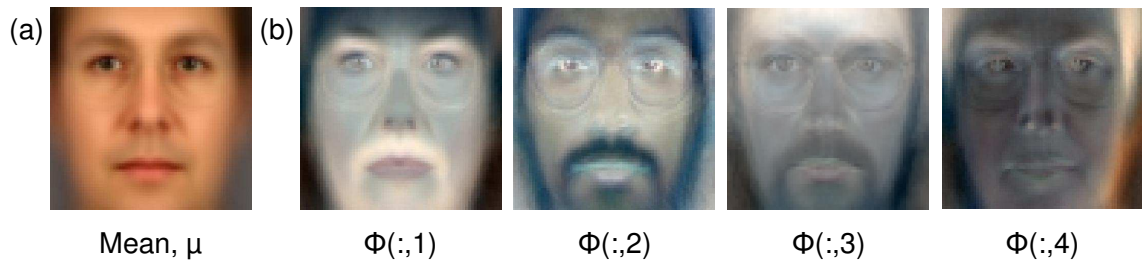


Figure 2.3: **Eigenfaces [132].** (a) Mean face. (b) Four eigenvectors with the largest eigenvalue are reshaped to form RGB images.

After the Eigenfaces algorithm was proposed, there was great interest in comparing these new appearance-based subspace algorithms with traditional geometry-based algorithms, which were widely used in Stage I. In 1993 Brunelli and Poggio [20] conducted a comparison experiment and drew the conclusion that appearance-based subspace algorithms produce better performance than geometry-based algorithms. Their conclusion drove researchers away from geometry-based algorithms and made appearance-based algorithms dominant.

One drawback of the Eigenfaces algorithm is that it only extracts global facial features but cannot use local features to describe local facial structures. However, representations to the local facial structure can offer robustness against within-individual variation. Atick et al. [103] proposed the local feature analysis (LFA) algorithm to overcome this drawback in 1996. The LFA algorithm represents face images in terms of statistically derived local features. They demonstrated the LFA algorithm produced better discriminant performance than the Eigenfaces algorithm. The LFA algorithm was commercialized and became the well-known FaceIt system.

Inspired by the Eigenfaces algorithm, Moghaddam et al. [98] proposed a Bayesian probability-based algorithm which measures the similarity of two face images by Bayesian probability instead of Euclidean distance. They define two subspaces to describe two types of image variation: within-individual variation and between-individual variation. The pixelwise difference of two face images is projected into within-individual subspace to obtain the within-individual probability density and the between-individual



Figure 2.4: **Two subspaces of the Bayesian Face algorithm [98].** The Bayesian Face algorithm defines two subspaces to describe two types of image variation. (a) Four directions in the between-individual subspace. Images look like different people. (b) Four directions in the within-individual subspace. Images appear to be from the same person.

subspace to obtain between-individual probability density respectively. Figure 2.4 illustrates the two subspaces. Then the maximum a posteriori (MAP) approach is used to estimate which type of variation is the main reason for the image difference. If the difference is caused mainly by the within-individual variation, then two images are assumed from the same person. If the between-individual variation is the main reason, then two images do not match. In the FERET 2000 [110] this new Bayesian probability based algorithm produced better performance than the Eigenfaces algorithm.

The Fisherfaces algorithm proposed by Belhumeur et al. [10] is another well-known algorithm that exploits between- and within- individual statistics. The Eigenfaces algorithm maximizes the scatter of all face images by projecting the high-dimensional image into a low dimension subspace. Thus the Eigenfaces algorithm not only maximizes the between-individual scatter which is important for classification but also to the within-individual scatter that should be eliminated. Unwanted within-individual variations due to noise are retained. The Fisherfaces algorithm applied Fisher's linear discriminant analysis (LDA) to project images into a low dimensional subspace, which maximizes the between-individual scatter and minimizes the within-individual scatter simultaneously. In this way the Fisherfaces algorithm obtains a more optimal subspace and performs much better than the Eigenfaces algorithm when there is lighting and expression variation in face images.

Linear discriminant analysis was a well-known classification method but it was difficult to use until the Fisherfaces algorithm was proposed. The main reason is that the within-individual scatter matrix of LDA becomes singular when the number of images from a person is less than the number of pixels in the image. In fact, this situation is present in nearly all face databases. To overcome this problem the Fisherfaces algorithm first uses PCA to reduce data dimensionality and this makes the application of LDA possible.

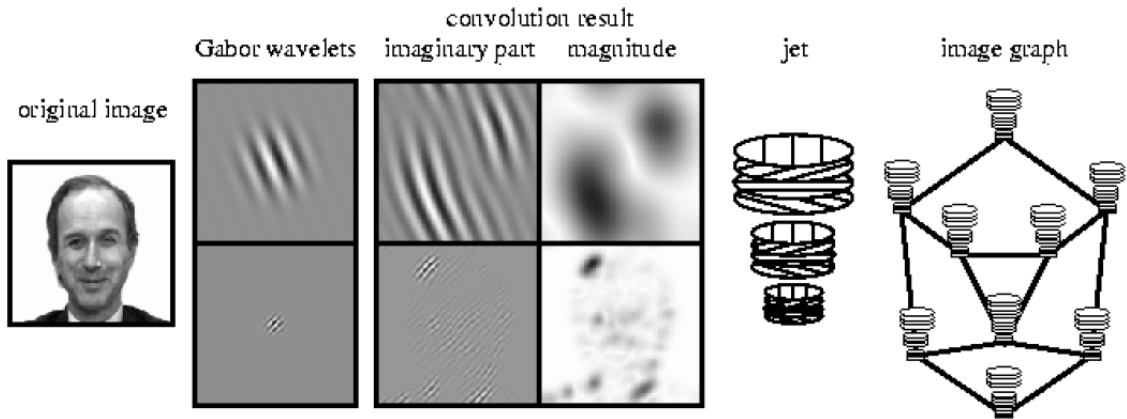


Figure 2.5: **Generation process of label graph of the EBG algorithm [139].** Firstly, the input image is processed by the Gabor wavelets as shown in the middle image. Then the corresponding label graph as shown in the right image is generated. (Adapted from Wiskott et al. [139])

Although the most mainstream face recognition algorithms in Stage II were subspace-based algorithms, Wiskott et al. [139] proposed the elastic bunch graph matching (EBGM) algorithm to use information from some local facial landmarks instead of the whole face image to do face recognition. In their algorithm a face image is described by a graph which includes N nodes and E edges. The nodes correspond to fiducial points, which are a set of salient facial points and usually located on corners of the eyebrows, the corners of the eyes, tip of the nose, and corners and outer mid points of the lip. The pixels around each node are processed by Gabor Wavelets. Each edge represents the geometric relationship between two nodes.

In training each gallery image is represented by a graph. Figure 2.5 demonstrates the process to generate a labeled graph for a face image. In test an input image is firstly processed to generate a graph to represent the new image and then we compare the graph of the input image with the graph of each gallery image. The input image is assigned the identity of gallery image which has the most similar graph. The advantage of this algorithm is that it considers the global structure and the local features together. The main disadvantage is that this algorithm requires good alignment and accurate localization of the fiducial points.

During Stage II the Face Recognition Technology Test (FERET) sponsored by the Department of Defense of the American government played a important role to encourage the improvement of face recognition algorithms. The target of the FERET project was to develop a reliable face recognition system for the American government. The project included three parts: sponsoring research on face recognition, constructing the FERET face database and organizing performance evaluations. They arranged three evaluations in 1994, 1995, and 1996 respectively. The evaluations record the development of face recognition but also indicate the drawback of Stage II algorithms: the performance of face recognition fails when large pose and lighting variation exists [110]. The test report guided researchers into the third stage to propose new algorithms to solve the two problems.

To conclude, face recognition developed very quickly and automatic face recognition became prac-

tical in Stage II. The proposed algorithms produced good performance when users cooperate, pose is frontal, lighting is controlled, and the size of databases is relatively small. However, they fail when large pose variation and lighting variation exists.

2.2.3 Stage III (1998 - 2007)

According to the evaluation results of FERET 1996 pose and lighting variation were the main challenges for face recognition systems [110]. Therefore, methods to handle pose and lighting variation became popular in this stage. Moreover, the development of real time face detection and face alignment made commercial applications of face recognition systems more practical. Many commercial face recognition systems were produced. Subsequently, the American government organized three evaluations to the commercial face recognition systems in 2000 [106], 2002 [109] and 2006 [105] respectively. In the following text we will introduce the development of face recognition in this stage by four sub-tasks: face detection, face alignment, pose invariant face recognition, and illumination invariant face recognition.

Face Detection

In 2001 Viola and Jones [133] proposed an AdaBoost-based face detection algorithm which was the first algorithm to achieve real-time high-quality face detection. Their algorithm could detect frontal faces at a speed of five frames in a second. Their main contributions include: using simple features which can be computed very fast; weighting multiple weak classifiers to form a final strong classifier by AdaBoost method; applying a cascade method to improve detection speed.

Face Alignment

The output of face detection algorithms is normally a rough bounding box around each face. We require an automated alignment method to align the detected face image to a pre-defined template to compensate for variation of size, rotation and location.

Flexible models [77] played a important role in automatic face alignment. These include active shape models (ASM) [33] and active appearance models (AAM) [46] [34]. The AAM algorithm is the extension to the ASM algorithm. The ASM algorithm only models the shapes of face images whereas the AAM algorithm models the shapes and textures of face images. The AAM algorithm firstly applies PCA to model the shape and the 2D texture separately and then combines the two models to obtain a set of unified appearance parameters which describe shape and 2D texture synchronously. The AAM algorithm can be used to align face images. It can also be used to synthesize model faces, locate fiducial points and recognize faces [94].

The AAM algorithm has a drawback in that a number of fiducial points are required to be manually labeled in the training phase. In 2004 Learned-Miller [79] proposed an unsupervised face alignment algorithm called ‘congealing’ to overcome this problem. The principle of congealing is to apply affine transformations to a set of face images to make them look similar. Congealing performs very well in aligning binary images, such as binary handwritten digits and magnetic resonance image volumes. However, it fails for complex real world images. Huang et al. [64] extended Learned-Miller’s work to align real world images by using SIFT descriptors [88] instead of pixel intensities. Their method demonstrated good performance to align real-world images. Later Cox et al. [37] extended Huang’s

work to speed up alignment by using sum of squared error instead of entropy to measure the image similarity. However, the congealing algorithm has a drawback that it only uses affine transformations and cannot deal with pose variations.

Pose Invariant Face Recognition

Depending on the type of gallery and probe images, algorithms addressing pose variation can be classified into two categories: multi-view face recognition and face recognition across pose. Multi-view face recognition algorithms [14] [83] [104] compare probe and gallery images at the same pose using the same methods as frontal face recognition. Therefore, these algorithms are simple extensions of the existing frontal face recognition algorithms. For face recognition across pose, the viewpoint of the probe images are different from the gallery images, so it is more difficult. In this part, we focus on face recognition across pose. Generally there are two types of algorithms to solve face recognition across pose: the 3D model based algorithms and the 2D statistically based algorithms.

Early three-dimensional algorithms [50] use several face images at different poses but from the same individual to generate a 3D model of each gallery individual's head and then compare the input image with a re-rendered gallery image at the same pose as the input image. Here, gallery images are the images with known identities to a face recognition system and probe images are the images presented to the system for recognition. The drawback of this type of algorithm is that it requires multiple images for a individual and it is not practical for many face databases. In 2003 Blanz et al. [16] [17] [15] proposed a morphable model based algorithm which only requires a single face image to construct a 3D model. Their algorithm provides two distinct methods to do face recognition. The first method is to re-render the frontal view of the probe image. Recognition is performed by comparing the transformed frontal probe image with each frontal gallery image. Figure 2.6 (a) shows the pipeline of the first method. In the second method, 3D model coefficients are estimated for the probe and gallery images respectively. Recognition is performed directly by comparing the coefficients of the input image and each gallery image. The second method is demonstrated in Figure 2.6 (b). Their experimental results show the performance of the first method is better than the second method at some viewing angles but overall there is not much difference. This 3D morphable model algorithm achieved 87% correct in a database which includes 87 people with pose variation of up to $\pm 45^\circ$. Unfortunately, it is very slow to estimate 3D coefficients of an image in practical applications and any noise in the face image often makes the estimation to the 3D coefficients inaccurate.

Two-dimensional statistical models treat the transformation between frontal and non-frontal images as a learning problem. Vasilescu et al. [143] presented an algorithm in which an unseen view image of the person can be generated. Later, Gross et al. [54] proposed the Eigen Light Fields algorithm which treats pose invariant face recognition as a missing data problem. They assume there is a large data vector containing all the images of a subject under all the possible viewpoints. Their algorithm can achieve 75% correct in a database of 100 subjects with pose variation of up to $\pm 30^\circ$.

In contrast with the above statistical algorithms, which model the transformation of the entire facial region between frontal and non frontal images, Yamada et al. [72] proposed a patch based approach

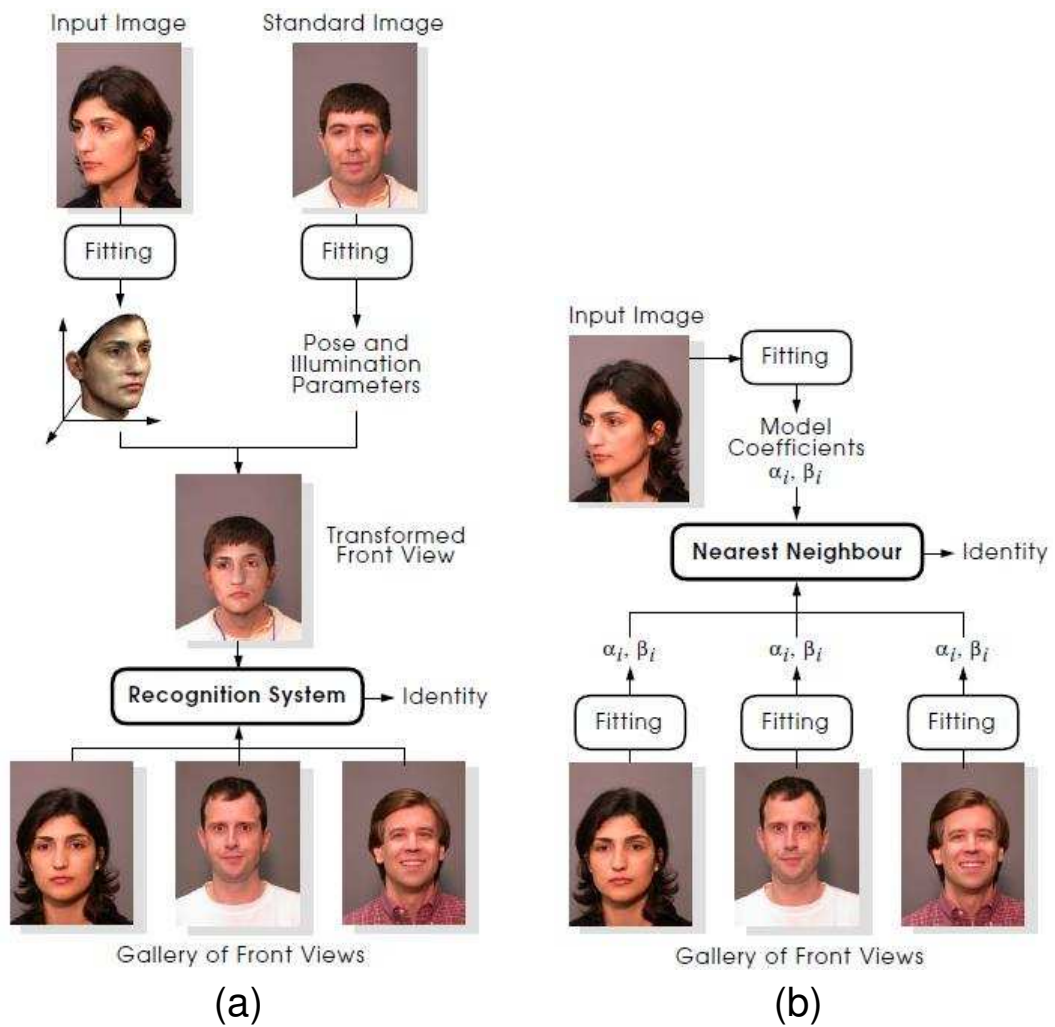


Figure 2.6: **Two recognition approaches of the 3D morphable model based algorithm [15].** a) A frontal view of an input probe image is firstly generated and then the generated frontal probe image is compared with each gallery face image. b) The model coefficients of probe and gallery images are firstly estimated and then face recognition is conducted by comparing the model coefficients directly. (Adapted from Blanz et al. [15])

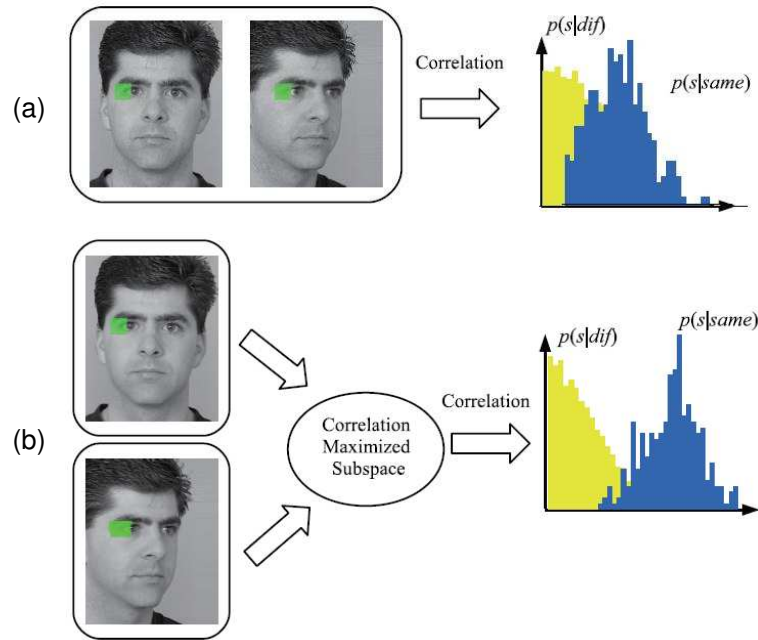


Figure 2.7: **Maximizing within-individual correlations improves pose invariance [81].** (a) Pose variation confuses the correlation distribution of two images. (b) Pose invariance can be achieved in the correlation maximized subspace. (Adapted from Li et al. [81])

to do face recognition across pose. They demonstrated that patches are more robust to pose variations than the holistic appearance. They applied a Gaussian probabilistic model and a Bayesian classifier to recognize faces. Lucey et al. [90] extended Yamada’s algorithm by modeling the statistical relationship between the frontal patches and holistic non-frontal image. Ashraf et al. [7] made the further improvement by applying a 2D affine transform to learn the patch correspondences. However, human faces have a complex 3D geometric structure and misalignment still exists. Thus Li et al. [81] applied a generic mean 3D face model to reduce the patch misalignment. Although their method obtains better patch correspondences of different poses, they found that the size of the corresponding patches might be different. The reason is because some surface points are visible and some points are not. To solve this problem, Li et al. used Canonical Correlation Analysis [61] to construct an intermediate subspace between the frontal and non-frontal subspaces. When the unequal-length vectors of different poses are projected into the intermediate subspace, the length of the projected vectors will become equal. They measure the similarity of patches from different poses by correlations in the intermediate subspace, in which the within-individual correlations are maximized and pose invariance can be improved as shown in Figure 2.7. They used two approaches to do recognition. In the first way, they transform non-frontal face images into frontal and then compare with the frontal gallery images. The second approach is to transform both the frontal gallery images and non-frontal probe images into the intermediate subspace and then compare them directly in the intermediate subspace.

Although the aforementioned 2D statistical methods are easy to implement and have low computation cost, their performance was worse than the 3D morphable model based algorithm until the tied

model was proposed [112]. The ‘Tied’ model means images from the same person but under two viewing conditions have a common hidden variable but different generation processes. The tied generative model produces better performance than the 3D morphable model. In Table 2.2 we compare the performance of the important pose invariant face recognition algorithms. The table demonstrates that a version of the tied generative model, Tied PLDA [82], produces the best performance.

Algorithm	Database	Pose Diff	% Correct
Light Fields [54]	FERET (100)	30	75
3D Morphable Model [15]	FRVT(87)	45	86
LLDA [75]	XM2VTS(125)	30	53
Tied Factor Analysis [113]	XM2VTS(100)	90	77
Tied PLDA [83]	XM2VTS(100)	90	87

Table 2.2: Performance comparison among pose invariant face recognition algorithms.

In Tied PLDA, face images are considered to be generated from the underlying identity variables which denote the identity of images. The generation process is different for different poses. More formally, the model can be described by the following equation:

$$\mathbf{x}_{ijk} = \boldsymbol{\mu}_k + \mathbf{F}_k \mathbf{h}_i + \mathbf{G}_k \mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ijk}, \quad (2.3)$$

where \mathbf{x}_{ijk} denotes the k^{th} pose of the j^{th} image of the i^{th} individual, $\boldsymbol{\mu}_k$ represents the mean image at pose k , \mathbf{F}_k is a matrix containing the between-individual basis functions in columns for pose k . The term \mathbf{h}_i represents the hidden identity variable which is constant for all the images from the i^{th} individual. The matrix \mathbf{G}_k is a matrix containing the within-individual basis functions in columns for pose k . The term \mathbf{w}_{ij} denotes the hidden noise variable which is different for each image. The term $\boldsymbol{\epsilon}_{ijk}$ represents a stochastic noise. We will introduce more details in section 5.3.2.

Although the 3D model based algorithms demonstrated a great potential to solve face recognition across pose, in practice most 3D model based algorithms are too slow to be applied in real time applications and noise within image reduces performance dramatically. It is probably preferable to use a 2D static method to solve face recognition across pose variation because it produces good performance, can be easily implemented and requires low computation cost.

Illumination Invariant Face Recognition

It has been argued in [2] that the variation among images of the same person due to illumination and viewing direction is almost always larger than image variation due to face identity. This observation has been confirmed by [106] [109]: the performance of face recognition methods of Stage III degrades significantly when illumination changes. As shown in Figure 2.8, there are four illumination components that affect the appearance of face images: diffuse reflection, specular reflection, attached shadow and cast shadow. The goal of research into lighting invariant face recognition is to handle the four components.

Shashua and Riklin-Raviv [122] proposed a quotient image based algorithm which models face appearance variation under the assumption of diffuse reflectance. However, the assumption of only diffuse

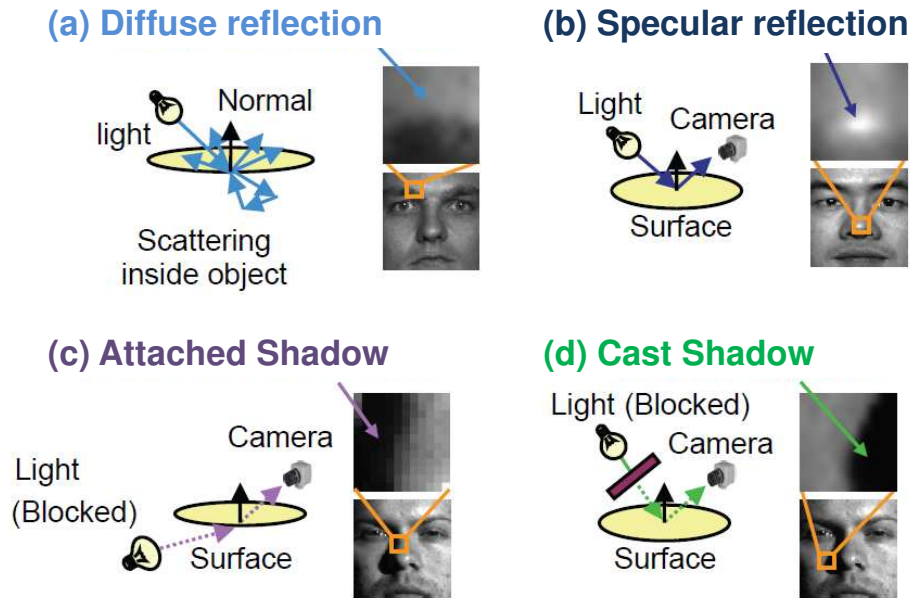


Figure 2.8: **Four illumination components to affect appearance of face images [100].** (a) Diffuse reflection occurs when incident light is scattered by an object. (b) Specular reflection occurs when incident light is reflected by an object. (c) Attached shadows occur when an object itself blocks the incident light. (d) Cast shadows occur when an other object blocks the incident light. (Adapted from Nishiyama et al. [100])

reflection existing is too strict for images in real life. Ramamoorthi [114] demonstrated that the appearance of a convex Lambertian object under distant illumination without cast and attached shadow can be completely described by a 3D linear subspace. Their algorithm only requires three images per person if images are taken under linear independent lighting. However, this requirement is still too difficult to be satisfied because normally only a single training image is available per individual in many face databases. To solve the problem, Wang et al. [125] presented the Self-Quotient image based algorithm which can use a training image to synthesize images under different lighting. However, their algorithm fails when cast shadows and attached shadows exist.

To handle cast shadows and attached shadows, Georghiades et al. proposed an algorithm based on illumination cones [52]. Their algorithm used seven images per person to synthesize the face image under different lighting. They demonstrated that their algorithm produced very good performance under different illumination conditions. However, similar to [122], it is unrealistic to have seven training images for each individual in a practical face recognition system. In Stage III illumination invariant face recognition is still an unsolved problem.

Overall, the performance of face recognition algorithms increased dramatically in Stage III. However, the algorithms in this stage are still sensitive to pose, lighting variation and long image capture intervals between probe and gallery images [105]. To obtain a wider application of face recognition, research was required to shift from controlled environments to uncontrolled environments.

2.2.4 Stage IV (2008 - present)

In the past 7 years, researchers focused on face recognition in uncontrolled environments. The Labeled Faces in the Wild (LFW) database [65] is the evaluation benchmark for face recognition under uncontrolled environments since its images are collected from the internet and it also provided a rigorous evaluation framework. Many new algorithms were proposed to improve the verification performance in the LFW database. We will describe the development of face recognition algorithms in this stage by introducing the progress in the four steps of a modern face recognition pipeline: face alignment, facial landmark detection, face representation methods and recognition algorithms. In Table 2.3 we list the important methods in each category.

Category	Main Algorithms	
Face Alignment	Congealing [69], Fiducial Points Based Similarity Alignment [134], Identity Preserving Alignment [13], 3D Alignment [135]	
Landmarks Detection	Component Based Detector [91], Shape Regression Based Detector [27], Local Model and Global Exemplar Combined Detector [11]	
Face Representation Methods	LBP [109], TPLBP FPLBP [147], Multi-Region Histogram [126], LE [28], LARK [128], LQP [71], Discriminant Face Descriptor [85], Large-Scale-Search-derived Feature [39], Spatial Face Region Descriptor [41], High Dimension LBP [33], Dense SIFT [131], Over-complete LBP [9]	
Verification Algorithms	Similarity Learning	Nowak Similarity Learning [107], Joint Bayesian Face [32]
	Reference Based Algorithm	Attribute and Simile Classifier [81], Multishot [134], Associate Predict Model [153], Tom-Vs-Pete Classifier [13]
	Metric Learning	LDML [62], CSML [105], DML-EIG [154], CMD [67], SUB-SML [26]
	Discriminative Subspace	PLDA [88]

Table 2.3: Main papers in Stage IV.

Face alignment

Alignment is critical to recognize uncontrolled images [13] [140] because alignment can reduce the image variation effectively. The authors of the LFW database provided the aligned images using the congealing alignment method [64]. However, misalignment still exists for some facial landmarks [127], for example, the eyes, mouth, nose. Wolf et al. [127] demonstrated that their fiducial points based alignment method can remove these misalignments. They first used a commercial fiducial points detector to locate seven fiducial points (the corners of the eyes, the mouth and the nose tip) and then applied a similarity transformation to register these fiducial points into a pre-defined template. They demonstrated that their alignment method helped improve the recognition performance [124]. This similarity transformation based alignment method was adopted by other literature [82] [30] [124]. However, the similarity

transformation fails for out-of-plane rotation caused by pose variation. To address the issue, Taigman et al. [128] proposed a 3D alignment method. They firstly detected 67 facial landmarks in the input face image and built up a correspondence of 67 detected landmarks between the input image and a 3D generic shape model. Then they obtained an affine 3D-to-2D camera by minimizing the loss between 2D points and their 3D references. Lastly they applied a piece-wise affine transformation to obtain the frontalized image of the input image. Figure 2.9 shows the pipeline of the 3D alignment method. They demonstrated that their method can handle out-of-plane rotations and this 3D frontalization alignment method obtained a 3% improvement from 94.3% to 97.33% in the LFW verification test when the rest of the pipeline (feature extraction and verification) was held constant.

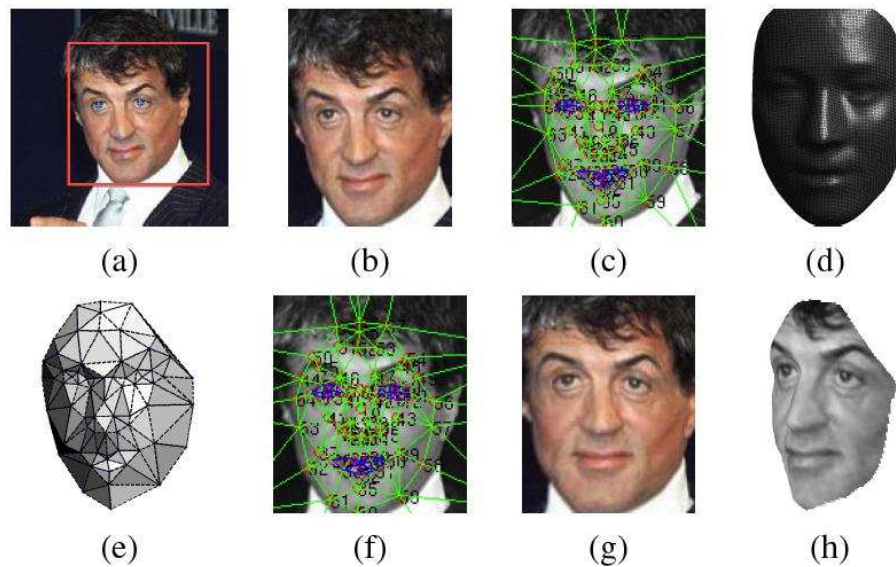


Figure 2.9: **3D alignment method pipeline [128]**. (a) A face is detected and 6 labeled landmarks are located. (b) The detected image is aligned and face region is cropped. (c) 67 landmarks are detected from the aligned image and the corresponding Delaunay triangulation is generated. (d) The reference 3D shape. (e) In the Delaunay triangulation, images are marked darker when they are less visible. (f) A piece-wise affine wrapping is conducted to generate frontal image based on 67 fiducial points. (g) The generated frontal image. (h) The generated non-frontal image. (Adapted from Taigman et al. [128])

Facial Landmarks Detection

Facial landmarks detection is a very important step in the face recognition pipeline. Face alignment and facial feature extraction depend on accurate facial landmarks localization. An early study [21] described facial landmarks detection as a component of face detection. For example Ding et al. [44] provided bounding boxes around facial components when detecting faces from images. Recently many landmark detectors are trained to respond to a specific landmark [24] (e.g. the eye corners or nose tip). These landmark detectors search over a small image region and return a score at each location. One or multiple locations with the highest score are selected to be candidates for the specific landmark. However, false detection results are often obtained. A common mistake [148] [24] [135] is that the left corner of the left eye detector often locates the left corner of the right eye.

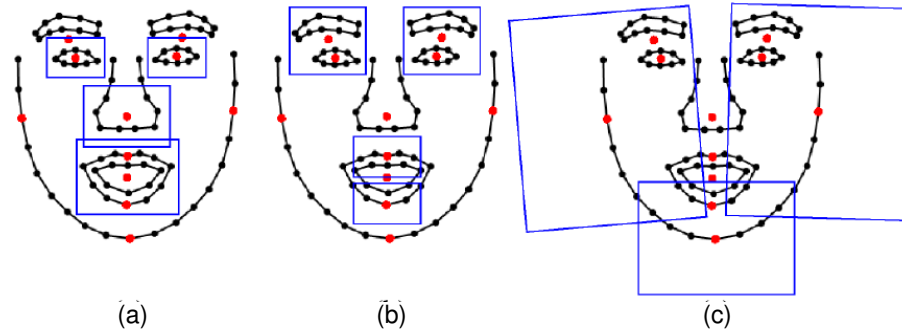


Figure 2.10: **Facial components and the corresponding patches used for training of [85].** Facial components are located by red points. Blue rectangles denote image patches used to train component detector. (a) Eyes, nose and mouth. (b) Brows, upper and lower lips. (c) Left, right, lower profiles. (Adapted from Liang et al. [85])

Eckhardt et al. [45] solved the problem by detecting a larger area, for example increasing the detection region from one eye to the whole area of two eyes. However, searching a larger area increased the chances of false detections. Therefore, researchers established constraints about the relative locations of landmarks to each other rather than the locations of landmarks to the detected face bounding box [120]. The predicted location of a facial landmark can be expressed as a conditional probability distribution given the other landmark positions. In this way, local landmark detectors are often combined with prior global landmark configurations [57] [96].

The locations of some facial landmarks vary significantly with expression. Examples include the eyebrows and lip. The solutions of [57] and [96] are to detect stable points, for example eye corners. However, these stable points might be difficult to detect when partial occlusion exists. To address this issue, Belhumeur et al. [11] proposed a RANSAC-like method to sample different types of landmark exemplars. Their method can locate facial landmarks accurately even for uncontrolled images from the internet.

Instead of searching for a single landmark, Liang et al. [85] proposed a component-based method to search face landmarks in a large range at the facial component level. Figure 2.10 shows facial components (e.g., eyes, nose, mouth and profiles) they defined. They showed their approach can discover the configuration of facial components effectively and rapidly in a large searching range. A very good fitted face shape can be refined within a few iterations. Chen et al. demonstrated this alignment approach helped improve their recognition performance in LFW verification test [30].

Due to large pose change, expression variation and partial occlusion, current facial landmark detectors still fail for some uncontrolled images. A more robust and efficient method is still required.

Face representation methods

It has been shown in many studies [140] [121] that extracting facial features to represent a face image instead of using raw pixels improves performance significantly. Local binary patterns (LBP) [102], one of the most successful features, are found to be very effective for the face verification task in the LFW database [5] [140] [82]. Variants, such as Three-Patch LBP (TPLBP) [140], Four-Patch LBP (FPLBP)

[140], and Local Quantized Patterns (LQP) [66], have been proposed to improve the discriminative performance. Other local image descriptors, such as the Scale Invariant Feature Transform (SIFT) [88] and the Histogram of Oriented Gradients (HOG) [38], have also been applied to verify face images in [58]. The aforementioned descriptors describe local geometric structures of face images by quantising gray level patterns or the image gradients. Seo et al. [121] proposed the Locally Adaptive Regression Kernel (LARK) feature without using any quantization. They use the geodesic distance between a center pixel and surrounding neighbor pixels to encode the local image structure. They demonstrated that their method can capture the local image structure more robustly and their feature has more discriminative ability. Their experiments demonstrate that the LARK feature outperforms LBP feature.

Unlike for the above features [102] [88] [121], Cao et al. [26] learn the local image structure encoder from training images. Therefore, they called their feature a learning-based (LE) descriptor. They indicated the above handcrafted features (LBP, SIFT and HOG features) can be viewed as a quantized code of the image gradient. The handcrafted features have two limitations: an optimal encoding method is difficult to define manually and the code distribution of real images is uneven. Some codes rarely appear for real life images. This uneven distribution means the final code histograms are less informative and will decrease the discriminant power. They demonstrated that these issues can be addressed if an unsupervised method is applied to learn the encoding method. They demonstrated that their learned encoder could achieve a good balance between invariance and verification power automatically. Their experiments demonstrated that their LE feature produced better performance than the LBP and HOG features.

Generally the dimensions of the above features [140] [82] [30] vary from 1K to 5K. Cao et al. [31] found high dimensional features can improve performance significantly. They built an image pyramid with different resize scales for each image and extracted LBP features from 27 fiducial points of each sample scale. Using this approach, the dimensions of LBP features from an image can be 100K. Their experiment results demonstrate 5% improvement by using high dimensional features instead of the traditional way to extract features. This conclusion is confirmed by other studies [124] [9]. For example, Simonyan et al. [124] extracted high dimensional dense SIFT features from face image and also achieved a significant performance improvement. By applying a similar principle, Barkan et al. [9] proposed high dimensional OCLBP features and confirmed that high dimensionality helps achieve high performance.

Instead of using high dimensional features, Taigman et al. [48] demonstrated that an extremely compact face representation can also produce very good performance. They applied a nine-layer deep neural network to derive a compact face representation method. The advantage of this new deep neural method is that it can be trained by using millions of face images efficiently and learnt nearly all the possible variations from the huge training data. Their method produced good performance in the LFW database [63].

Many researchers [140] [82] [26] [30] found performance can be improved by combining multiple features. For example, Wolf et al. [140] obtained 3.1% improvement by combining four features (LBP, TPLBP, FPLBP, SIFT) instead of only using the LBP feature.

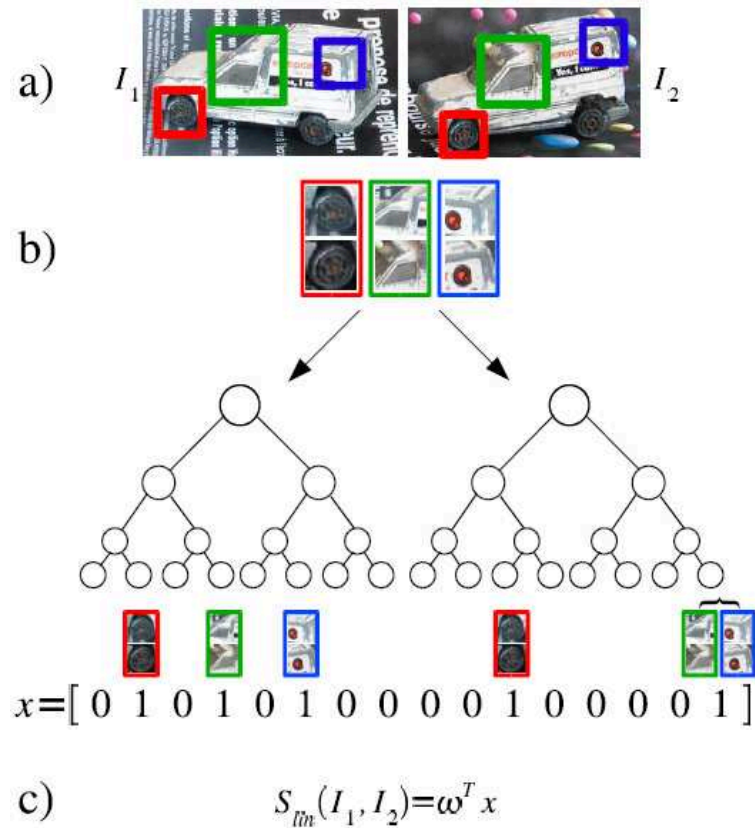


Figure 2.11: **Calculation of similarity score in [101]**. (a) Patch pairs are collected from two images. (b) Some randomized trees are applied to these patch pairs to obtain a binary vector x . (c) A SVM classifier is used to calculate the similarity score by aggregating the output of all decision trees. (Adapted from Nowak and Jurie [101])

Verification algorithms

The verification algorithms in Stage IV can be divided into four categories: similarity learning, reference based algorithms, metric learning and discriminative subspace algorithms. We will introduce each in turn. We describe the performance of these algorithms in the LFW database. The LFW verification scheme defines two testing protocols: unrestricted and restricted protocol. In the unrestricted protocol, identity labels associated with images can be used to generate more training pairs. In the restricted protocol, identity labels cannot be used.

Similarity learning algorithms estimate the visual similarities between two images and then determine whether two images are from the same person. Nowak and Jurie [101] proposed the first verification result in the LFW database in 2007. They used Randomized Decision Trees [53] and Support Vector Machines (SVM) [36] to estimate the similarity of two face images. For two face images they firstly chose a patch of random size at a random position in the first image and then searched for the

most similar patch at a nearby location in the second image. Using this approach many patch pairs can be generated from the two images. Then several randomized binary decision trees are trained to label each patch pair. If the patch pair reaches the leaf of a tree, the label of leaf is set to 1; if a leaf is never reached, it is set to 0. So each patch pair can be represented by a binary vector and an image pair can be represented by a concatenating vector from all patches. The similarity score of two images is calculated by applying a SVM classifier. Figure 2.11 illustrates how to compute the similarity score between two images. When the similarity score is bigger than a threshold, two face images are considered to be from the same person. Their algorithm achieves 84.2% correct under the restricted test protocol in the LFW database. However, their algorithm performs slowly because it needs to search all the similar patches among two face images. Later Chen et al. [30] proposed the Joint Bayesian Face algorithm. The Joint Bayesian Face algorithm divided each face image into two components: identity and within-individual variation. In training, an EM-like algorithm is used to estimate two components of each image and learn the between-individual covariance matrix and the within-individual covariance matrix. In test, the match and non-match covariance matrix are derived based on the between-individual covariance matrix and the within-individual covariance matrix, then a log likelihood ratio between two images is computed to decide identity assignment. Their algorithm achieved 90.9 % correct by combining four types of facial features and 93.18 % correct by using high dimensional local binary patterns (LBP) feature in the LFW database [31].

Reference based algorithms represent a face image by comparing to a set of reference images. In 2009 Kumar et al. [76] proposed the first reference based algorithm. They use the output of attribute and simile classifiers to represent an image. Attribute classifiers are to use binary classifiers to estimate the presence of 65 describable aspects of visual appearance, such as gender, race, age, hair color, etc. These visual traits were called attributes. Each face is represented by a vector in which each element represents the presence of attribute. Simile classifiers is to compare the whole faces or facial component with a pre-defined image set. For example a face can be described as having eyes similar to George Bush and a mouth similar to David Cameron. These traits are called ‘similes’. A face is represented by a vector, in which each element represents whether a visual feature of the input face is similar to one of the reference people. Figure 2.12 shows the attribute classifier and simile classifier. Their experiments show attribute classifiers can achieve 83.62% correct and simile classifiers can achieve 84.14% correct in the LFW database. Their algorithm can achieve 85.29% correct by combining the two classifiers. However, their algorithm has a drawback that it requires a large amount of manual labeling.

Wolf et al. [75] proposed another reference based algorithm in the same year. Their algorithm is called One-Shot Similarity (OSS) measure. They assume there are two face image vectors \mathbf{x}_i , \mathbf{x}_j and a face image set \mathbf{A} which has different identities from images \mathbf{x}_i and \mathbf{x}_j . Firstly image set \mathbf{A} is used as negative examples and \mathbf{x}_i as a positive example to train a model and then the learned model is used to classify image \mathbf{x}_j to get a classification score η_1 . This score represents the likelihood of image \mathbf{x}_i having the same identity as \mathbf{x}_j . Then switch the role of \mathbf{x}_i and \mathbf{x}_j to obtain another score η_2 . The final similarity score for two images is given by the average of η_1 and η_2 . Their algorithm produced 82.5%

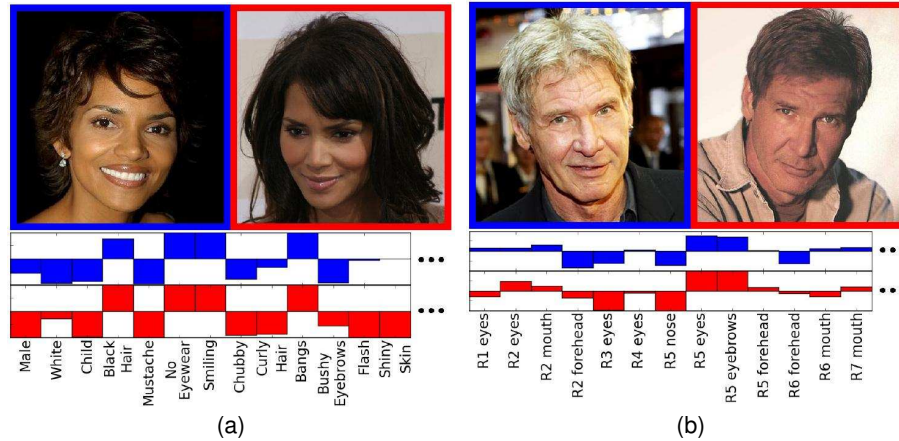


Figure 2.12: **Attribute and simile classifiers in [76].** (a) Attribute classifiers are trained to judge describable aspects of visual appearance are present or absent. (b) Simile classifiers are trained to judge whether some parts of faces are similar to the predefined reference people. (Adapted from Kumar et al. [76])

correct under the unrestricted testing protocol in the LFW database. Later they extended their algorithm to propose Multiple One-Shots to handle pose variation and improve the performance to achieve 89.5% correct [127].

Yin et al. [146] demonstrated that large within-individual variations are the bottleneck for improving performance in the LFW database. They proposed an associate-predict model to address this issue. Their model is built on a reference identity data set in which each of 200 identities have 28 images with seven pose categories and four lighting conditions. To compare two face images x_i and x_j , they firstly estimate the pose category and lighting condition of each image. If the input image pair has very similar pose and lighting condition, they apply a direct appearance matching method [10] to compute a similarity score; otherwise, they apply associate-predict model to handle large within-individual variation. They demonstrated that their associate-predict model produced better performance by using facial components than the holistic face, so they divided each input image into 12 facial components as shown in Figure 2.13 (a). The associate-predict model contains two models: appearance-prediction model and likelihood-prediction model. In the appearance-prediction model they selected a reference identity from the reference identity data set for each of 12 facial components of image x_i . The selected reference identity has the most alike component as the component of image x_i . A different component may associate a different alike identity. Then they chose the image with the same pose and lighting condition as image x_j from 28 images of the selected reference identity and picked the corresponding facial component. By this approach they selected 12 reference components and formed a new face image x'_i , which has the same pose and lighting condition as image x_j . The new face image x'_i is shown in Figure 2.13 (b). Lastly the 12 distances of the corresponding component pairs between image x'_i and x_j were calculated and a linear SVM [29] was applied to fuse these distances to obtain a final similarity score. In the likelihood-prediction model they selected 3 most alike reference identities for each component of image

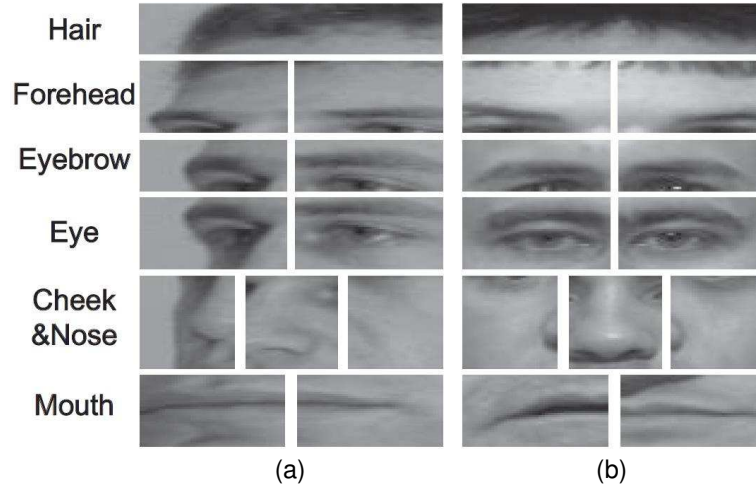


Figure 2.13: **The frontalization effect of the appearance-prediction model in [146].** (a) Each input image is divided into 12 facial components. (b) A frontal image is formed by selecting facial components from reference data set. (Adapted from Yin et al. [146])

\mathbf{x}_i . After that they constructed a component discriminative classifier [10] by using all the components of the 3 selected reference identities and the input components of \mathbf{x}_i . Then each of the corresponding components of \mathbf{x}_j was fed to this classifier to compute a component distance. Lastly a linear SVM [29] was applied to fuse 12 component distances to compute the final similarity score. Their algorithm achieved 90.57% correct under the unrestricted protocol in the LFW database when they fused 24 distances of appearance-prediction model and likelihood-prediction model by a linear SVM.

Metric learning algorithms aim to find a metric to separate two classes. The main goal is to learn a Mahalanobis distance $(\mathbf{x}_1 - \mathbf{x}_2)^T \Psi (\mathbf{x}_1 - \mathbf{x}_2)$, where Ψ is a positive definite matrix. In 2009 Guillaumin et al. [58] proposed two approaches to learn robust distance measures for two images: a) the logistic discriminant base metric learning method (LDML) used a logistic discriminant to learn a metric from a set of labeled image pairs; b) Marginalized k-nearest-neighbour (MkNN) method computed the number of positive neighbor pairs (having the same class) out of the possible pairs within the neighborhoods of images \mathbf{x}_i and \mathbf{x}_j to obtain a similarity score. Figure 2.14 describes LDML and MKNN. Their experiments demonstrated that applying LDML could achieve 79.27% correct under the restricted testing protocol and combining LDML and MkNN can achieve 87.5% under the unrestricted test protocol in the LFW database. Later Nguyen and Bai [99] proposed a cosine similarity metric to replace the Euclidean distance in the learning problem. Their experiment showed that they could achieve 88% correct under the unrestricted test protocol in the LFW database.

Discriminative subspace algorithms model the image difference by projecting the two images into a low-dimensional subspace. The Eigenfaces algorithm [132] is the earliest subspace algorithm and became the benchmark algorithm in the LFW evaluation. Probabilistic LDA (PLDA) [111] divides the image into three components: identity, within-individual variation and unexplained noise. In training, the basis functions for between-individual and within-individual subspace are estimated. In test, the

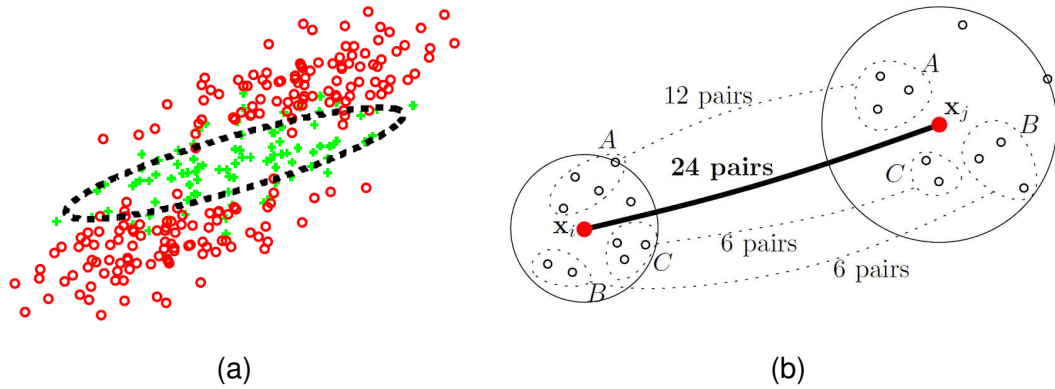


Figure 2.14: **Visualization of the logistic discriminant base metric learning (LDML) algorithm and the marginalized k-nearest-neighbour (MKNN) method [58].** (a) LDML aims to find an ellipsoid to separate classes. (b) MKNN aims to find the number of the corresponding positive pairs within the neighborhoods of image x_i and x_j , where a positive pair means two members belong to the same class. Here each image x_i and x_j has 10 neighborhoods and there are three classes A, B and C within the neighborhoods of the image pair. There are 24 positive pairs out of 100 possible pairs and thus the similarity score is 0.24. (Adapted from Guillaumin et al. [58])

match and non-match covariance matrix is obtained to give the match and non-match likelihood for two input images. PLDA achieved 90.03% under the restricted testing protocol in the LFW database [82].

Overall, in recent years, significant performance improvement has been achieved for face recognition under uncontrolled environments. This improvement comes from larger training database, better alignment, more accurate landmarks detector, more sophisticated face features and better verification algorithms. More training images and more accurate landmark detectors are probably the most significant to cause the improvement. Outside academic research, commercial applications to face recognition have extended from the traditional security domain to social networks, electronic entertainment, online face search. Examples include automatically tagging identity in Facebook.com and searching the best potential lover based on faces in Jiayuan.com.

2.3 Face Databases and Performance Evaluation

Performance evaluation schemes play an important role in the development of face recognition as they determine the most promising algorithms and indicate future research directions. Face databases play an important role in algorithm development, model training and performance evaluation in face recognition research. In this section we will introduce the main face databases and performance evaluation methods.

2.3.1 Face Databases

In a performance evaluation scheme, it is important to choose a proper face database. There are a number of face databases available to researchers. We list the main databases at each development stage in Table 2.4 and give a brief introduction to each database:

Stage	Database	Year	Identities	Images	Poses	Illumination	Expressions	Sessions
II	FERET [106]	1993	1,199	14,051	9-20	2	2	2
	ORL [119]	1994	40	400	3	3	2	2
	Yale [10]	1997	16	160	1	3	6	1
	AR [92]	1998	116	3,288	1	4	4	2
III	XM2VTS [95]	1997	295	1,526	1	1	1	4
	PIE [123]	2000	68	41,368	13	43	3	1
	Yale B [52]	2001	10	5,760	9	64	1	1
	KFDB [18]	2002	1,000	52,000	7	16	5	1
	CAS-PEAL [51]	2003	1,040	99,954	9	15	5	2
	FRGC [107]	2004	4,003	50,000		2	2	
	Multi-PIE [56]	2008	337	750,000	15	19	6	4
IV	LFW [65]	2007	5,749	13,233				
	Pubfig [76]	2009	200	58,797				
	WDRRef [30]	2012	2,995	99,773				

Table 2.4: **Main face databases at each development stage.** For each database we list its key features, which include (where available) collection date, the number of subjects, images, poses, lighting conditions, expressions and recording sessions. Blank entries indicate that image capture was not controlled.

The Facial Recognition Technology (FERET) Database [106] was sponsored by the Department of Defense of the American government and was collected by George Mason University. The famous three FERET tests [106] [116] [110] and facial recognition vendor test (FRVT) 2002 [109] used this database. Recognition performance from many academic and commercial algorithms [139] [118] [17] are available and the direct comparison with other algorithms is possible. All the images are gray and the image size is 256×384 pixels.

The Olivetti Research Ltd (ORL) Database [119] was collected by Cambridge University between 1992 and 1994. Each subject has ten images with varying pose (left or right head movement), facial expression (open/close eyes, smiling/neutral), illumination, and facial attributes (glasses/without glasses). All the images are grey and with the size 92×110 pixels. This database was often used in the 1990s [78] [59] [4], but now it is regarded as too easy since the variation is relatively limited.

The Yale Face Database [10] was collected by Yale University. It contains 160 frontal images from 16 people under 10 conditions: an image under ambient lighting, one with or without glasses, three images under different light sources, and five images with different expressions. All the images are grey and the image size is 320×243 pixels. The motivation of Belhumeur et al. to collect this database is to test his well-known Fisherfaces algorithm [10]. However, it has been regarded as an easy database now.

The AR Face Database [92] was collected by the Universitat Autnoma de Barcelona in 1998. It includes 3,288 images from 63 men and 53 women. All the images are color and of the size 768×576

pixels. Images were collected in two sessions. The database has been accessed by more than 200 research groups [145] [142].

The Extended M2VTS Database (XM2VTSDB) [95] was collected by the University of Surrey. It is designed for the development of multi modal verification. The database contains 295 subjects, each of which was recorded at four sessions over a period of four months. All the images are color and with the size 92×110 pixels. They also provide 3D head models for 293 subjects. It was a popular database [69] [35] [111] at Stage III.

The CMU Pose, Illumination, and Expression (PIE) Database [123] was collected by the Carnegie Mellon University. The database designers sampled images by varying the pose, illumination and expression. This database has an important influence in face recognition across pose [117] [55] [22]. All the images were color and with the size 640×480 pixels.

The Yale Face Database B [52] was collected by Yale University. It is an extended version of the Yale Face database [10]. Its purpose was to verify the performance of the database designers' new algorithm under large variation of pose and illumination. All the images are grey images with the size 640×480 pixels. It has been used by many researchers [142] [141] [25].

The Korean Face (KFDB) Database [18] was collected only from Korean people. The collection scheme is quite similar as the FERET database. The database was designed to evaluate face recognition performance for Asian people.

The CAS-PEAL Face Database [51] was collected by the Chinese Academy of Science. All the subjects are Chinese. All the images are grey and with the size 360×480 pixels. It has been used by many Chinese researchers [84] [149] [130].

The Face Recognition Grand Challenge (FRGC) Data Set [107] is the publicly accessible face data set of Face Recognition Vendor Test (FRVT) 2006, which is sponsored by the Federal Bureau of Investigation (FBI), the Department of Homeland Security of United States, the National Institute of Standards and Technology of United States, etc. The FRGC data set was collected for the Face Recognition Grand Challenge project, whose goal is to advance face recognition technology for the U.S. Government. It is a very important database for face recognition research [19] [108].

The Multi-PIE Face Database [56] is the extended version of the PIE database. It is designed to address the shortcomings of the PIE database: a limited number of identities, few expressions and a single recording session. The Multi-PIE database consists of 750,000 images from 337 identities under 15 view points and 19 lighting conditions. Many researchers [26] [153] [136] used it for face recognition across pose and illumination.

The Labeled Faces in the Wild (LFW) Database [65] includes 13233 images, which were collected from the internet by the University of Massachusetts Amherst. The database was designed to study unconstrained face recognition. It is the most important face database at Stage IV. Most important papers [127] [76] [82] [48] regarding face recognition under uncontrolled environments used this database.

The Public Figures (Pubfig) Database [76] contains 58,797 images from 200 people, which were collected from the internet by using the same method as the LFW database. There are fewer identities

but much more images per person than the LFW database. Its disadvantage is that the database designers only provide image URLs instead of images because of copyright issues. As of 2014, 15% of the image URLs have been expired, so using this database to have a fair comparison among algorithms has become impossible.

The Wide and Deep Reference (WDRRef) Database [30] consists of 99,773 images from 2995 identities. It is more wide (more images in total) and deep (more images per person) than the LFW database. However, the disadvantage of the database is that it only provides the extracted LBP [5] and LE [26] image descriptors for each image rather than the images themselves because of copyright issues. Therefore, it restricts other researchers from using this database.

The aforementioned databases can be divided into three categories based on image acquisition method. The first type of database is built by a small group of researchers in the laboratory. Examples include the Yale database [52], the AT&T face database [12]. Images are obtained in a short time and the database size is small. The variabilities of images in these databases are well controlled. The second type of database is still collected in the laboratory but with much greater variation. Examples include the XM2VTS database [95] and the Multi-PIE database [56]. The designers of these databases attempt to capture the face distribution of various parameters to make the most useful database. However, there is a drawback for this type of database that it is difficult to capture the correct statistics. For example, it is not clear how researchers should decide the ratio of face images wearing glasses, the percentage of images with smile expression, the proportion of images with office or conference background. The third type of database collects the existing images from the internet rather than capturing images in the laboratory. Examples include the LFW database [65], the Pubfig database [76], and the WDRRef database [30]. Although this type of database has its own biases, for example there are limited non-frontal images because of using Viola-Jones [134] frontal face detector. However, the third type of database contains images with very large of diversity: it is more suitable for studying face recognition in uncontrolled environments when comparing with the previous two types of databases.

Based on the above analysis, the LFW database has many advantages and is most appropriate to be used to evaluate face recognition in uncontrolled environments.

2.3.2 Performance Evaluation

Performance evaluation for face recognition algorithms provides a framework to measure recognition performance, determine the most promising algorithms and indicate future development directions. We will firstly introduce the precepts and methodology of performance evaluation [47] proposed by Phillips et al., then we will describe the most popular evaluation methods at each development stage of face recognition. Lastly I will summarise the results of FERET tests, Face Recognition Vendor Tests (FRVT) and LFW tests.

Evaluation Precepts

Phillips et al. [47] proposed evaluation precepts and applied them to design three FERET tests at Stage II and three FRVT tests at Stage III. The details of the precepts are as following:

1. Evaluation should be designed and administered by groups that are independent of the

algorithm developers and vendors.

2. Test data should be sequestered and not seen by the participants prior to the evaluation.
3. The design, protocol and methodology of the evaluation should be published
4. Evaluation results should be spread in a manner that shows meaningful differences among the participants.

Evaluation Methodology

In a typical evaluation there are three sets of images. The first set is called the gallery image set \mathbf{G} , in which we have already known the identity of each image. The other two sets are both probe set. The first probe set, in which identities of images can be found in the gallery set \mathbf{G} , is denoted as \mathbf{P}_g . The second probe set, in which identities of images cannot be found in the gallery set, is denoted as \mathbf{P}_n . The similarity between a probe image and a gallery image is measured by the similarity score \mathcal{S} . If the similarity score is higher than an pre-defined threshold τ , the probe image and gallery image are considered as matching. If we want to obtain the n most similar gallery images for a probe image, we refer to this as rank n match. The rank 1 match is called first match or top match.

There are three fundamental face recognition tasks: open-set identification, closed-set identification, face verification. Each task has its relevant performance measure methods.

The goal of **Open-Set Identification** is to find which gallery image matches the probe face image. However, it is also possible that a probe image might not match any gallery image. There are two performance statistics: the identification rate P_{IR} and the false alarm rate P_{FA} . The identification rate is the fraction of probe images in \mathbf{P}_g identified correctly. The false alarm rate is the fraction of probe images in \mathbf{P}_n identified wrongly. The identification rate P_{IR} and the false alarm rate P_{FA} for top match can be calculate by

$$P_{IR} = \frac{|p_i : \eta_{i*} \geq \tau|}{|\mathbf{P}_g|} \quad (2.4)$$

$$P_{FA} = \frac{|p_j : \eta_{j*} \geq \tau|}{|\mathbf{P}_n|}, \quad (2.5)$$

where the term p_i denotes a probe image which belongs to \mathbf{P}_g ; the term η_{i*} denotes the similarity score between the probe image p_i and its most matched gallery image g^* ; the term p_j denotes a probe image which belongs to \mathbf{P}_n ; the term η_{j*} denotes the similarity score between the probe image p_j and its most matched gallery image g^* .

The ideal system should have a identification rate of 1.0 and a false alarm rate of 0.0. In real world systems performance varies when the threshold τ changes. The identification rate and false alarm cannot increase simultaneously. The algorithm designers have to make a trade-off between the identification rate and the false alarm rate. The receiver operator characteristic (ROC) is used to measure the trade-off. In a ROC plot the horizontal axis depicts the false alarm rate which is normally scaled logarithmically, and the vertical axis depicts the identification rate. Figure 2.15 shows an example of an ROC.

Closed-Set Identification is a special case of open-set identification. Here, the probe image is known to definitely match a gallery image in close-set identification. Consequently, the performance is

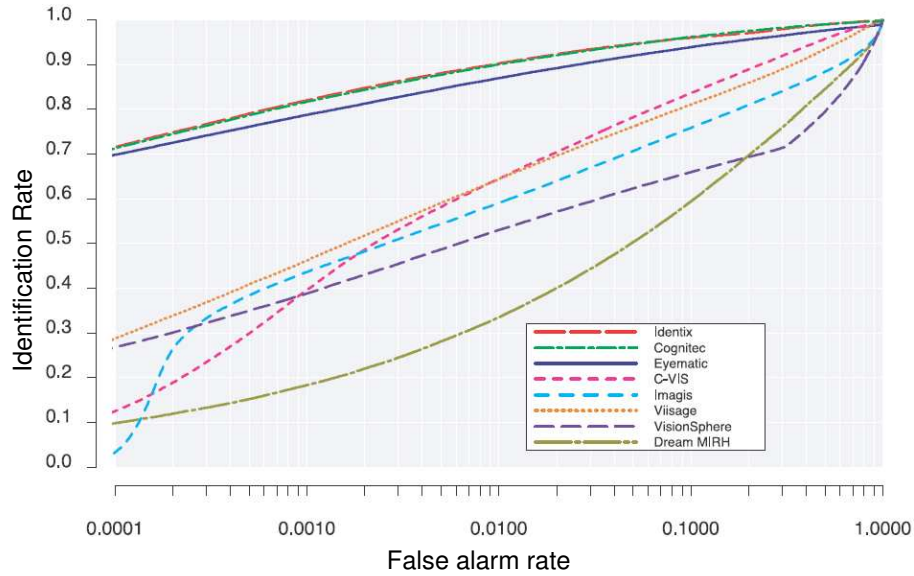


Figure 2.15: **Open-set identification performance reported on an ROC figure.** This graph demonstrates the trade-off between the identification rate and false alarm rate. The horizontal axis depicts false alarm rate on a logarithmic scale. The vertical axis depicts identification rate. (Adapted from Phillips et al. [109])

only measured by the identification rate. The cumulative match characteristic (CMC) figure is generally used to describe the performance. In a CMC figure the horizontal axis depicts the rank and the vertical axis depicts identification rate. When only rank 1 is considered, it is called the first match and is used most frequently. Occasionally people might have interest to know the performance when rank $n = 5, 10$. An example of a CMC is shown in Figure 2.16.

The goal of **Verification** is to verify whether two images match or not. There are two standard protocols to evaluate verification. The first protocol is called the round-robin method. The probe sets \mathbf{P}_g and \mathbf{P}_n are combined together. The verification rate and false alarm rate are computed by matching all the probe images to all the gallery images. The disadvantage of the first protocol is that it cannot model the case where false identities are caused by people not in the gallery. The second protocol, called the true imposter protocol, overcomes this drawback. In the second protocol, the identification rate is calculated by using gallery set G and probe set \mathbf{P}_g , the false alarm rate is computed by using the gallery set G and the probe set \mathbf{P}_n . Since the identities in \mathbf{P}_n are not in the gallery, the nonmatching scores between the gallery and \mathbf{P}_n are called true imposters.

At each development stage the main evaluation methods are different. Table 2.5 lists the main evaluation methods in each stage.

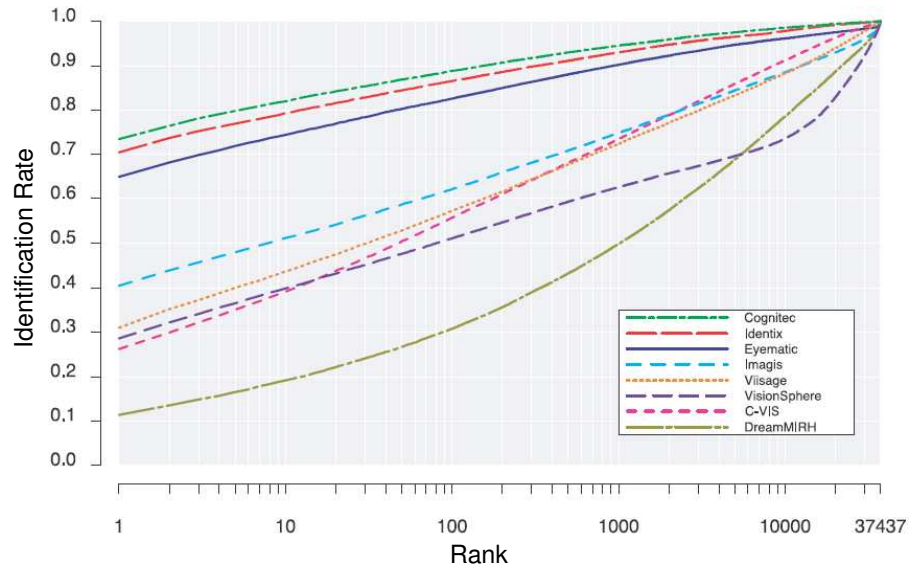


Figure 2.16: A CMC figure plots identification rate as a function of rank n . The vertical axis depicts identification rate, and the horizontal axis depicts rank on a logarithmic scale. (Adapted from Phillips et al. [109])

Stage	Main Evaluation Methodology
II	Closed-Set Recognition
III	Closed-Set Recognition
IV	Verification

Table 2.5: Main evaluation methodology for each stage. Closed-Set recognition refers to the identification that the identities of the input images are in the gallery. Verification refers to verifying whether two images match or not.

Landmark Tests

Because face recognition methods at Stage I were far from practical application, there is no well-known evaluation. In Stage II, the three FERET tests [106] [116] [110] were organized to evaluate academic algorithms. In Stage III, the FRVT Tests were applied to evaluate commercial face recognition systems. In Stage IV, the comparison among algorithms have been applied mainly in the LFW database.

The three FERET tests were carried out in 1994, 1995 and 1996. The three tests applied the aforementioned evaluation and methodology. The detail of the FERET evaluations can be found in [106] [116] [110]. The test results of the FERET evaluation in 1996 showed the elastic bunch graph matching method (EBGM) algorithm [139], the Bayesian algorithm [97] and the Fisherfaces algorithm [10] produced the best performance. The FERET tests recorded the advance of face recognition technology but also revealed three major challenges to face recognition algorithms: pose changes, illumination variation

and large image capture interval.

The three FRVT evaluations were the successor of the FERET evaluations. Since 1997 there was a quick development of commercial face recognition systems. This development not only includes the face recognition technology but also the relevant supporting system and infrastructure. By 2000 many commercial face recognition systems were available. To assess the state of the art of face recognition systems, the FRVT evaluations were organized. Therefore, the main difference between the FERET tests and FRVT evaluations is that the FRVT participants were commercial systems while the participants of the FERET evaluation were laboratory systems. Another difference is that image variation in the FRVT test was larger than in the FERET test. The three FRVT tests were applied in 2000, 2002, 2006 respectively [106] [109] [105]. The test report of the FRVT 2006 concluded [105]: 1. Compared with the results of the FRVT 2002, the performance improved by a order of magnitude. The best system can achieve a False Reject Rate (FRR) of 0.02 at a False Accept Rate (FAR) of 0.001 under controlled illumination. 2. The first 3D face recognition benchmark was built. 3. The performance of face recognition algorithms is better than humans when lighting varies.

The verification test in the LFW database focusses on the problem of whether two images match. The LFW designers established an evaluation protocol to compare the performance of different algorithms [65]. In the LFW test, 6000 images are divided into 10 subsets which are mutually exclusive in terms of identities and images. The experiments are required to be performed 10 times by applying a leave-one-out validation scheme. In each experiment, one subset is selected for testing and the remainder of the 9 subsets are used for training. The final performance is reported using a receiver operating characteristic (ROC) curve or the mean of 10 experiment results and the standard error of the mean. Two separate paradigms are provided to use the training data: the restricted and unrestricted schemes. In the restricted scheme identity labels associated with images are not allowed to be used so only provided pairs can be used in training. In the unrestricted scheme a large number of training image pairs can be generated because identity labels are allowed to be used. The current best results in the LFW database is 97.35% correct by a commercial system [128] using the unrestricted protocol and training images from outside the database. This performance is close to human performance 97.5% correct [76].

2.4 Conclusion

In this chapter we reviewed the development of face recognition technology. After decades of development, significant progress has been made. The research has shifted from controlled environments to uncontrolled environments. The current face recognition systems can produce good performance with uncontrolled images. However, it does not mean face recognition in uncontrolled environments is a solved problem. In fact there are many challenges still existing. Therefore, we are motivated to propose new algorithms in this report to overcome the challenges.

We have also described and assessed the main face databases and find that the LFW dataset is the most appropriate database for comparing the performance of face recognition algorithms in uncontrolled environments. We will use the LFW database to evaluate our algorithms in the following chapters.

We also reviewed performance evaluation methods of face recognition and find that face verification

has become the mainstream evaluation target for face recognition in uncontrolled environments. We will focus on improving the performance of face verification in this report.

Chapter 3

Investigating the Spatial Support of Signal and Noise in Face Recognition

3.1 Introduction

Automated face recognition has many real world applications. Unfortunately, many systems perform well only in controlled conditions where the pose, illumination and expression of the probe face are the same as the gallery face. Recognition in uncontrolled conditions is the subject of much research and can be divided into two categories. *Physically-based* algorithms have a forward model with knowledge of 3d geometry and light transfer. They attempt to explicitly fit the pose and lighting parameters (e.g. [17]). *Statistical algorithms* eschew this knowledge in favour of attempting to directly model the images themselves as abstract feature vectors (e.g. [132] [10] [137]). In this chapter we will restrict our discussion to statistical algorithms, especially statistical subspace algorithms.

Preprocessing for statistical face recognition can be divided into global and local algorithms. In a typical global approach, the pixel values from the whole image are vectorized. A linear or non-linear transformation is applied to this vector to move it to a space in which signal:noise is improved before making the decision. Examples of global algorithms include the Eigenfaces algorithm [132], the Fisherfaces algorithm [10] and the Laplacianfaces algorithm [62]. Implicit in this algorithm is that it is sensible to model the joint covariance of all image pixels. This particularly makes sense in the presence of illumination and pose changes which affect the whole face.

In local methods, facial keypoints (eyes, nose etc.) are found. A separate data vector is extracted from each keypoint. These data vectors are modelled separately and treated as independent contributions to the final recognition decision. Examples of this approach include [67] and [139]. The logic of the local approach is that each part of the face contains information about identity that is independent from that in other regions. A disadvantage is that it is harder to account for global factors such as lighting changes if we only look at a small part of the image at a time.

Many algorithms are suitable for both global and local feature vectors. Of particular relevance to this chapter is the Fisherfaces algorithm of Belhumeur et al. [10] which is based on linear discriminant analysis (LDA). LDA methods separately model the within- individual and between- individual covariance of the data. The original technique [1] projected the data onto a new basis that maximizes the ratio

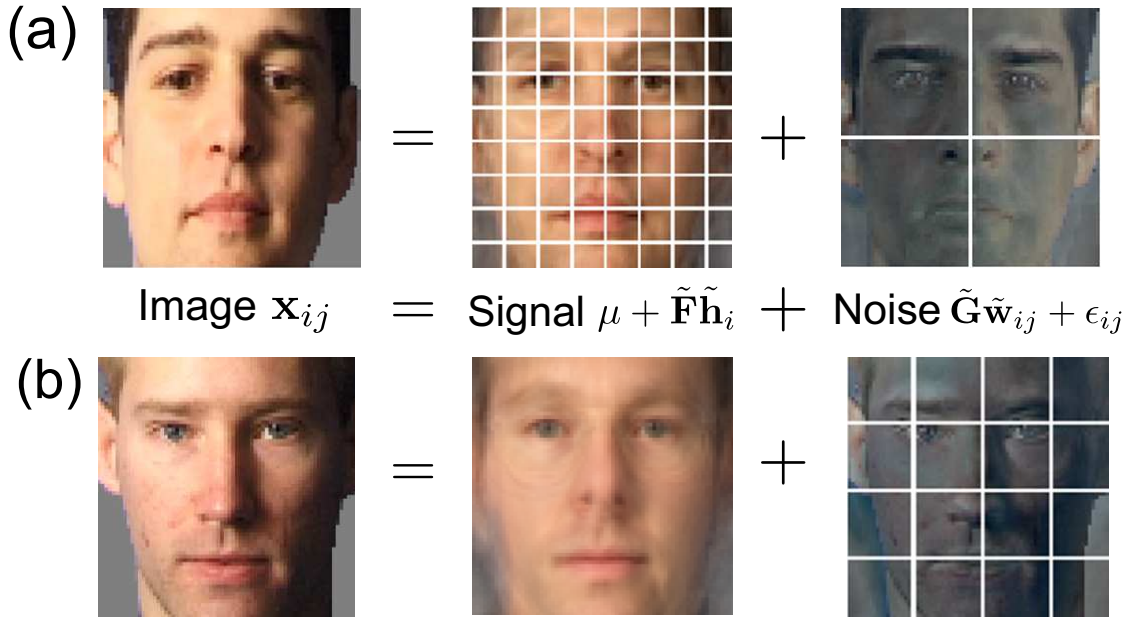


Figure 3.1: **Face images are described as a sum of signal and noise components and we investigate the spatial support of each.** Both signal and noise components are divided into regular grids of independent patches. The grid resolution is manipulated separately. (a) The signal component is divided into 8×8 patches, the noise component is divided into 2×2 patches. (b) Signal as 1×1 patches, noise as 4×4 patches.

of between- to within- individual variation in an attempt to improve the signal to noise ratio. In recent work, Ioffe [68] and Prince and Elder [111] have described probabilistic interpretations of this algorithm.

In this chapter we adapt the probabilistic LDA model of [111] to investigate the continuum between local and global approaches. Probabilistic LDA describes data as an additive mixture of signal (between-individual changes) and noise (within-individual changes). Here, we manipulate the spatial extent of the signal and noise components separately. In particular we break the signal and noise into regular grids of non-overlapping patches at various resolutions (see Figure 3.1). Several previous studies have considered breaking the image into patches ([20], [90], [93], [98]), but none have independently manipulated the scale of signal and noise elements.

By investigating face images as shown in Figure 3.2, we notice that there is independent identity information everywhere in the image. In other words identity information appears locally. However, within-individual variation, such as face expression, illumination and pose, cannot be understood at a small region of a face image. Therefore, we hypothesize that recognition performance will be best when the signal is local. However, we predict that performance will be worse when the noise is treated locally.

The structure of this chapter is as follows. In Section 3.2 we review statistical subspace algorithms and patch-based face recognition algorithms. In Section 3.3 we describe how to control the spatial extent of signal and noise elements and propose a new face recognition algorithm: Multi-scale PLDA. In Section 3.4.1 we discuss four controlled datasets used in our experiments. In Sections 3.4.2-3.4.4 we



Figure 3.2: **Signal exists locally and noise should be understood globally.** Each part of face includes independent identity information, for example the eyes, nose, mouth of person A and person B are all different. However, illumination, face expression and pose can only be understood by considering a large region.

present results on three controlled databases where our approach performs well and compare to other methods. In section 3.4.5 we present results on a fourth controlled database where performance is less good, and we discuss why this is the case. We also apply Multi-scale PLDA to a uncontrolled dataset to do face verification in section 3.5. Finally we draw a conclusion in section 3.6.

3.2 Related Works

3.2.1 Statistical Subspace Algorithms

Statistical subspace algorithms are important state of the art face recognition algorithms. In subspace algorithms face images are projected into a low dimensional subspace and then represented as a weighted sum of basis functions. Compared to image intensities, the new representation is more compact and increase the signal-to-noise ratio effectively. The Eigenfaces algorithm [132] was the first subspace algorithm and applied Principal Component Analysis (PCA) to reduce the dimensions. The projection by PCA maximizes the scatter of all face images and concentrates the data's energy. However, the scatter maximized by PCA is due not only to the between-individual scatter that is important for classification but also to the within-individual scatter that is not wanted.

To address this issue, the Fisherfaces algorithm [10] applied Linear Discriminant Analysis (LDA) to obtain a set of projections that maximizes the ratio of the between-individual scatter matrix to the within-individual scatter matrix. In the subspace obtained by LDA face images from different people are more spread out than images from the same person. Therefore, the Fisherfaces algorithm can improve performance when within-individual variation exists. Hastie et al. [60] interpreted LDA in a probabilistic context that LDA maximizing the ratio of the between-individual scatter matrix to the within-individual scatter matrix is mathematically equivalent to maximizing the likelihood of a Gaussian mixture model. This maximization process can be described as a linear regression of class label assignment. However, this type of regression is only useful when the class to be classified already exists in the training data. This assumption cannot satisfy the requirement of face recognition. In many face recognition evaluations

identities in training and test set are mutually exclusive.

In Gaussian mixture model, which LDA is mathematically equivalent to a probabilistic view, the class distribution is finite, which cannot handle unseen classes. Prince et al. [111] proposed Probabilistic Linear Discriminant Analysis (PLDA), which used hidden variables to represent classes and assumed a continuous distribution of these hidden variables. They marginalized over unknown hidden variables to obtain the capability to make inference about the unseen classes. As demonstrated in [111], PLDA produced better performance than the Fisherfaces algorithm. We will introduce the Eigenfaces algorithm, the Fisherfaces algorithm and PLDA briefly in the following text.

In the Eigenfaces algorithm a face image \mathbf{x} is represented by the following equation:

$$\mathbf{x} \approx \boldsymbol{\mu} + \hat{\boldsymbol{\Phi}}\boldsymbol{\omega}, \quad (3.1)$$

where \mathbf{x} is a pixel intensity vector obtained by concatenating the columns of pixels in a face image, $\boldsymbol{\mu}$ is the mean vector of all the training images, $\hat{\boldsymbol{\Phi}}$ contains the basis functions of the feature subspace in its columns, and $\boldsymbol{\omega}$ is a coefficient vector.

In training we aim to learn the basis functions $\hat{\boldsymbol{\Phi}}$. We assume there are n training images $\{\mathbf{x}_1 \cdots \mathbf{x}_n\}$, the total scatter matrix \mathbf{S} is

$$\mathbf{S} = \sum_{k=1}^n (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T. \quad (3.2)$$

In the Eigenfaces algorithm the basis function $\hat{\boldsymbol{\Phi}}$ is defined by maximizing the determinant of the total scatter matrix:

$$\hat{\boldsymbol{\Phi}} = \arg \max |\boldsymbol{\Phi}^T \mathbf{S} \boldsymbol{\Phi}|. \quad (3.3)$$

Similar to the Eigenfaces algorithm, we learn the basis functions of the feature subspace in the Fisherfaces algorithm. However, we learn the basis functions $\hat{\mathbf{W}}$ using Linear Discriminant Analysis instead of Principal Component Analysis. We assume there are n training images $\{\mathbf{x}_1 \cdots \mathbf{x}_n\}$ and each image belongs to one of m identities. The between-individual and the within-individual scatter matrices are computed by

$$\mathbf{S}_B = \sum_{c=1}^m N_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T \quad (3.4)$$

$$\mathbf{S}_W = \sum_{c=1}^m \sum_{k=1}^K (\mathbf{x}_k - \boldsymbol{\mu}_c)(\mathbf{x}_k - \boldsymbol{\mu}_c)^T, \quad (3.5)$$

where N_c denotes the image number of identity c , $\boldsymbol{\mu}_c$ denotes the mean image of identity c , $\boldsymbol{\mu}$ denotes the mean vector of all the training images, and K denotes the image number of identity c .

In the Fisherfaces algorithm the basis functions $\hat{\mathbf{W}}$ are defined by maximizing the ratio of the determinant of the between-individual scatter matrix and that of the within-individual scatter matrix:

$$\hat{\mathbf{W}} = \arg \max \frac{|\mathbf{W}^T \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_W \mathbf{W}|}. \quad (3.6)$$

Probabilistic Linear Discriminant Analysis (PLDA) is a probabilistic version of the Fisherfaces algorithm. In PLDA a face image \mathbf{x}_{ij} is represented as:

$$\mathbf{x}_{ij} = \boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ij}, \quad (3.7)$$

where \mathbf{x}_{ij} denotes the j^{th} of J training images of each of the i^{th} of I individuals, $\boldsymbol{\mu}$ is the mean of the data, \mathbf{F} is a matrix with the basis vectors of the between-individual subspace in its columns, and \mathbf{h}_i is an identity variable that is constant for all J images $\mathbf{x}_{i1\dots iJ}$ of person i . The term \mathbf{G} is a matrix with the basis vectors of the within-individual subspace in its columns. The term \mathbf{w}_{ij} represents a position in this subspace. The term $\boldsymbol{\epsilon}_{ij}$ denotes stochastic noise with diagonal covariance $\boldsymbol{\Sigma}$. The term $\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i$ consists of the signal and accounts for between-individual variation. For a given individual, this term is constant. The term $\mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ij}$ consists of the noise or within-individual variation. It explains why two images of the same individual do not look identical.

We can alternately describe the generative process in terms of conditional probabilities:

$$Pr(\mathbf{x}_{ij}|\mathbf{h}_i, \mathbf{w}_{ij}) = \mathcal{G}_x[\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij}, \boldsymbol{\Sigma}] \quad (3.8)$$

$$Pr(\mathbf{h}_i) = \mathcal{G}_{\mathbf{h}_i}[\mathbf{0}, \mathbf{I}] \quad (3.9)$$

$$Pr(\mathbf{w}_{ij}) = \mathcal{G}_{\mathbf{w}_{ij}}[\mathbf{0}, \mathbf{I}], \quad (3.10)$$

where $\mathcal{G}_o[\boldsymbol{\rho}, \boldsymbol{\varsigma}]$ denotes a Gaussian in \mathcal{o} with mean $\boldsymbol{\rho}$ and covariance $\boldsymbol{\varsigma}$; the term \mathbf{I} denotes an identity matrix.

In training, the Expectation Maximization (EM) algorithm is applied to learn the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \mathbf{F}, \mathbf{G}, \boldsymbol{\Sigma}\}$. In test they compare the likelihood of two images when they are assumed to match and not match.

In this chapter, we use PLDA as a platform to verify our hypothesis.

3.2.2 Patch-based Face Representation Methods

Face representation methods play an important role in face recognition. The goal of face representation methods is to obtain a compact form to describe face images but retain sufficient discriminant information. A good face representation method can capture sufficient identity information and is robust to within-individual variation. Face representation methods can be divided into two categories: global and local representation methods.

The global face representation methods model the statistic regarding the whole face. Examples include the Eigenfaces algorithm [132], the Fisherfaces algorithms [10], the Active Appearance Model [34]. The global representation methods perform well for face images with limited variation. However, their performance depends on accurate image registration and cannot deal with geometric transformation and occlusion.

The local face representation methods model the local statistic regarding face parts. Examples include the Elastic Bunch Graph Matching algorithm [139], the Fisher Vector Faces algorithm [124]. A face image is normally represented by a vector of local features. To obtain the local features, a set of fiducial points are normally firstly detected. Fiducial points are a set of salient facial parts. They are usually located on the corners of the eyebrows, the corners of the eyes, the tip of the nose, the corners of the lips, etc. After fiducial points are obtained, multiple image descriptors are used to characterize the

region around each point. Finally, an vector to represent the whole image is formed by concatenating descriptors of each point. Examples of image descriptors include Local Binary Patterns (LBP) descriptor [102], Scale Invariant Feature Transform (SIFT) descriptor [88], Gabor Filter [41], etc. Compared with global face representation methods, local face representation methods are more invariant to within-individual variations, such as expression and pose changes. However, local face representation methods normally treat image parts independently or conditionally independently and may not consider the global connection among image parts. Moreover, local representation methods apply a sparse representation and may lose potentially useful information.

Patch-based representation methods combine the advantages of global and local representation methods. In a typical patch-based representation method a face image is represented as a collection of patches. The configurations of image patches can be grids of non-overlapping or overlapping patches. Patch-based representation methods provide a dense representation to face images and retain the global structure of face image. Moreover, patch-based representation methods allow us to vary the patch configuration and model the covariance at a pre-defined scale to avoid the expensive computation of the full covariance matrix. Yamada et al. [72] proposed a patch-based representation method to do face recognition across pose. Their experiments demonstrated that their patch-based representations are more robust to pose variation than the global representation methods. Lucey et al. [90] inherited the principle of Yamada's algorithm and modelled the relation of corresponding patches from images with different poses. Their experiments confirmed Yamada's conclusion.

Despite the aforementioned successful applications of patch-based representation methods, there has never been an experiment to manipulate the patch configuration of images to affect the spatial support of between-individual and within-individual basis functions of a statistical subspace algorithm. In this chapter we will explore this intrinsic combination of statistical subspace algorithms and patch-based face representation methods.

3.3 Multi-scale PLDA

As shown in Figure 3.1 we vary the effect of localization of the basis functions of the between- and within- individual subspace \mathbf{F} and \mathbf{G} respectively. The signal component is divided into a grid of P regular square non-overlapping patches. In a similar way, the noise component is divided into Q patches. We increase P and Q to make the spatial support of the basis function more local. So when the value of P or Q is 1, 4, 16, 64, the grid resolution of the signal or noise component is 1×1 , 2×2 , 4×4 and 8×8 accordingly.

The generative process of Multi-scale PLDA can be described by the following equation:

$$\mathbf{x}_{ij} = \boldsymbol{\mu} + \sum_{p=1}^P \mathbf{F}^p \mathbf{h}_i^p + \sum_{q=1}^Q \mathbf{G}^q \mathbf{w}_{ij}^q + \boldsymbol{\epsilon}_{ij}, \quad (3.11)$$

Where $\boldsymbol{\mu}$ is the mean of the data, \mathbf{F}^p denotes the basis vectors of between-individual variation for the p^{th} patch. The term \mathbf{h}_i^p represents the weighting of these basis vectors for the i^{th} individual. Similarly, \mathbf{G}^q contains the basis vectors of within-individual variation for the q^{th} patch. The term \mathbf{w}_{ij}^q denotes the

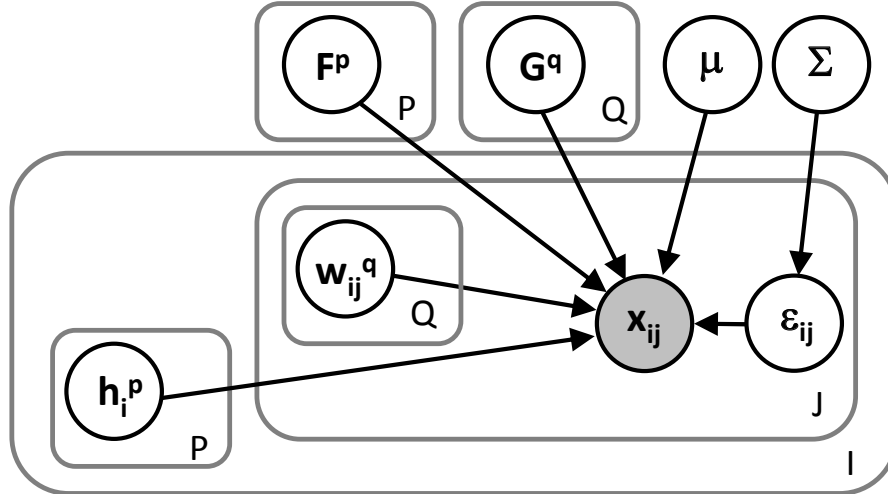


Figure 3.3: **Graphical model for Multi-scale PLDA** showing the image x_{ij} , hidden identity variables h_i^p , hidden within-individual variables w_{ij}^q and basis functions for between-individual subspace F^p , basis functions for within-individual subspace G^q , mean of all training images μ , and diagonal covariance matrix Σ for the stochastic noise ϵ_{ij} in images.

weighting of these basis vectors for the j^{th} image of the i^{th} individual. In spite of the presence of patches, the dimensions of basis vector $F^{1\dots P}$ and $G^{1\dots Q}$ are still the size of the full data vector x . However, they become sparse and non-zero entries only exist for pixels in the patch in question. The relation of variables is shown in Figure 3.3 and the model is illustrated in Figure 3.4.

The generative formulation in Equation 3.11 can be rewritten in the form of the original PLDA algorithm:

$$x_{ij} = \mu + \tilde{F}\tilde{h}_i + \tilde{G}\tilde{w}_{ij} + \epsilon_{ij}, \quad (3.12)$$

where $\tilde{F} = [F^1 \dots F^P]$, $\tilde{G} = [G^1 \dots G^Q]$, $\tilde{h} = [h^1 \dots h^P]^T$ and $\tilde{w}^T = [w^1 \dots w^Q]^T$.

Unfortunately, this relatively small change significantly complicates the learning and inference algorithms: firstly, the system of equations may now be considerably bigger (we may have a large number of basis functions at each separate block of the image) and this makes straightforward inversion of matrices in the learning and inference steps impossible. Second, we must now ensure that the sparsity structure of the matrices \tilde{F} and \tilde{G} are retained. Matrices \tilde{F} and \tilde{G} are Block diagonal.

3.3.1 Learning

It would be easy to estimate the parameters $\theta = \{\mu, \tilde{F}, \tilde{G}, \Sigma\}$ if we knew the hidden variables $\tilde{h}_i, \tilde{w}_{ij}$. Likewise, it would be easy to infer the hidden variables if we knew the parameters. This type of “chicken and egg” problem is well suited to the expectation-maximization (EM) algorithm [43]. In the expectation- or E-step we will calculate a joint posterior distribution over the hidden variables. In the maximization- or M-step we update the parameters θ . We now consider each in turn:

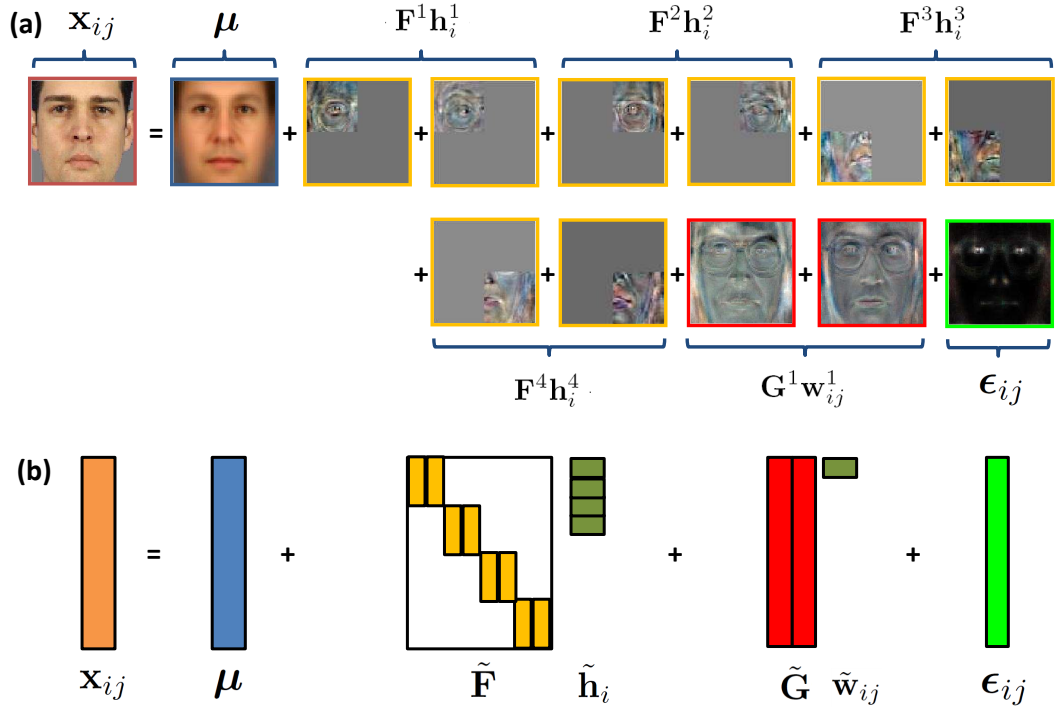


Figure 3.4: **Structure of Multi-Scale PLDA model.** (a) We describe images as the sum of a mean μ , the between-individual variation $\sum_{p=1}^4 \mathbf{F}^p \mathbf{h}_i^p$, the within-individual variation $\sum_{q=1}^1 \mathbf{G}^q \mathbf{w}_{ij}^q$, and per pixel noise ϵ (image shows per-pixel variance). Here the signal is analyzed independently in $P=4$ patches, corresponding to the image quadrants. Each has $D_f = 2$ basis functions associated with them. In this example, the noise is analyzed on a global scale using only $Q=1$ patch which has $D_g = 2$ basis functions associated with it. (b) We can write this same model in matrix form. Now the localization is embodied in the structure of sparsity of the matrices $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$.

E-Step: In the E-step we aim to take all of the data $\mathbf{x}_{i1} \dots \mathbf{x}_{iJ}$ pertaining to one individual and calculate the joint posterior distribution of all of the hidden variables $\tilde{\mathbf{h}}_i, \tilde{\mathbf{w}}_{i1} \dots \tilde{\mathbf{w}}_{iJ}$. To accomplish this, we express this problem in a composite form:

$$\begin{bmatrix} \mathbf{x}_{i1} \\ \mathbf{x}_{i2} \\ \vdots \\ \mathbf{x}_{iJ} \end{bmatrix} = \begin{bmatrix} \mu \\ \mu \\ \vdots \\ \mu \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{F}} & \tilde{\mathbf{G}} & \mathbf{0} & \dots & \mathbf{0} \\ \tilde{\mathbf{F}} & \mathbf{0} & \tilde{\mathbf{G}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{G}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{h}}_i \\ \tilde{\mathbf{w}}_{i1} \\ \tilde{\mathbf{w}}_{i2} \\ \vdots \\ \tilde{\mathbf{w}}_{iN} \end{bmatrix} + \begin{bmatrix} \epsilon_{i1} \\ \epsilon_{i2} \\ \vdots \\ \epsilon_{iN} \end{bmatrix} \quad (3.13)$$

or, giving names to these composite matrices:

$$\mathbf{x}'_i = \mu' + \mathbf{A} \mathbf{y}_i + \epsilon'_i. \quad (3.14)$$

In probabilistic notation we can equivalently write:

$$Pr(\mathbf{x}'_i | \mathbf{y}_i) = \mathcal{G}_{\mathbf{x}'_i}[\mu' + \mathbf{A} \mathbf{y}_i, \Sigma'] \quad (3.15)$$

$$Pr(\mathbf{y}_i) = \mathcal{G}_{\mathbf{y}_i}[\mathbf{0}, \mathbf{I}], \quad (3.16)$$

where

$$\Sigma' = \begin{bmatrix} \Sigma & & & \\ & \Sigma & & \\ & & \ddots & \\ & & & \Sigma \end{bmatrix}. \quad (3.17)$$

Applying Bayes' rule to calculate the posterior, we get:

$$Pr(\mathbf{y}_i | \mathbf{x}'_i, \theta) \propto Pr(\mathbf{x}'_i | \mathbf{y}_i, \theta) Pr(\mathbf{y}_i). \quad (3.18)$$

Since both terms on the right are Gaussian, the term on the left must also be Gaussian. It can be shown as in [111] that the first two moments are:

$$E[\mathbf{y}_i] = (\mathbf{A}^T \Sigma'^{-1} \mathbf{A} + \mathbf{I})^{-1} \mathbf{A}^T \Sigma'^{-1} (\mathbf{x}'_i - \boldsymbol{\mu}') \quad (3.19)$$

$$E[\mathbf{y}_i \mathbf{y}_i^T] = (\mathbf{A}^T \Sigma'^{-1} \mathbf{A}^T + \mathbf{I})^{-1} + E[\mathbf{y}_i] E[\mathbf{y}_i]^T. \quad (3.20)$$

In practice, these quantities are hard to calculate. For example, consider $P=64$ blocks, $\mathbf{F}^{1 \dots P}$ representing the signal, and $Q=64$ blocks, $\mathbf{G}^{1 \dots Q}$ representing the noise. If each block consists of $N_p = N_q$ basis functions and there are J images of person I , then the matrix $\mathbf{A}^T \Sigma'^{-1} \mathbf{A} + \mathbf{I}$ will be of dimension $(N_p \times N_q \times (J + 1))^2$ which can be very large. However, the matrix has considerable structure:

$$(\mathbf{A}^T \Sigma'^{-1} \mathbf{A} + \mathbf{I})^{-1} = \begin{bmatrix} J\tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{F}}^T + \mathbf{I} & \tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{G}} & \dots & \tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{G}} \\ \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{F}} & \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{G}} + \mathbf{I} & & \\ \vdots & & \ddots & \vdots \\ \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{F}} & & \dots & \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{G}} + \mathbf{I} \end{bmatrix}^{-1}. \quad (3.21)$$

Moreover, the blocks of this matrix themselves exhibit structure. For example the top-left quadrant $J\tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{F}}^T + \mathbf{I}$ is block diagonal, as is the bottom right. We use Schur's lemma to exploit this structure in inversion. The concept of Schur's lemma is to divide a matrix into four components and the inversion of the matrix can be described by polynomials of the four components. Equation 3.22 shows the inversion of matrix \mathbf{U} using Schur's lemma:

$$\begin{aligned} \mathbf{U}^{-1} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \\ \mathbf{V}_3 & \mathbf{V}_4 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (\mathbf{V}_1 - \mathbf{V}_2 \mathbf{V}_4^{-1} \mathbf{V}_3)^{-1} & -(\mathbf{V}_1 - \mathbf{V}_2 \mathbf{V}_4^{-1} \mathbf{V}_3)^{-1} \mathbf{V}_2 \mathbf{V}_4^{-1} \\ -\mathbf{V}_4^{-1} \mathbf{V}_3 (\mathbf{V}_1 - \mathbf{V}_2 \mathbf{V}_4^{-1} \mathbf{V}_3)^{-1} & \mathbf{V}_4^{-1} + \mathbf{V}_4^{-1} \mathbf{V}_3 (\mathbf{V}_1 - \mathbf{V}_2 \mathbf{V}_4^{-1} \mathbf{V}_3)^{-1} \mathbf{V}_2 \mathbf{V}_4^{-1} \end{bmatrix}, \end{aligned} \quad (3.22)$$

where $(\mathbf{V}_1 - \mathbf{V}_2 \mathbf{V}_4^{-1} \mathbf{V}_3)^{-1}$ is called the Schur Complement.

Applying Shur's lemma, we divide equation 3.21 to four parts:

$$\mathbf{V}_1 = J\tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{F}}^T + \mathbf{I} \quad (3.23)$$

$$\mathbf{V}_2 = \begin{bmatrix} \tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{G}} & \dots & \tilde{\mathbf{F}}^T \Sigma^{-1} \tilde{\mathbf{G}} \end{bmatrix} \quad (3.24)$$

$$\mathbf{V}_3 = \begin{bmatrix} \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{F}} \\ \vdots \\ \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{F}} \end{bmatrix} \quad (3.25)$$

$$\mathbf{V}_4 = \begin{bmatrix} \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{G}} + \mathbf{I} & & \\ & \ddots & \\ & & \tilde{\mathbf{G}}^T \Sigma^{-1} \tilde{\mathbf{G}} + \mathbf{I} \end{bmatrix}. \quad (3.26)$$

The matrix in equation 3.26 is a diagonal block matrix, so we invert each matrix block to obtain its inverse. Then we follow Shur's lemma to compute the term 3.21. By using Schur's lemma, the effective dimension of the inverse is reduced from $(P \times D_f + P \times D_f \times J) \times (P \times D_f + Q \times D_g \times J)$ to P times of $(D_f \times D_f)$ and Q times of $(D_g \times D_g)$, where D_f is the number of basis functions for between-individual subspace and D_g is the number of basis functions for the within-individual subspace. Assuming $P = 64$, $Q = 16$, $D_f = 64$, $D_g = 64$, $J = 10$, we need to invert a 59392×59392 matrix if we compute the term 3.21 directly. However, after applying Shur's lemma, we only invert 64 matrixes with the size 64×64 and 16 matrixes with the size 64×64 .

M-Step: In the M-Step, we aim to update the values of the parameters $\theta = \{\mu, \tilde{\mathbf{F}}, \tilde{\mathbf{G}}, \Sigma\}$. We must do this in such a way that the sparsity structure of the matrices $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ is maintained. We perform a separate calculation for every pixel (row of the generative equation) ensuring that the appropriate elements remain zero. We first write a single equation for each observed data vector:

$$\mathbf{x}_{ij} = \mu + \begin{bmatrix} \tilde{\mathbf{F}} & \tilde{\mathbf{G}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{h}}_i \\ \tilde{\mathbf{w}}_{ij} \end{bmatrix} + \epsilon_{ij}. \quad (3.27)$$

This has the form:

$$\mathbf{x}_{ij} = \mu + \mathbf{B} \quad \mathbf{z}_{ij} + \epsilon_{ij}. \quad (3.28)$$

We optimize:

$$Q(\theta_t, \theta_{t-1}) = \sum_{i=1}^I \sum_{j=1}^J \int Pr(\mathbf{z}_i | \mathbf{x}_{i1...iJ}, \theta_{t-1}) \log[Pr(\mathbf{x}_{ij} | \mathbf{z}_i) Pr(\mathbf{z}_i)] d\mathbf{z}_i, \quad (3.29)$$

where t is the iteration index. The first log probability term in Equation 3.29 can be written as:

$$\log[Pr(\mathbf{x}_{ij} | \mathbf{z}_i, \theta_t)] = \kappa - 0.5 (\log |\Sigma^{-1}| + (\mathbf{x}_{ij} - \mu - \mathbf{B}\mathbf{z}_i)^T \Sigma^{-1} (\mathbf{x}_{ij} - \mu - \mathbf{B}\mathbf{z}_i)), \quad (3.30)$$

where κ is an unimportant constant. Since the matrix Σ^{-1} is diagonal, this can be written as a sum of terms over the N pixels in the image:

$$\log[Pr(\mathbf{x}_{ij} | \mathbf{z}_i, \theta_t)] = \kappa - 0.5 \sum_{n=1}^N (\log |\sigma_n^2| + (\mathbf{x}_{ijn} - \mu_n - \mathbf{b}_n \mathbf{z}_{in})^T \sigma_n^{-2} (\mathbf{x}_{ijn} - \mu_n - \mathbf{b}_n \mathbf{z}_{in})), \quad (3.31)$$

where \mathbf{x}_{ijn} refers to the n^{th} pixel of the j^{th} image of the i^{th} individual, $\boldsymbol{\mu}_n$ denotes the n^{th} pixel of the mean vector and σ_n^2 represents the n^{th} entry in the diagonal matrix Σ . The term \mathbf{b}_n describes the n^{th} row of the matrix \mathbf{B} , modified so only the non-zero entries remain. The term \mathbf{z}_{in} consists of the entries in the vector \mathbf{z}_i that correspond to the non-zero entries of the n^{th} row of \mathbf{B} .

We substitute this term into Equation 3.29 and take derivatives with respect to $\boldsymbol{\mu}_n$, \mathbf{b}_n and σ_n^2 . The second log term in Equation 3.29 has no dependence on these parameters. We equate these derivatives to zero and re-arrange to provide the following update rules:

$$\boldsymbol{\mu}_n = \frac{1}{IJ} \sum_{i,j} \mathbf{x}_{ijn} \quad (3.32)$$

$$\mathbf{b}_n = \left(\sum_{i,j} (\mathbf{x}_{ijn} - \boldsymbol{\mu}_n) E[\mathbf{z}_{in}]^T \right) \left(\sum_{i,j} E[\mathbf{z}_{in} \mathbf{z}_{in}^T] \right)^{-1} \quad (3.33)$$

$$\sigma^2 = \frac{1}{IJ} \sum_{i,j} \mathbf{diag} [(\mathbf{x}_{ijn} - \boldsymbol{\mu}_n)(\mathbf{x}_{ijn} - \boldsymbol{\mu}_n)^T - \mathbf{b}_n E[\mathbf{z}_{in}] (\mathbf{x}_{ijn} - \boldsymbol{\mu}_n)^T], \quad (3.34)$$

where **diag** represents retaining only the diagonal elements of a matrix. The expectation terms $E[\mathbf{z}_i]$ and $E[\mathbf{z}_i \mathbf{z}_i^T]$ can be extracted from Equations 3.19 and 3.20 using the equivalence between Equations 3.13 and 3.14. The updated values of \mathbf{F} and \mathbf{G} are retrieved from $\mathbf{b}_{1\dots N}$.

3.3.2 Inference

We perform recognition by comparing the likelihood of different models of the data. For example, consider a closed set face recognition task in which we wish to know whether the probe face \mathbf{x}_p matches gallery faces \mathbf{x}_1 or \mathbf{x}_2 . We build two models \mathcal{M}_1 and \mathcal{M}_2 corresponding to these two situations and compare them with Bayes' rule:

$$Pr(\mathcal{M}_1 | \mathbf{x}_{1,2,p}) = \frac{Pr(\mathbf{x}_{1,2,p} | \mathcal{M}_1) Pr(\mathcal{M}_1)}{\sum_{k=1}^2 Pr(\mathbf{x}_{1,2,p} | \mathcal{M}_k) Pr(\mathcal{M}_k)}. \quad (3.35)$$

Model \mathcal{M}_1 hypothesizes that the probe face \mathbf{x}_p shares an identity \mathbf{h}_1 with gallery image \mathbf{x}_1 although the noise vectors still differ. Probe face \mathbf{x}_2 has a different identity \mathbf{h}_2 . We write the generative equation for this data as:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \mathbf{F} & \mathbf{0} & \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{G} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \boldsymbol{\epsilon}_p \end{bmatrix}. \quad (3.36)$$

To calculate the likelihood for model 2 we assume that \mathbf{x}_p has to share an identity with gallery face \mathbf{x}_2 to give a similar generative equation:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} + \begin{bmatrix} \mathbf{F} & \mathbf{0} & \mathbf{G} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \boldsymbol{\epsilon}_p \end{bmatrix}. \quad (3.37)$$

Each of these generative equations is of the form

$$\mathbf{x}' = \boldsymbol{\mu}' + \mathbf{A}\mathbf{y} + \boldsymbol{\epsilon}'. \quad (3.38)$$

We can write this more precisely in probabilistic form:

$$Pr(\mathbf{x}'|\mathbf{y}) = \mathcal{G}_{\mathbf{x}'}[\boldsymbol{\mu} + \mathbf{A}\mathbf{y}, \boldsymbol{\Sigma}'] \quad (3.39)$$

$$Pr(\mathbf{y}) = \mathcal{G}_{\mathbf{y}}[\mathbf{0}, \mathbf{I}], \quad (3.40)$$

where we define $\boldsymbol{\Sigma}'$ similarly to in Equation 3.17. We note that Equations 3.39-3.40 describe a factor analyzer. It is possible to marginalize over the hidden variable \mathbf{y} and find a closed form expression for the likelihood:

$$Pr(\mathbf{x}') = \int Pr(\mathbf{x}'|\mathbf{y})Pr(\mathbf{y})d\mathbf{y} = \mathcal{G}_{\mathbf{x}'}[\boldsymbol{\mu}, \mathbf{A}\mathbf{A}^T + \boldsymbol{\Sigma}]. \quad (3.41)$$

This can be calculated efficiently by (i) calculating the likelihoods separately for all independent terms (for example Equation 3.36 can be broken down into two parts, one of which contains only \mathbf{x}_1 and \mathbf{x}_p and the other contains only \mathbf{x}_2) and (ii) exploiting the sparse structure of the matrix $\mathbf{A}\mathbf{A}^T + \boldsymbol{\Sigma}$. One way to do this is to use the matrix inversion lemma to convert the precision matrix so that:

$$(\mathbf{A}\mathbf{A}^T + \boldsymbol{\Sigma})^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\mathbf{A}(\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A} + \mathbf{I})^{-1}\mathbf{A}^T\boldsymbol{\Sigma}^{-1}. \quad (3.42)$$

The inverse term on the right hand side can then be inverted in a similar manner to the similar terms of the E-Step of the learning algorithm presented in section 3.3.1.

3.4 Experiments in Constrained Databases

3.4.1 Datasets and Preprocessing

We investigate closed set identification using four datasets, each of which has different properties (see Figure 3.5). We discuss the preprocessing of each in turn.

XM2VTS Frontal: The training set consists of 4 images each of 195 individuals. The test set consists of 100 different individuals, where gallery images were taken from the first recording session and the probes from the fourth session. The color images were affine aligned and resized to size 70×70 . The raw RGB pixel values were concatenated into a vector of length $70 \times 70 \times 3 = 14700$.

XM2VTS Lighting: The training set consists of 7 images each of 195 individuals and contained 2 lighting conditions. For each individual there were 5 images under frontal lighting and 2 under side-lighting. The test set consists of 100 different individuals, where the gallery images were taken from the first recording session and were under frontal lighting and the probe images were taken from the fourth session and were lit from the side. As for the XM2VTS frontal dataset, the images were affine aligned and resized to size 70×70 . The raw RGB pixel values were concatenated into a vector of length $70 \times 70 \times 3 = 14700$.

Yale: The data were divided into 7 sets of training / test data as in [23]. The same 15 individuals are present in training and test phases. We train with 2-8 images of each person depending on the condition.

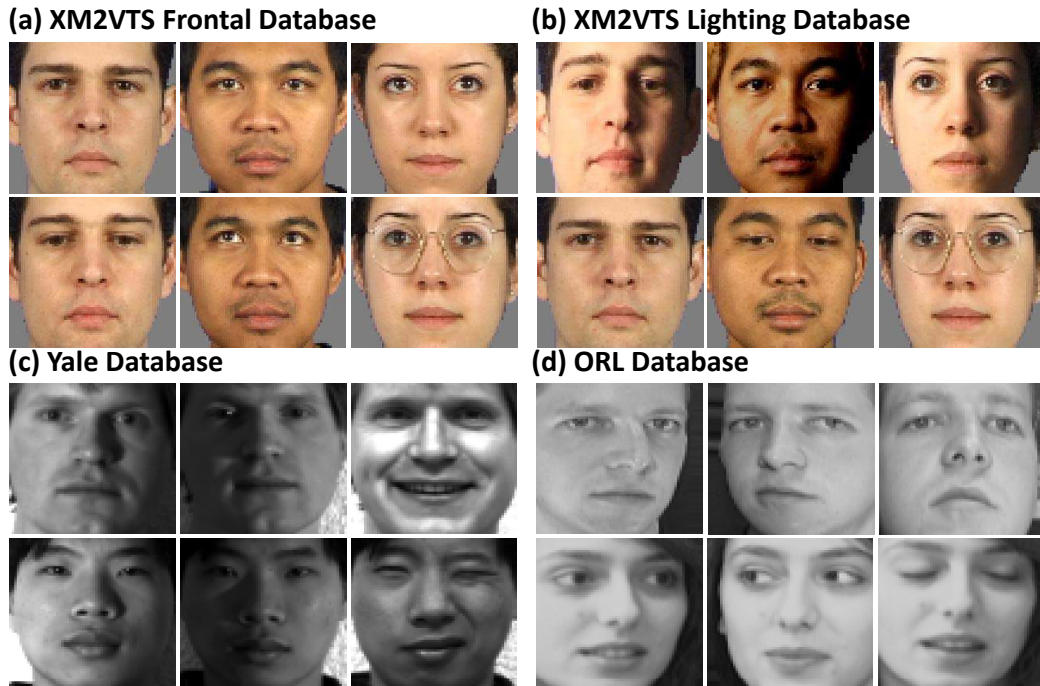


Figure 3.5: **Datasets used in this chapter.** (a) The XM2VTS frontal database contains frontal faces under diffuse lighting. (b) The XM2VTS lighting dataset contains frontal faces viewed under two lighting conditions. (c) The Yale dataset contains frontal faces with variations in expression and lighting. (d) The ORL dataset contains variations in pose.

These images also form the gallery. The probe images consist of the remaining faces. Lighting and facial expressions vary widely across training and test data. Each image was grayscale and 64×64 pixels in size.

ORL: As for the Yale dataset, the data were divided into 7 sets of training / test data (see [23]). The same 40 individuals are present in training and test phases. We train with 2-8 images of each person depending on the condition. These images also form the gallery. The probe images consist of the remaining faces. Each was grayscale and 64×64 pixels.

All models were trained using 6 iterations of the EM algorithm and the model parameters θ are initialized to random values. There are two sets of parameters in our model: (i) the number of patches for signal and noise and (ii) number of basis functions for each signal and noise patch. The latter two parameters were always varied together in our experiments and will be referred to as “subspace dimension”.

3.4.2 Experiments for Frontal Lighting Data Set (XM2VTS)

In Table 3.1 we present % correct results for face identification using the XM2VTS frontal dataset and a model with subspace dimension of 64. The results show that recognition generally gets better as the number of signal patches P increases (signal basis functions become more local). However, performance declines as the number of noise patches Q increases (noise basis functions become more local). Peak

performance is 99% when the noise has only 4 patches, but decreases when the signal is broken into either 16 or 64 patches.

Signal becomes more local →

	Q \ P	1	4	16	64
1	89%	91%	97%	97%	
4	85%	93%	99%	99%	
16	78%	88%	96%	98%	
64	71%	77%	89%	97%	

↓ **Noise becomes more local**

Table 3.1: % Correct results for the XM2VTS frontal data set as we vary patch resolution P and Q of signal and noise respectively. The results show that the recognition performance increases as the number of signal patches P increases (signal is treated more locally). However, performance drops as the number of noise patches Q increases (noise is treated more locally).

In Figure 3.6 we investigate performance as a function of the number of basis functions associated with each signal and noise patch (subspace dimension). The graph shows that best performance is reliably achieved when the signal is more local and the noise is more global. The performance falls off rapidly with large subspace sizes when both the signal and noise are local. This may be because the total number of basis functions in the columns of matrices \mathbf{F}^p and \mathbf{G}^q becomes similar to the number of data values in each patch.

In Figure 3.7 we compare performance to our own implementations of a number of contemporary algorithms that use completely global representations. Our performance is superior to that for PLDA [111], a second PLDA algorithm [68], the Fisherfaces algorithm [10], Dual Space LDA [137], the Bayesian face algorithm [97], and the Eigenfaces algorithm [132].

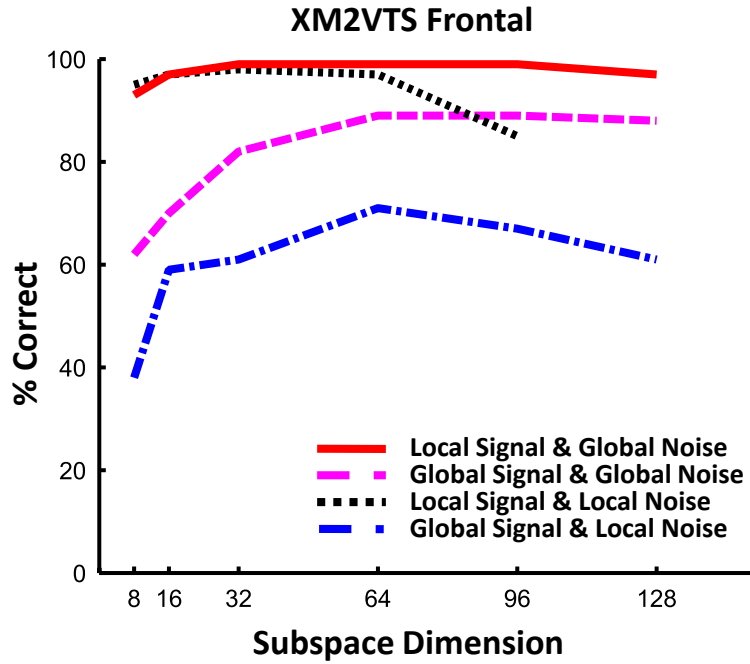


Figure 3.6: % Correct face identification for the XM2VTS frontal dataset as a function of signal and noise subspace size when signal/noise are local/local ($P=64, Q=64$), global/global ($P=1, Q=1$), global/local ($P=1, Q=64$), local/global ($P=64, Q=4$). Performance is best in the latter condition.

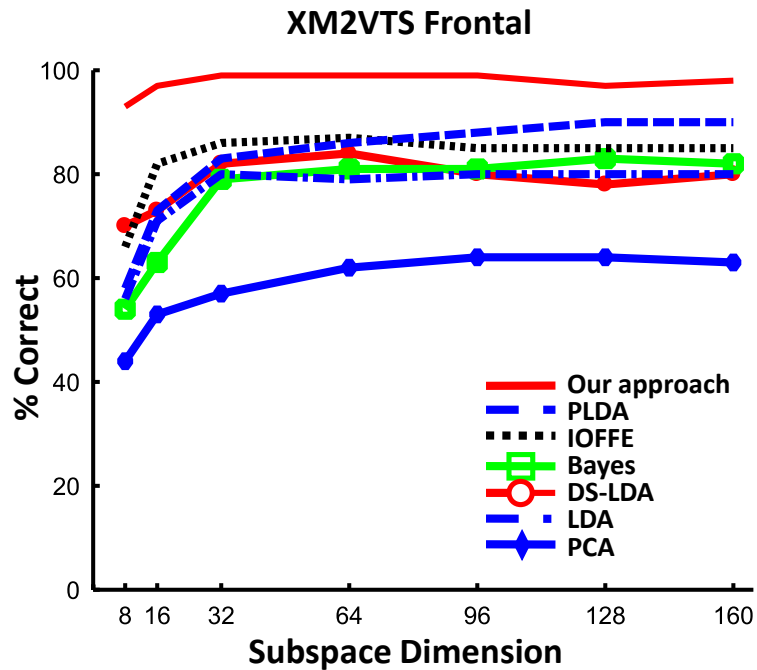


Figure 3.7: % Correct face identification for the XM2VTS frontal dataset as a function of signal and noise subspace size when signal/noise are local/global ($P=64, Q=4$). Results compare favorably to PLDA [111], Ioffe's PLDA algorithm [68], the Fisherfaces algorithm [10], the Dual Space LDA algorithm [137], the Bayesian face algorithm [97], and the Eigenfaces algorithm [132].

3.4.3 Experiments for Illumination Variation Data Set (XM2VTS Lighting)

In Table 3.2 we present % correct results for face identification using the XM2VTS lighting dataset and a model with subspace dimensions 64. Unsurprisingly, the performance is worse than for the dataset containing only frontal images. However, the pattern of results remains the same. Performance improves as the signal basis functions become more localized, but worse as the noise basis functions become more localized. Peak performance is 91% when the noise has only 4 patches, but the signal is broken into 64 patches. Figure 3.8 confirms that good performance is reliably achieved when the noise basis functions span a large part of the image, but the signal is very local regardless of the subspace dimensions used. Figure 3.9 shows that performance compares favorably to other algorithms.

Signal becomes more local \longrightarrow

		P			
		1	4	16	64
Noise becomes more local \downarrow	Q				
	1	80%	82%	90%	83%
	4	76%	89%	87%	91%
	16	70%	75%	90%	85%
	64	37%	56%	76%	84%

Table 3.2: % Correct results for the XM2VTS lighting dataset as we vary patch resolution P and Q of signal and noise respectively.

We do not apply any illumination preprocessing in the experiments illustrated in Figure 3.8 and 3.9 because illumination preprocessing cause a performance drop in our experiments. For example, when the subspace dimensions are set to 64, the patch number of the signal component P is 64, the patch number of the noise component Q is 4, the performance is 91% correct without any preprocessing. When we use histogram equalization to preprocess the images, the performance becomes 90%. When we use the preprocessing method proposed by Tan and Triggs [129], which is series of steps including Gamma correction, Difference of Gaussian filtering and contrast equalization, we only achieve 85% correct. We hypothesize that the reason is that some useful discriminative information is discarded during the preprocessing.

To fit the lighting condition in the XM2VTS Lighting database, we try different patch division methods to manipulate the degree of localization of signal and noise component. In the above experiments, we divided the signal and noise component into a regular grid of patches. However, we can also divide the signal component into a regular grid but divide the noise component into columns. In this case, our experiments show that recognition performance improves. Figure 3.10 shows two different patch division methods and the second division method performs better than the first method for XM2VTS lighting database. We conjecture that the second division method estimates left lit better.

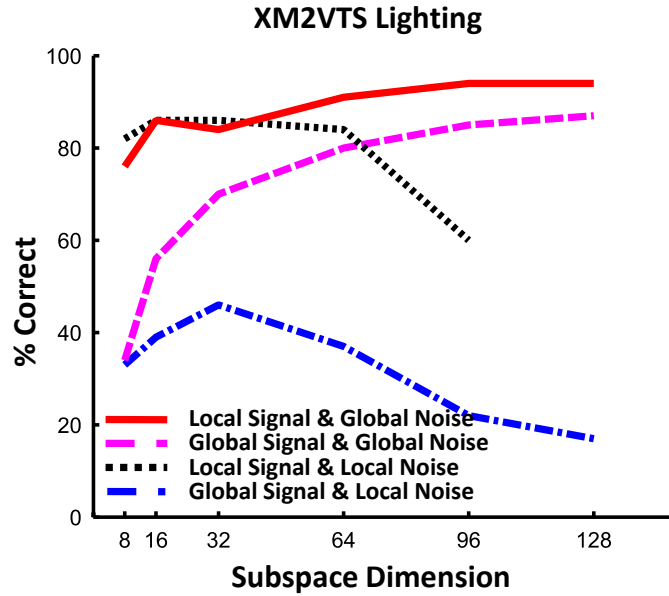


Figure 3.8: % Correct face identification for the XM2VTS lighting dataset as a function of signal and noise subspace size when signal/noise are local/local ($P=64, Q=64$), global/global ($P=1, Q=1$), global/local ($P=1, Q=64$), local/global ($P=64, Q=4$). A similar pattern is revealed as on the XM2VTS frontal dataset. Performance is best when $P=64$ and $Q=4$.

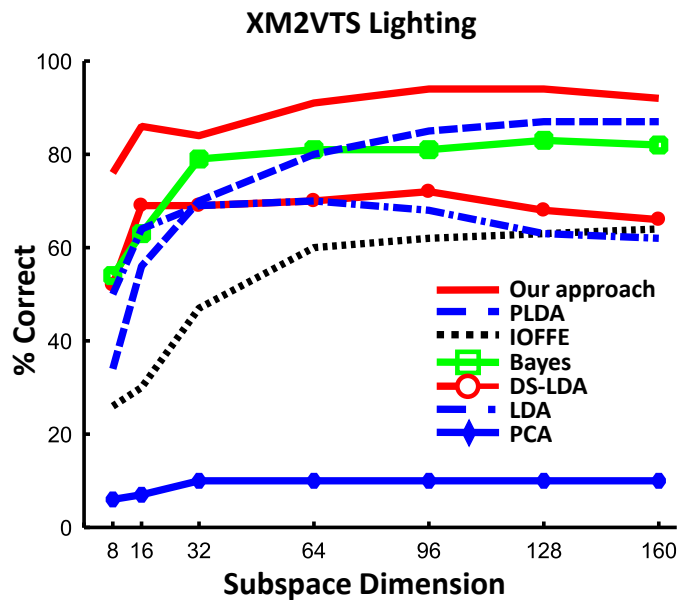


Figure 3.9: % Correct face identification for XM2VTS lighting dataset as a function of signal and noise subspace size when signal/noise are local/global ($P=64, Q=4$). Our results are better than PLDA [111], Ioffe's PLDA algorithm [68], the Fisherfaces algorithm [10], the Dual Space LDA algorithm [137], the Bayesian face algorithm [97] and the Eigenfaces algorithm [132].

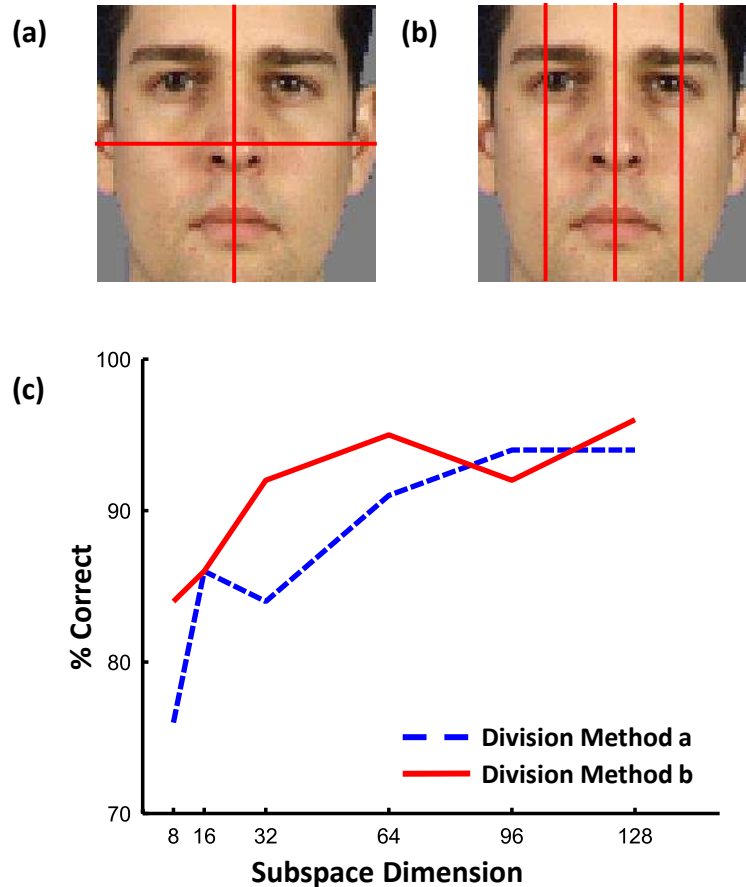


Figure 3.10: **Two patch division methods.** (a) The first method divides the signal and noise components into regular grids. (b) The second method still divides the signal component into a regular grid but divides the noise component into columns. (c) % Correct face identification for two region division methods in the XM2VTS Lighting database as a function of signal and noise subspace size.

3.4.4 Experiments for Expression and Illumination Variation Data Set (Yale)

In the Yale dataset, there are multiple gallery images per individual. There are two ways to proceed. We could treat the gallery images as a single individual with a single identity vector. For traditional distance-based algorithms this is equivalent to finding the centroid of the gallery images in feature space and matching to the nearest centroid. Hence, for compatibility with other work we refer to this as the nearest centroid (NC) method. Alternatively, we could treat each gallery image as a different individual with a different identity vector and consider it a success if we correctly match to any of these representations. We refer to this as the nearest neighbors (NN) method.

Table 3.3 shows % correct results from the Yale dataset using 8 gallery images for each individual as a function of the localization of the signal and noise basis functions using the nearest centroid method with subspace dimensions 14. The pattern of results is very similar as for the two XM2VTS datasets. Performance improves as the representation of the signal becomes more local, but declines as the noise becomes more local. The peak performance is again when the signal has 64 patches, but the noise has

only 4 patches and reaches a level of 93.8%.

Signal becomes more local \rightarrow

		P			
		1	4	16	64
Noise becomes more local \downarrow	Q				
	1	92.0%	91.2%	92.1%	93.1%
	4	91.7%	90.4%	91.7%	93.8%
	16	89.5%	89.3%	92.1%	92.2%
64	82.0%	84.0%	87.0%	86.2%	

Table 3.3: % Correct results for the Yale dataset as we vary patch resolution P and Q of signal and noise respectively when we use the nearest centroid method.

Figure 3.11 (a) and (b) shows the performance as a function of the number of gallery items for the nearest centroid and nearest neighbour metrics respectively. We also re-plot published results from [23]. In each case, the error bars represent the standard error of the results from the 7 training/test splits. We can draw two conclusions from this: first, our algorithm reliably outperforms the other methods. The only exception is for the nearest-neighbour PLDA method with a large number of gallery images per person. Second, for our algorithm the nearest centroid method consistently outperforms the nearest neighbour method. This is unsurprising as by combining information from gallery images it becomes possible to better distinguish signal and noise.

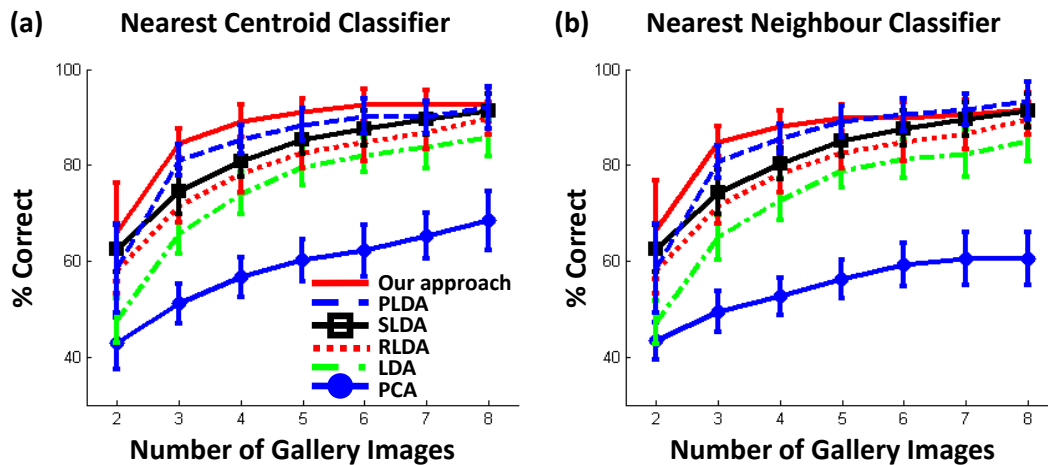


Figure 3.11: Plot of % correct identification performance for the Yale database with P=64 signal patches and Q=4 noise patches for (a) nearest centroid metric and (b) and nearest neighbour metrics. Results from PLDA [111], RLDA and SLDA [23], the Fisherfaces algorithm [10] and the Eigenfaces algorithm [132] are shown for comparison.

3.4.5 Experiments for Pose Variation Data Set (ORL)

Table 3.4 shows % correct results from the ORL dataset using 8 gallery images for each individual as a function of the localization of the signal and noise using the NC method with subspace dimension 39. Because there are 40 people in the ORL database, the maximum dimensions of the subspace can only be 39. As before, the performance decreases as the noise becomes more localized. However, making the signal more local has no net benefit here. When it becomes very local (64 patches) performance becomes worse. In fact the best performance is found when both the signal and noise are completely global (the original PLDA algorithm). An explanation of this effect can be found when we examine the images themselves. The faces in the ORL set contain considerable pose variation (see Figure 3.5d).

		Signal becomes more local →				
		P	1	4	16	64
↓ Noise becomes more local	Q					
	1		99.2%	99.1%	99.0%	98.6%
	4		98.0%	98.3%	98.8%	98.0%
	16		93.3%	95.0%	93.8%	88.4%
	64		81.3%	72.6%	74.0%	66.0%

Table 3.4: % Correct results for the ORL dataset as we vary patch resolution P and Q of the signal and noise respectively.

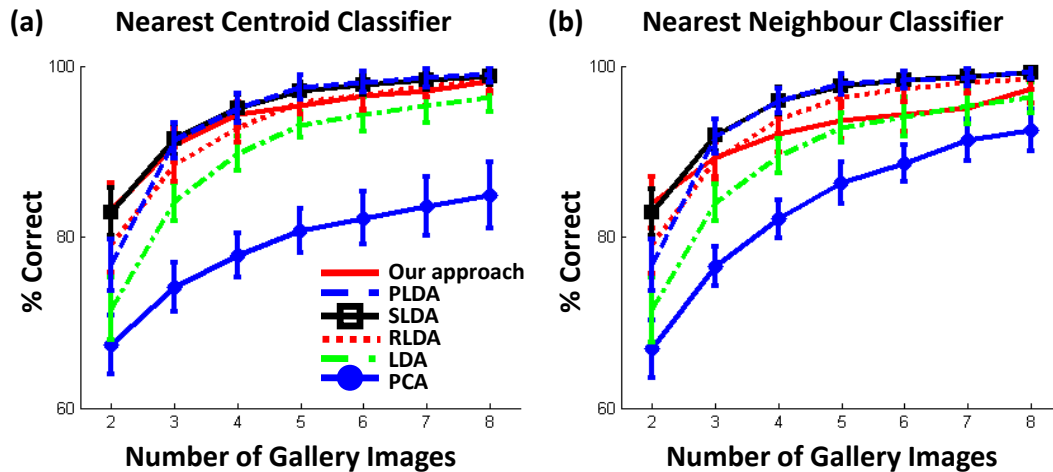


Figure 3.12: Plot of % correct identification performance for the ORL database with P=64 signal patches and Q=4 noise patches for nearest centroid metric. Results from PLDA [111], RLDA and SLDA [23], the Fisherfaces algorithm [10] and the Eigenfaces algorithm [132] are shown for comparison.

Hence, modeling the signal with very local basis functions becomes detrimental: the corresponding part of the face will not necessarily remain within the same patch.

Figure 3.12 (a) and (b) show performance for the NC and NN conditions respectively as a function of the number of gallery individuals. As for the Yale database, the NC metric outperforms the NN metric. However, we now find that our algorithm with local signal and global noise performs worse than either SLDA [23] or than fully global PLDA [111].

3.5 Experiments in the Unconstrained Database

3.5.1 Dataset

In this section we investigate the face verification performance of Multi-scale PLDA in the uncontrolled face database: Labeled Faces in the Wild (LFW) database [65]. The goal of face verification is to decide whether a pair of images are from the same person or not. As described in section 2.3.1, the LFW database is the most popular uncontrolled face database. It consists of 13233 images from 5749 individuals. All the images are captured from the internet. The number of images per person varies from 1 to 530. The images contain large variations in pose, illumination, expression, gender, age, etc. Figure 3.13(a) shows several examples from the LFW database.

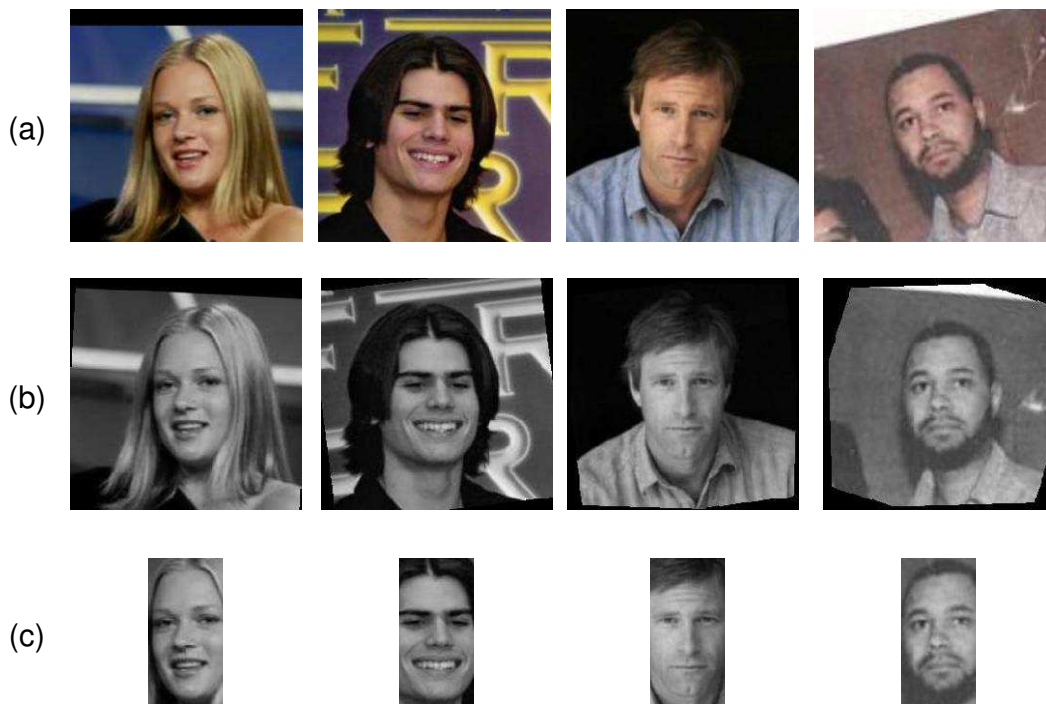


Figure 3.13: **Several examples from the Labeled Faces in the Wild (LFW) database [65].** (a) Color images with the size 250×250 pixels, which are collected from the internet and vary in pose, illumination, expression, gender, age, race, resolution, occlusion, background, and photo quality. (b) Aligned black-and-white images provided by [127]. (c) The 160×80 face regions are obtained by cropping the central part from the aligned images provided by [127].

In the LFW database, images are divided into 10 groups with mutually exclusive identities. In each group there are 300 matched pairs and 300 non-matched pairs. The verification protocol applies 10-fold

cross validation. In each repetition one group is used for testing and the other nine groups are used for training.

The LFW database designer defines two training configurations. In the ‘restricted configuration’ only same/not-same training labels can be used. The training examples are restricted to the given match and not match pairs. It is not allowed to use the names of people associated with images to generate additional training examples. In the ‘unrestricted configuration’ the identity information can be used. New training pairs may be created by leveraging the names of people.

As defined in [65], the verification results are reported by the estimated mean accuracy $\hat{\mu}$ and the standard error S_E of the mean :

$$\hat{\mu} = \frac{\sum_{i=1}^{10} p_i}{10} \quad (3.43)$$

$$S_E = \frac{\hat{\sigma}}{\sqrt{10}}, \quad (3.44)$$

where p_i is the percentage of correct assignment using group i for testing and $\hat{\sigma}$ is the estimate of the standard deviation given by

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{10} (p_i - \hat{\mu})^2}{9}}. \quad (3.45)$$

3.5.2 Experiments Using Image Intensities

The LFW images contains large variation. Face alignment can reduce scale and rotation variation effectively and increase verification performance. The authors of the LFW database provided images aligned by the congealing alignment method [64]. However, there is less misalignment in the aligned images provided by [127], which are obtained by applying a similarity transformation to register four fiducial points to a pre-defined template. Figure 3.13(b) shows several examples of the aligned images. In this section we adopt their aligned images to do experiments.

The size of the aligned LFW image is 250×250 pixels. We firstly crop the central 160×80 pixels from each image to obtain face region (Figure 3.13c). We use 6 iterations of the EM algorithm to train the model parameters θ , which are initialized to random values. The subspace dimension is set to 64 as in section 3.4.

Table 3.5 shows the mean % correct and the standard error of the mean of ten cross validation tests as a function of the localization of the signal and noise basis functions. The performance decreases when the signal becomes more local. There is also no performance increase when the noise becomes more local. The reason is the same as experiments using the ORL database. There is significant pose variation in the LFW database. The corresponding patches of the two images do not always include the same facial features because of pose changes.

To show the test results in the LFW database, I used three places of decimal to report both the mean and the standard error of the mean in Table 3.5. It was given by the following Matlab script:


```
meanCorrect = mean(resultList); % resultList is a 1-by-10 array, each element of 1
```

which represents the result of an experiment

```
stdCorrect = sqrt(sum((resultList - meanCorrect).^2)/9)/sqrt(10);
```

2

We used the default Matlab accuracy for reporting results. We considered three decimal places may be accurate enough to report the verification performance, so we will use the same style to report the test results in the LFW database in the following chapters.

Signal becomes more local 

Q \ P		Signal becomes more local		
		1	2	4
Noise becomes more local	1	59.733% ± 1.006	59.317% ± 0.993	58.417% ± 0.872
	2	58.000% ± 0.797	56.967% ± 0.924	55.817% ± 0.782
	4	55.817% ± 1.090	55.333% ± 0.992	54.050% ± 0.644

Table 3.5: **The mean accuracy and the standard error of the mean of ten LFW cross validation tests using image intensities as we vary patch resolution P and Q of signal and noise respectively.** The results show the performance decreases as the noise become more localized, but making the signal more local does not increase performance: we lose the correspondence of facial features in corresponding patches when pose variation exists.

3.6 Conclusion

In this chapter we combined patch-based face representation methods and Probabilistic Linear Discriminant Analysis (PLDA). We described a face image as a sum of the signal component and the noise component. We break both the signal and noise into regular grids of non-overlapping patches. We manipulate the patch configuration of the signal and noise to affect the spatial support of signal and noise basis functions. We investigated the effect of the degree of localization of these basis functions for frontal face recognition. We conjectured that performance would be best when the signal was treated locally (reflecting the fact that each point in the face provides independent information about identity) but that the noise was treated globally. This pattern of performance was confirmed for three controlled datasets, although in each case, the best performance was when the noise was at a large scale but not totally global. It appears that there is sufficient information in one quadrant of the image to capture the noise.

For a fourth controlled dataset, performance did not increase as the signal became more local. We attribute this difference to the pose changes that are present in this dataset but not in the three others. A local representation of identity fails if the images are not well registered as the same part of the face will not always appear in the same patch.

We also applied Multi-scale PLDA to an uncontrolled face database: the LFW database, whose images are collected from the internet instead of in the laboratory. The LFW images include large

variations. We used image intensities to do face verification. We found the best performance is obtained when we treat the signal globally. The reason is the same as for the fourth controlled database in which large pose variation exists.

One interesting aspect of this work is that the dimensions of the hidden space actually increase as we make the basis functions more local. Here, the hidden space refers to the between-individual subspace or within-individual subspace. This is because we maintain a fixed number of basis functions per patch. Although we marginalize over the hidden dimensions, at some level we still compare faces in a higher dimensional space than before. However, the number of non-zero parameters in the matrices $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ remains the same for a given subspace dimension, regardless of the scale of signal or noise. The model is not more complex, but makes different assumptions about independence of its parameters.

Independent manipulation of signal and noise subspaces was particularly easy to apply to the PLDA algorithm [111]. However, it could be adapted for any algorithm that calculates within-individual and between individual covariance matrices by assuming a block diagonal structure in these matrices.

This algorithm has connections with the Mosaicface model [3], in which each face image is approximated by a regular grid of patches and each patch is taken from a patch library. Faces are finally represented as a list of indices to the library. The common places between their model and Multi-Scale PLDA is that face images are represented as a set of non-overlapping patches. However, the latent variables of the two models are different. Multi-Scale PLDA applies continuous hidden variables while the Mosaicface model used discrete latent variables. The Mosaicface model improves performance when lighting variation exists. Performance might be improved if we combine the Mosaicface model and our Multi-Scale PLDA.

One of the drawbacks of Multi-Scale PLDA is that it is sensitive to pose variation. To address this problem we could estimate the corresponding patches for two images with different poses in future work. One possible solution is to extend the shiftmap representation [113] for patches and use it to find the corresponding patches containing the same facial features from two images.

Some aspects of our model remain unexplored. Our Multi-Scale model only used image intensities to represent images. We can extract image descriptors from each patch. Since image descriptors are generally more robust to image variation, performance might be improved if we use image descriptors instead of intensities.

In the following chapter, we will propose a new algorithm which will produce good performance when the pose of the image varies.

Chapter 4

Joint Probabilistic Linear Discriminant Analysis for Face Recognition

In the previous chapter we explored the combination of patch based face representation methods and Probabilistic Linear Discriminant Analysis (PLDA) [82]. In this chapter we compare PLDA and another Bayesian face recognition algorithm: the Joint Bayesian Face algorithm [30], which also produces good performance in the Labeled Faces in the Wild (LFW) database [65]. We analyze the commonalities and key differences between PLDA and the Joint Bayesian Face algorithm and propose Joint PLDA to combine the advantages of the two algorithms.

4.1 Introduction

The current state of the art algorithms in face recognition are to some extent dominated by a family of subspace algorithms. The Eigenfaces algorithm [132] was the first subspace algorithm and has become the most common performance benchmark. The principle behind the Eigenfaces algorithm is to apply principal components analysis (PCA) to project face images linearly to a low dimensional subspace. The goal of this projection is to maximize the scatter of all face images in the low dimensional subspace. The disadvantage of the Eigenfaces algorithm is that the maximized scatter is due not only to the between-individual scatter that is important for classification but also to the within-individual scatter that is not wanted. Therefore, unwanted variations due to pose, lighting, and expression are retained and the Eigenfaces algorithm is not an optimal algorithm from a discrimination viewpoint.

The Fisherfaces algorithm [10] overcame the drawback of the Eigenfaces algorithm. The Fisherfaces algorithm applies Linear discriminant analysis (LDA) to project face images to a low dimensional subspace by a set of projection vectors that maximize the ratio of the between-individual scatter matrix to the within-individual scatter matrix. The Fisherfaces algorithm improves the performance when lighting and expression variation exists. However, LDA often confronts the small sample problem, especially when dealing with high dimensional face image data. The small sample problem refers to the fact that the within-individual scatter matrix may become singular when the image number per individual is much smaller than the dimensions of the data.

To overcome the drawback of LDA, the Fisherfaces algorithm firstly uses PCA to reduce data

dimensionality and then performs LDA. However, this method has a drawback that the Fisherfaces algorithm is limited to the discriminant information in the principal subspace. Chen et al. [32] exploit the discriminant information that also exists in the null space spanned by the eigenvectors of the within-individual scatter matrix with zero eigenvalues. To use all the discriminative information, Wang et al. [137] proposed the Dual-Space LDA algorithm, which performs Linear Discriminant Analysis in both the principal and null subspace of within-individual scatter matrix. They demonstrated that the Dual-Space LDA algorithm produces better performance than the Fisherfaces algorithm.

All the aforementioned face recognition algorithms and their variations are distance-based algorithms. Face images are projected into a low dimensional subspace and the match assignment between two images is based on whether the distance of two images in the subspace is bigger than a threshold.

The Bayesian Face algorithm [97] uses a different method to verify whether two images match. It makes a match assignment by verifying whether the difference of a face image pair is caused mainly by between-individual variation or within-individual variation. The Bayesian Face algorithm uses a probabilistic framework to model between-individual and within-individual variation in training. In test, if the difference of two face images is mainly caused by between-individual variation, the two images have different identities. Conversely, if the image difference is mainly because of within-individual variation, two images are from the same person. The Bayesian Face algorithm demonstrated a performance advantage over the Fisherfaces algorithm in the FERET 1996 competition [97].

The aforementioned Bayesian Face algorithm and its variations generally model the image difference of a face image pair. Compared with modeling two images jointly, modeling the image difference can be understood as projecting a 2D space describing the relation of two images into a 1D space describing image difference. Such a projection can capture the major discriminative information but may reduce separability. Probabilistic Linear Discriminant Analysis (PLDA) [111] models two images jointly instead of the image difference and can capture more discriminative information. PLDA models the joint distribution of two images and makes verification assignment by comparing the match likelihood and the non-matching likelihood. In PLDA, each face image is considered to be generated from a hidden identity variable in the between-individual subspace and a hidden noise variable in the within-individual subspace pulsing some stochastic noise. In training, an EM algorithm is applied to estimate the model parameters: the basis functions for the between-individual subspace, the basis functions for the within-individual subspace and a diagonal matrix defining noise. In test, face verification is treated as a model selection problem. When two images are assumed to match, a match likelihood is computed by using the match covariance matrix derived from the learned model parameters. When two images are assumed to be in non-match model, a non-match likelihood is calculated by using the non-match covariance matrix derived from the trained model parameters. Two images are considered to match if the match likelihood is bigger than the non-match likelihood. Prince and Elder [111] demonstrated that their algorithm produces better performance than the Bayesian Face algorithm.

Chen et al. [30] claimed PLDA may discard some discriminative information because PLDA applies a subspace method to project high dimensional face data into a low dimensional subspace. To address

this issue, they proposed the Joint Bayesian Face algorithm which does not make the low dimension assumption and can estimate the match/non-match covariance matrix from high dimensional face data directly. They claimed their algorithm can capture more discriminative information and produced better performance than PLDA. In the Joint Bayesian Face algorithm each face is described to be the sum of two parts: identity and within-individual variation. In training, an EM-like algorithm is applied to learn the between-individual covariance matrix and the within-individual covariance matrix. In test, the match and non-match covariance matrix derived from the between-individual and within-individual covariance matrices are used to compute the match and non-match likelihoods for a given image pair. The match assignment is decided by comparing the two likelihoods.

The key difference between PLDA and the Joint Bayesian Face algorithm is that PLDA applies factor analysis to project face data into a low dimensional between-individual and within-individual subspaces to estimate the match and non-match covariance matrix while the Joint Bayesian Face algorithm uses an EM-like algorithm to partition each face image into an identity component and a within-individual component with the same dimensions as the original face data and then estimate the match and non-match covariance matrix directly from the identity components and within-individual components. Another difference is that PLDA uses a strict EM algorithm and guarantees the training log likelihood increases after each iteration while the Joint Bayesian Face algorithm uses an EM-like algorithm and cannot guarantee that the log likelihood converges.

Although Chen et al. [30] claimed their algorithm can capture more discriminative information and produce better performance than PLDA by using high dimensional face data instead of low dimensional vectors, the subspace method used by PLDA can improve signal-to-noise ratio and reduce the number of estimated entries when estimating the covariance matrix. Therefore, there is no obvious theoretical advantage for the Joint Bayesian Face algorithm. Although Chen et al. [30] demonstrated that the Joint Bayesian Face algorithm produced better performance than PLDA in the LFW database, the performance difference is marginal, only 0.8%. Moreover, the experiment settings in [30] may not be fair for PLDA. They chose the optimal parameters for the Joint Bayesian algorithm but did not use the optimal parameters for PLDA. Therefore, it is interesting to compare the Joint Bayesian Face algorithm and PLDA to find whether direct modeling or subspace method is better to estimate the match/non-match covariance matrix.

The structure of this chapter is as follows: we first introduce the detail of the Bayesian Face algorithm, PLDA, and the Joint Bayesian Face algorithm and analyze the commonalities and key differences of the three Bayesian face recognition algorithms in section 4.2. To show the difference between PLDA and the Joint Bayesian Face algorithm more clearly, we make an empirical comparison between the two algorithms in section 4.3. Then we propose Joint PLDA to combine the advantages of PLDA and the Joint Bayesian Face algorithm in section 4.4. After that we compare the performance of the three Bayesian face recognition algorithms using different image descriptors in section 4.5.2. We also use different approaches to combine multiple image descriptors in section 4.5.3. Finally, we draw a conclusion in section 4.6.

4.2 Bayesian Face Recognition Algorithms

In this section we will first give a brief introduction to three Bayesian face recognition algorithms: the Bayesian Face algorithm, PLDA and the Joint Bayesian Face algorithm. Then we will compare the three algorithms.

4.2.1 The Bayesian Face Algorithm

The Bayesian Face algorithm [97] models the image difference Δ of two images and makes the match assignment based on whether the image difference is mainly caused by between-individual or within-individual variation. The image difference is modeled by a probabilistic framework:

$$\Delta = \mathbf{x}_1 - \mathbf{x}_2 \quad (4.1)$$

$$P(\Delta|\mathcal{M}_s) = \mathcal{G}_\Delta[\mathbf{0}, \mathbf{s}] \quad (4.2)$$

$$P(\Delta|\mathcal{M}_d) = \mathcal{G}_\Delta[\mathbf{0}, \mathbf{d}], \quad (4.3)$$

where image \mathbf{x}_1 and \mathbf{x}_2 have been subtracted with the mean of all images; model \mathcal{M}_s denotes two images are from the same person and model \mathcal{M}_d denotes two images are from different people; the function $\mathcal{G}_o[\boldsymbol{\rho}, \boldsymbol{\varsigma}]$ denotes a Gaussian in o with mean $\boldsymbol{\rho}$ and covariance $\boldsymbol{\varsigma}$; the term $\boldsymbol{\Sigma}_s$ is the covariance matrix for within-individual variation and $\boldsymbol{\Sigma}_d$ is the covariance matrix for between-individual variation.

Learning

In training, model parameters $\boldsymbol{\theta} = \{\boldsymbol{\Lambda}_s, \mathbf{V}_s, \boldsymbol{\Sigma}_s, \boldsymbol{\Lambda}_d, \mathbf{V}_d, \boldsymbol{\Sigma}_d\}$ are learned from training images. Two sets of image pairs, which comprise intra-personal image pairs and extra-personal image pairs, are firstly collected from training images. Then the eigenvalues $\boldsymbol{\Lambda}_s$ and the eigenvectors \mathbf{V}_s of the within-individual covariance matrix $\boldsymbol{\Sigma}_s$ are learnt from intra-personal image pairs. The eigenvalues $\boldsymbol{\Lambda}_d$ and eigenvectors \mathbf{V}_d of the between-individual covariance matrix $\boldsymbol{\Sigma}_d$ are learnt from extra-personal image pairs.

Verification

The Bayesian Face algorithm makes match decision for two images by comparing the likelihood for within-individual variation $P(\Delta|\mathcal{M}_s)$ and the likelihood for between-individual variation $P(\Delta|\mathcal{M}_d)$. To compute two likelihoods more efficiently, each test image \mathbf{x}_k is firstly preprocessed with whitening transformation and then is stored as two vectors: the between-individual subspace coefficients \mathbf{h}_k and the within-individual subspace coefficients \mathbf{w}_k and :

$$\mathbf{h}_k = \boldsymbol{\Lambda}_d^{-1/2} \mathbf{V}_d \mathbf{x}_k \quad (4.4)$$

$$\mathbf{w}_k = \boldsymbol{\Lambda}_s^{-1/2} \mathbf{V}_s \mathbf{x}_k. \quad (4.5)$$

Whitening transformation is a decortication transformation, which can transfer a set of random variables having a known covariance matrix into a set of new random variables having a identity covariance matrix. A typical whitening process to a random vector \mathbf{X} with a not singular covariance matrix $\boldsymbol{\Sigma}$ means \mathbf{X} multiplying by $\boldsymbol{\Sigma}^{-1/2}$. Then the match likelihood $P(\Delta|\mathcal{M}_s)$ and the non-match likelihood $P(\Delta|\mathcal{M}_d)$

are computed by:

$$\Delta = \mathbf{x}_1 - \mathbf{x}_2 \quad (4.6)$$

$$P(\Delta|\mathcal{M}_s) = \frac{e^{-1/2\|\mathbf{w}_1 - \mathbf{w}_2\|^2}}{(2\pi)^{D/2}|\Sigma_s|^{1/2}} \quad (4.7)$$

$$P(\Delta|\mathcal{M}_d) = \frac{e^{-1/2\|\mathbf{h}_1 - \mathbf{h}_2\|^2}}{(2\pi)^{D/2}|\Sigma_d|^{1/2}}, \quad (4.8)$$

where D is subspace dimension.

4.2.2 PLDA

PLDA models two images jointly instead of the image difference. In PLDA, a face image is represented by:

$$\mathbf{x}_{ij} = \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij} \quad (4.9)$$

where \mathbf{x}_{ij} denotes the j^{th} image of the i^{th} individual which has subtracted the mean of all face images, the matrix \mathbf{F} consists of the basis functions for the between-individual subspace in columns and \mathbf{h}_i denotes the hidden identity variable that is constant for all J images $\mathbf{x}_{i1\dots iJ}$ of the person i . The matrix \mathbf{G} contains the basis functions for the within-individual subspace in columns. The term \mathbf{w}_{ij} denotes the hidden noise variable that is different for each image. The term ϵ_{ij} represents a stochastic noise. The identity information is represented by $\mathbf{F}\mathbf{h}_i$, which accounts for between-individual variation. For a given individual, the term $\mathbf{F}\mathbf{h}_i$ is constant. Within-individual variation is represented by $\mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}$, which explains why two images of the same individual do not look identical.

We can alternately describe the image generation in terms of conditional probabilities:

$$Pr(\mathbf{x}_{ij}|\mathbf{h}_i, \mathbf{w}_{ij}) = \mathcal{G}_{\mathbf{x}}[\mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij}, \Sigma] \quad (4.10)$$

$$Pr(\mathbf{h}_i) = \mathcal{G}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}] \quad (4.11)$$

$$Pr(\mathbf{w}_{ij}) = \mathcal{G}_{\mathbf{w}}[\mathbf{0}, \mathbf{I}]. \quad (4.12)$$

where the term Σ is a covariance matrix and \mathbf{I} is a identity matrix.

Learning

In training, an EM algorithm is applied to learn the parameters $\theta = \{\mathbf{F}, \mathbf{G}, \Sigma\}$. In the Expectation- or E-Step, we fix the parameters θ and compute a full posterior distribution over the hidden variables \mathbf{h}_i and \mathbf{w}_{ij} . In the Maximization- or M-Step, we optimize the estimates of the parameters θ . The EM algorithm guarantees the likelihood increases at each training iteration.

Verification

In PLDA the match assignment for two images is decided by comparing the non-match likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)$ and the match likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s)$, where model \mathcal{M}_d denotes that two images do not match and model \mathcal{M}_s denotes that two images match.

When two images are assumed to be from different people (model \mathcal{M}_d) and two images are assumed to be generated independently, the non-match likelihood of two images is as

$$Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d) = Pr(\mathbf{x}_1|\mathcal{M}_d)Pr(\mathbf{x}_2|\mathcal{M}_d). \quad (4.13)$$

Here we need to compute the term $Pr(\mathbf{x}_1|\mathcal{M}_d)$ and $Pr(\mathbf{x}_2|\mathcal{M}_d)$. According to the equation 4.9, the generation of image \mathbf{x}_1 can be written as

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{F} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \mathbf{w} \end{bmatrix} + \epsilon \quad (4.14)$$

or

$$\mathbf{x}_1 = \mathbf{A}\mathbf{y} + \epsilon. \quad (4.15)$$

According to the equations 4.10, 4.11, and 4.12, the generation of image \mathbf{x}_1 can be described in terms of conditional probabilities:

$$\begin{aligned} Pr(\mathbf{x}_1) &= Pr(\mathbf{x}_1|\mathbf{y})Pr(\mathbf{y}) \\ &= \mathcal{G}_{\mathbf{x}_1}[\mathbf{A}\mathbf{y}, \Sigma']\mathcal{G}_{\mathbf{y}}[\mathbf{0}, \mathbf{I}] \\ &= \mathcal{G}_{\mathbf{x}_1}[\mathbf{0}, \mathbf{A}\mathbf{A}^T + \Sigma] \\ &= \mathcal{G}_{\mathbf{x}_1}[\mathbf{0}, \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \Sigma], \end{aligned} \quad (4.16)$$

where

$$\Sigma' = \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Sigma \end{bmatrix}.$$

The generation of image \mathbf{x}_2 can be described in the similar format as image \mathbf{x}_1 , so the equation 4.13 can be written as

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d) &= Pr(\mathbf{x}_1|\mathcal{M}_d)Pr(\mathbf{x}_2|\mathcal{M}_d) \\ &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_d] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \mathbf{A}\mathbf{A}^T + \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{A}^T + \Sigma \end{bmatrix} \right] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \Sigma \end{bmatrix} \right], \end{aligned} \quad (4.17)$$

where the term \mathbf{x} is the concatenation of image \mathbf{x}_1 and \mathbf{x}_2 , the term Σ_d is the non-match covariance matrix.

When two images are assumed to be from the same person (Model \mathcal{M}_s), according to the equation 4.9, the generation of image \mathbf{x}_1 and \mathbf{x}_2 can be described as:

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{h} \\ \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} \quad (4.18)$$

or

$$\mathbf{x} = \mathbf{B}\mathbf{z} + \epsilon'. \quad (4.19)$$

According to the equation 4.10, 4.11, and 4.12, the match likelihood of two images \mathbf{x}_1 and \mathbf{x}_2 can be

written as:

$$\begin{aligned}
 Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s) &= Pr(\mathbf{x} | \mathbf{z}) Pr(\mathbf{z}) \\
 &= \mathcal{G}_{\mathbf{x}}[\mathbf{Bz}, \Sigma'] \mathcal{G}_{\mathbf{z}}[\mathbf{0}, \mathbf{I}] \\
 &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_s] \\
 &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \mathbf{B}\mathbf{B}^T + \Sigma'] \\
 &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \Sigma & \mathbf{F}\mathbf{F}^T \\ \mathbf{F}\mathbf{F}^T & \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \Sigma \end{bmatrix} \right], \quad (4.20)
 \end{aligned}$$

where the term Σ_s is the match covariance matrix.

With the two above likelihoods, we make match decision by the log likelihood ratio $r(\mathbf{x}_1, \mathbf{x}_2)$:

$$\begin{aligned}
 r(\mathbf{x}_1, \mathbf{x}_2) &= \log \frac{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s)}{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d)} \\
 &= \log Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s) - \log Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d) \\
 &= \frac{2 \log(2\pi) - \log |\Sigma_s| - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x} + 2 \log(2\pi) + \log |\Sigma_d| + \mathbf{x}^T \Sigma_d^{-1} \mathbf{x}}{2} \\
 &\propto \kappa + \mathbf{x}^T \Sigma_d^{-1} \mathbf{x} - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x}, \quad (4.21)
 \end{aligned}$$

where κ is a constant.

4.2.3 The Joint Bayesian Face Algorithm

The Joint Bayesian Face algorithm modes two images jointly but does not make low dimensional assumption as PLDA. In the Joint Bayesian Face algorithm a face image \mathbf{x}_{ij} is represented as the sum of the identity component α_i and the within-individual variation component β_{ij} :

$$\mathbf{x}_{ij} = \alpha_i + \beta_{ij}, \quad (4.22)$$

where the term \mathbf{x}_{ij} is the j^{th} image of the i^{th} person. Both the identity component α_i and the within-individual variation component β_{ij} follow Gaussian distributions:

$$\alpha_i = \mathcal{G}_{\alpha}[\mathbf{0}, \Sigma_{\alpha}] \quad (4.23)$$

$$\beta_{ij} = \mathcal{G}_{\beta}[\mathbf{0}, \Sigma_{\beta}], \quad (4.24)$$

where the term Σ_{α} is the covariance matrix for the identity component; the term Σ_{β} is the covariance matrix for the within-individual variation component.

Learning

In training, an EM-like algorithm is applied to learn the covariance matrices Σ_{α} and Σ_{β} from a set of training images. In the E-Step of the EM-like algorithm, the covariance matrices Σ_{α} and Σ_{β} are fixed to estimate the identity component α_i and the within-individual variation component β_{ij} for each image \mathbf{x}_{ij} . In the M-Step, the covariance matrices Σ_{α} and Σ_{β} are updated. This training method is not a strict EM algorithm and this training method cannot guarantee that the likelihood increases at each iteration.

Verification

Similar to PLDA, the Joint Bayesian Face algorithm makes match assignment for two images based

on the log likelihood ratio $r(\mathbf{x}_1, \mathbf{x}_2)$ between the match likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s)$ and the non-match likelihood $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)$:

$$\begin{aligned} r(\mathbf{x}_1, \mathbf{x}_2) &= \log \frac{Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s)}{Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)} \\ &\propto \mathbf{x}^T \Sigma_d^{-1} \mathbf{x} - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x}, \end{aligned} \quad (4.25)$$

where the match likelihood is obtained by

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s) &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_s] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \Sigma_\alpha + \Sigma_\beta & \Sigma_\alpha \\ \Sigma_\alpha & \Sigma_\alpha + \Sigma_\beta \end{bmatrix} \right], \end{aligned} \quad (4.26)$$

and the non-match likelihood is obtained by

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d) &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_d] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \Sigma_\alpha + \Sigma_\beta & \mathbf{0} \\ \mathbf{0} & \Sigma_\alpha + \Sigma_\beta \end{bmatrix} \right]. \end{aligned} \quad (4.27)$$

4.2.4 Discussion

The Bayesian Face algorithm, the Joint Bayesian Face algorithm, and PLDA all belong to a family of Bayesian face recognition algorithms and have the following features in common:

- All the three algorithms use a probabilistic framework.
- All the three algorithms consider two types of image variation: between-individual and within-individual variation.
- All the three algorithms are based on a comparison of two Gaussians for recognition, although the mean and variance of these Gaussians varies from algorithm to algorithm.

The difference among the three Bayesian face recognition algorithms is summarized in Table 4.1.

Category Algorithm	Modelling Target	Training Method	Verification Method
Bayesian Face	Probability of image difference	PCA subspace method	Comparing between-individual variation and within-individual variation
PLDA	Joint probability of two images	EM algorithm	Comparing the match and non-match log likelihood
Joint Bayesian Face	Joint probability of two images	EM-like algorithm	Comparing the match and non-match log likelihood

Table 4.1: Comparison of the three Bayesian face recognition algorithms.

From Table 4.1 we see that the difference between the Bayesian Face algorithm and the two other algorithms is that the Bayesian Face algorithm models the image difference while the Joint Bayesian Face algorithm and PLDA model the joint probability of two images. Compared with modeling two images jointly, modeling image difference might reduce separability.

The key difference between the Joint Bayesian Face algorithm and PLDA is that the Joint Bayesian Face algorithm uses an EM-like algorithm to estimate the covariance matrices directly from high dimensional data while PLDA applies an EM algorithm to approximate covariance matrices by a factor analysis subspace method. The advantage of the EM-like training method of the Joint Bayesian Face algorithm is that it can estimate the covariance matrix without projecting data into a low dimensional subspace and the disadvantage is that likelihood convergence cannot be guaranteed in theory. Conversely, PLDA applies a strict EM training algorithm and guarantees the likelihood increases at each iteration. However, PLDA uses factor analysis subspace method and makes the low dimension assumption, so it might discard some discriminatory information.

Although the training method of the Joint Bayesian Face algorithm and PLDA is different, the verification equations of the two algorithms are very similar in the test phase. To show this more clearly, we rewrite the verification equations of two algorithms for the matched model \mathcal{M}_s and the unmatched model \mathcal{M}_d .

When image \mathbf{x}_1 and \mathbf{x}_2 are assumed from the same person (Model \mathcal{M}_s), the match likelihood for both the two algorithms can be derived as

$$Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s) = \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \boldsymbol{\Sigma}_s],$$

where the covariance matrix $\boldsymbol{\Sigma}_s^J$ for the Joint Bayesian Face algorithm is defined in equation 4.28, the covariance matrix $\boldsymbol{\Sigma}_s^P$ for PLDA is defined in equation 4.29:

$$\boldsymbol{\Sigma}_s^J = \begin{bmatrix} \boldsymbol{\Sigma}_\alpha + \boldsymbol{\Sigma}_\beta & \boldsymbol{\Sigma}_\alpha \\ \boldsymbol{\Sigma}_\alpha & \boldsymbol{\Sigma}_\alpha + \boldsymbol{\Sigma}_\beta \end{bmatrix} \quad (4.28)$$

$$\boldsymbol{\Sigma}_s^P = \begin{bmatrix} \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \boldsymbol{\Sigma} & \mathbf{F}\mathbf{F}^T \\ \mathbf{F}\mathbf{F}^T & \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \boldsymbol{\Sigma} \end{bmatrix}. \quad (4.29)$$

When two images are assumed to be from different people (Model \mathcal{M}_d) and are generated independently, the non-match likelihood for both the two algorithms can be written as

$$Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d) = \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \boldsymbol{\Sigma}_d],$$

where the covariance matrix $\boldsymbol{\Sigma}_d^J$ of the Joint Bayesian Face algorithm is defined in equation 4.30, the covariance matrix $\boldsymbol{\Sigma}_d^P$ of PLDA is defined in equation 4.31:

$$\boldsymbol{\Sigma}_d^J = \begin{bmatrix} \boldsymbol{\Sigma}_\alpha + \boldsymbol{\Sigma}_\beta & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_\alpha + \boldsymbol{\Sigma}_\beta \end{bmatrix} \quad (4.30)$$

$$\boldsymbol{\Sigma}_d^P = \begin{bmatrix} \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}\mathbf{F}^T + \mathbf{G}\mathbf{G}^T + \boldsymbol{\Sigma} \end{bmatrix}. \quad (4.31)$$

4.3 Empirical Comparison of the Joint Bayesian Face algorithm and PLDA

In the previous section we argued that the Joint Bayesian Face algorithm used an EM-like training method and could not guarantee the likelihood increased at each iteration in training. We also claimed that PLDA used an EM training method and the likelihood increasing at each iteration was guaranteed. In this section, to validate our argument, we will show the likelihood using the model parameters of the Joint Bayesian Face algorithm and PLDA obtained at each training iteration. We also investigate the verification performance using the model parameters of the two algorithms obtained at each training iteration.

To perform the aforementioned experiments, we use the aligned LFW images provided by [127]. We preprocess each image as follows. We crop the central 160×80 pixels from each LFW image to obtain the face region. Then we extract Local Binary Patterns (LBP) descriptors [102] from each image by the following settings: we divide each face image into several 12×12 non-overlapping patches, we set the radius to form neighborhood over each pixel location to 3, we set the number of neighbor points to 8, and we use uniform binary patterns. We compute LBP histograms of each 12×12 patch and normalize the histograms in each patch to unit length, then truncate the histograms at 0.2 and normalize again to unit length. In the end each image is described by a LBP vector with 7552 dimensions.

We adopt the ‘unrestricted configuration’ of the LFW training data, which means identity labels associated with images are allowed to be used. We apply PCA to reduce the dimensions of the data to 100, 200, and 400 for both the Joint Bayesian Face algorithm and PLDA. We always set the subspace dimensions of PLDA to 128.

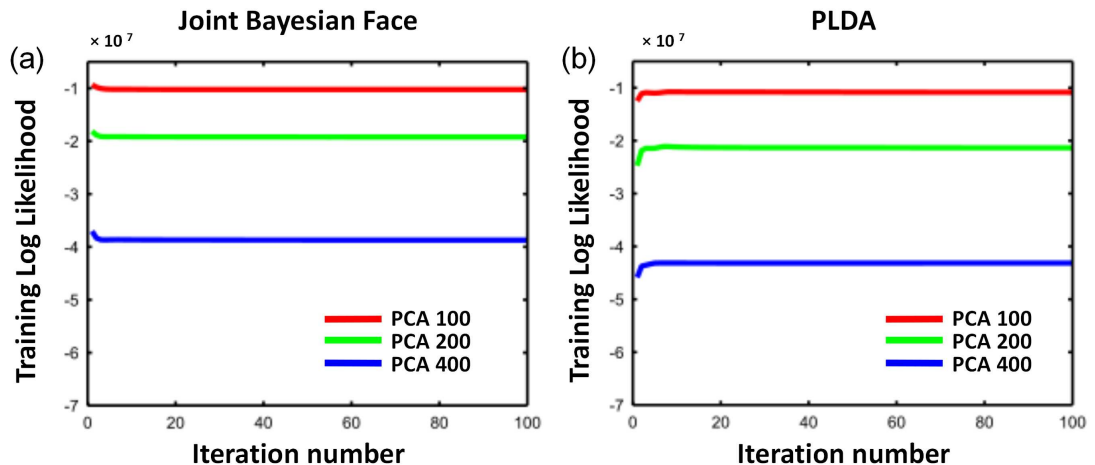


Figure 4.1: We compare the training likelihood of the Joint Bayesian Face algorithm and PLDA at each iteration when the PCA dimensions are set to 100, 200, 400. (a) The total Log likelihood of the Joint Bayesian Face algorithm over ten LFW cross-validation experiments as a function of iteration number when the PCA dimensions are set to 100, 200, 400. (b) The total Log likelihood of PLDA as a function of iteration number for the three PCA dimensions.

In the LFW database images are divided into ten non-overlapping sets and the verification performance is reported by 10 cross-validation experiments. In each experiment, one image set is used for testing and the nine other sets are used for training. In each experiment, for the Joint Bayesian Face algorithm, we initialize the model parameters to random values and calculate the log likelihood of all training images using the estimated model parameters at each iteration. We add up the 10 likelihoods of the 10 experiments at each iteration and show the sum of the log likelihoods as a function of iteration number in Figure 4.1 (a). We perform the experiments when the PCA dimensions are set to 100, 200, 400.

Similarly for PLDA, we initialize the model parameters to random values and compute the log likelihood of all training images using the model parameters obtained at each iteration in each experiment. We show the sum of the log likelihood of 10 experiments as a function of iteration number in Figure 4.1 (b). We perform the experiments when we set the PCA dimensions to 100, 200, 400.

From Figure 4.1 (a) we find that the total log likelihood of the Joint Bayesian Face algorithm first decreases and then gradually saturates for all the three PCA dimensions. However, the total log likelihood of PLDA as shown in Figure 4.1 (b) maintains a increasing trend as the iteration number increases and this pattern is revealed for all the three PCA dimensions. Therefore, it is clear that the EM-like training method of the Joint Bayesian Face algorithm cannot guarantee that the likelihood increases at each iteration while the EM training method of PLDA can.

We also investigate the verification performance of two algorithms using the model parameters obtained at each training iteration. For the two algorithms, in each experiment of 10 LFW cross-validation experiments, we use the obtained model parameters at each iteration to compute the % correct for the test set. We compute the mean % correct of 10 experiments at each iteration and show the mean % correct of the Joint Bayesian Face algorithm as a function of iteration number in Figure 4.2(a). We perform the experiments when the PCA dimensions are set to 100, 200, 400. Similarly, we show the mean % correct of PLDA as a function of iteration number for the three PCA dimensions in Figure 4.2(b).

From Figure 4.2 (a) we find that the mean % correct of the Joint Bayesian Face increases in the first 3 iterations and then decreases till closing to a stable value for the three PCA dimensions. Conversely, as shown in Figure 4.2 (b), the verification performance of PLDA maintains a increasing trend as the iteration number increases for the three PCA dimensions.

4.4 Joint PLDA

In this section we propose Joint PLDA to combine the advantages of the Joint Bayesian Face algorithm and PLDA.

4.4.1 Motivation

The Joint Bayesian Face algorithm and PLDA have their own advantages and disadvantages. The disadvantage of the Joint Bayesian Face algorithm is that its training method is only an EM-like method and cannot guarantee that the likelihood increases at each iteration. However, the Joint Bayesian Face algorithm does not make the low dimension assumption and may capture more discriminatory informa-

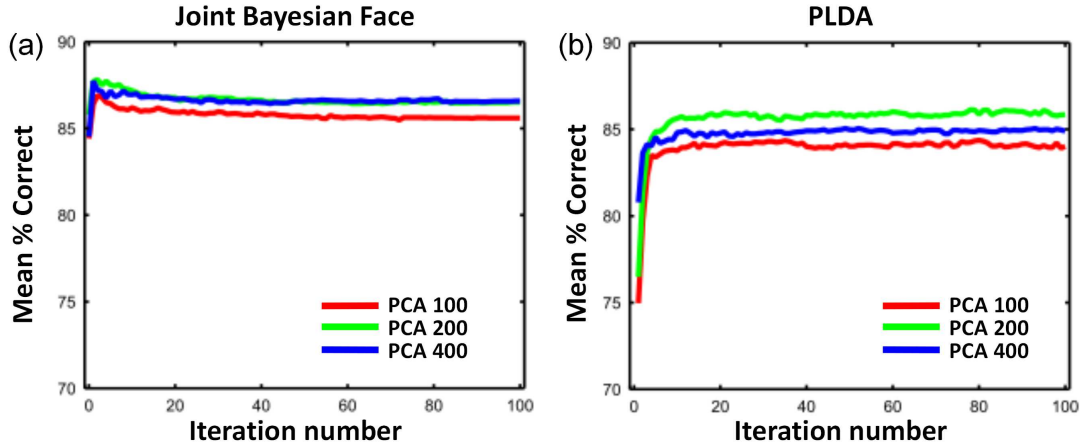


Figure 4.2: We compare the verification performance of the Joint Bayesian Face algorithm and PLDA at each iteration when the PCA dimensions are set to 100, 200, 400. (a) Mean % correct of the Joint Bayesian Face algorithm over the 10 LFW cross-validation experiments as a function of iteration number when the PCA dimensions are set to 100, 200, 400 respectively. (b) Mean % correct of PLDA as a function of iteration number for the three PCA dimensions.

tion by estimating the covariance matrix from high dimensional data directly. The advantage of PLDA is that its training algorithm is an EM method and the training likelihood keeps increasing as the training iteration number increases. To combine the advantages of the two algorithms, we propose a new algorithm: Joint PLDA. We apply the EM algorithm of PLDA to learn the model parameters of PLDA. Then we use the E-Step of the EM algorithm of PLDA to divide each image into the identity component and the within-individual variation component. We compute a covariance matrix for the identity component and another covariance matrix for the within-individual variation component. Lastly we derive the match and non-match covariance matrix to do verification as in the Joint Bayesian Face algorithm. By this approach we can guarantee that the likelihood increases in training and can also estimate the within-individual covariance matrix from high dimensional data.

4.4.2 Face Image Representation

In the Joint PLDA algorithm, a face image \mathbf{x}_{ij} is represented as the sum of the identity component α_i and the within-individual variation component β_{ij} :

$$\mathbf{x}_{ij} = \alpha_i + \beta_{ij} \quad (4.32)$$

$$\alpha_i = \mathbf{F}\mathbf{h}_i \quad (4.33)$$

$$\beta_{ij} = \mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}, \quad (4.34)$$

where image \mathbf{x}_{ij} has subtracted the mean of all images. The identity component α_i is equivalent to the term $\mathbf{F}\mathbf{h}_i$ of PLDA and the within-individual variation component β_{ij} is equivalent to the term $\mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}$ of PLDA.

Both the two components α_i and β_{ij} follow Gaussian distribution:

$$\alpha_i = \mathcal{G}_\alpha[\mathbf{0}, \Sigma_\alpha] \quad (4.35)$$

$$\beta_{ij} = \mathcal{G}_\beta[\mathbf{0}, \Sigma_\beta], \quad (4.36)$$

where the term Σ_α is the covariance matrix for the identity component; the term Σ_β is the covariance matrix for the within-individual variation component.

4.4.3 Learning

In training we aim to learn the covariance matrix Σ_α and Σ_β . We firstly use the EM algorithm of PLDA to estimate the model parameters $\theta = \{\mathbf{F}, \mathbf{G}, \Sigma\}$ from training images. Generally the iteration number of the EM training algorithm is set to 25. Then we use the model parameters $\hat{\theta}$ estimated at the end of the EM iterations and apply the E-Step of the EM algorithm of PLDA (defined in [111]) to compute the expectation of hidden identity variable \mathbf{h}_i and hidden noise variable \mathbf{w}_{ij} for each training image \mathbf{x}_{ij} :

$$E[\mathbf{y}_{ij}] = (\mathbf{A}^T \Sigma'^{-1} \mathbf{A} + \mathbf{I})^{-1} \mathbf{A}^T \Sigma'^{-1} \mathbf{x}_{ij}, \quad (4.37)$$

where

$$\mathbf{y}_{ij} = \begin{bmatrix} \mathbf{h}_i \\ \mathbf{w}_{ij} \end{bmatrix} \quad (4.38)$$

$$\mathbf{A} = [\mathbf{F} \quad \mathbf{G}] \quad (4.39)$$

$$\Sigma' = \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Sigma \end{bmatrix}. \quad (4.40)$$

After we obtain the estimated hidden variables \mathbf{h}_i and \mathbf{w}_{ij} , we can compute the identity component α_i and the within-individual variation component β_{ij} for each image \mathbf{x}_{ij} by

$$\alpha_i = \mathbf{F} \mathbf{h}_i \quad (4.41)$$

$$\beta_{ij} = \mathbf{x}_{ij} - \mathbf{F} \mathbf{h}_i. \quad (4.42)$$

Lastly, we calculate the covariance matrix Σ_α for the identity component and the covariance matrix Σ_β for the within-individual variation component by

$$\begin{aligned} \Sigma_\alpha &= \text{cov}(\alpha) \\ \Sigma_\beta &= \text{cov}(\beta), \end{aligned} \quad (4.43)$$

where the term α denotes the estimated identity components of all training images and the term β denotes the within-individual variation components of all training images.

4.4.4 Inference

Similar to the Joint Bayesian Face algorithm, the match decision for two images is made based on the log likelihood ratio:

$$\begin{aligned} r(\mathbf{x}_1, \mathbf{x}_2) &= \log \frac{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s)}{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d)} \\ &\propto \mathbf{x}^T \Sigma_d^{-1} \mathbf{x} - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x}, \end{aligned}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \quad (4.44)$$

$$\Sigma_d = \begin{bmatrix} \Sigma_\alpha + \Sigma_\beta & \mathbf{0} \\ \mathbf{0} & \Sigma_\alpha + \Sigma_\beta \end{bmatrix} \quad (4.45)$$

$$\Sigma_s = \begin{bmatrix} \Sigma_\alpha + \Sigma_\beta & \Sigma_\alpha \\ \Sigma_\alpha & \Sigma_\alpha + \Sigma_\beta \end{bmatrix}. \quad (4.46)$$

4.5 Experiments in the Unconstrained Database

In this section, we compare the verification performance of four Bayesian face recognition algorithms: the Bayesian Face algorithm, PLDA, the Joint Bayesian Face algorithm and Joint PLDA in the LFW database [65]. We will first introduce the preprocessing method for the LFW images in section 4.5.1. Then we will show the performance of four Bayesian face recognition algorithms using different image descriptors in section 4.5.2. Finally we will compare three combination approaches to combine multiple image descriptors in section 4.5.3.

4.5.1 Preprocessing

As introduced in section 3.5, the Labeled Faces in the Wild (LFW) dataset [65] has become a benchmark database to evaluate face recognition in uncontrolled environments. In this section we still adopt the ‘unrestricted configuration’, which means identity information can be used in training. We report face verification results by the mean % correct of 10 cross validation experiments and the standard error of the mean.

As in section 3.5 we used the aligned images provided by [127]. We crop the central 160×80 pixels from each aligned 250×250 black-and-white image to obtain face regions. The purpose of only preserving the face region is to reduce image variation from image background. The images used in the following experiments are black-and-white face regions with the size 160×80 pixels. Example images are shown in Figure 3.13(c).

In this chapter we always use 25 iterations of an EM algorithm as [82] to train the model parameters of PLDA and Joint PLDA, which are initialized to random values. As shown in section 4.3 that the best performance of Joint Bayesian Face algorithm is achieved when a small iteration number between 3 and 6 is chosen, so we always apply 5 iterations of an EM-like algorithm to train the model parameters of the Joint Bayesian Face algorithm.

4.5.2 Experiments Using Image Descriptors

In this section we apply Local Binary Pattern (LBP) descriptors [102], Three-Patch LBP (TPLBP) descriptors [140], Four-Patch LBP (FPLBP) descriptors [140], Scale Invariant Feature Transform (SIFT) descriptors [88], Histogram of oriented gradients (HOG) descriptors [38] to do face verification in the LFW database.

We firstly apply LBP descriptors to compare performance of four Bayesian face recognition algorithms: the Bayesian Face algorithm, PLDA, the Joint Bayesian Face algorithm and Joint PLDA. The

LBP descriptors are calculated at each pixel location. The simplest local binary pattern thresholds a 3×3 neighborhood over a pixel location by comparing with the intensity of the central pixel. Then the subsequent pattern of 8 bits, which is the comparison results, is treated as a binary number. The histogram of these binary numbers in a predefined region is then used as a character to describe the region. Normally, uniform binary patterns are used. Uniform binary patterns mean there are maximum 2 transitions from 0 to 1. For example, 11100011 is a uniform binary pattern and 01101101 is not. The non-uniform LBPs are considered to be equivalent and assigned into one histogram bin when the histogram of all uniform LBPs is computed. The LBP representation for the whole face image is to divide the image into a grid of regions and then compute the LBP histograms in each region. The concatenation of all the LBP histograms forms the LBP face image descriptor.

		Subspace Dims				
		32	64	96	128	160
Algorithms	PCA Dims					
	PLDA	100	80.150 ± 0.719	81.367 ± 0.659	69.700 ± 0.597	59.900 ± 1.054
200		80.467 ± 0.754	81.617 ± 0.728	81.450 ± 0.640	83.350 ± 0.800	82.000 ± 0.788
400		80.183 ± 0.747	81.350 ± 0.603	81.583 ± 0.612	82.217 ± 0.812	82.650 ± 0.618
600		79.833 ± 0.740	80.650 ± 0.540	81.000 ± 0.704	80.950 ± 0.637	80.417 ± 0.802
Joint PLDA	100	81.800 ± 0.652	81.917 ± 0.633	75.950 ± 0.578	67.500 ± 0.692	68.850 ± 1.103
	200	83.467 ± 0.885	83.883 ± 0.843	84.317 ± 0.836	83.917 ± 0.739	83.433 ± 0.722
	400	83.633 ± 0.643	83.783 ± 0.588	83.600 ± 0.769	83.417 ± 0.637	83.500 ± 0.716
	600	82.717 ± 0.595	82.150 ± 0.629	81.900 ± 0.684	81.683 ± 0.663	81.117 ± 0.792

Table 4.2: **The verification performance of PLDA and Joint PLDA in the LFW database using the LBP image descriptors provided by [30] as we vary the PCA dimensions and subspace dimensions.** The performance is shown by the mean % correct and the standard error of the mean based on 10 cross-validation experiments. Numbers with the red color indicate the best performance for fixed PCA dimension. We find the optimal PCA dimension and subspace dimension for PLDA are 200 and 128 respectively; the optimal PCA dimension and subspace dimension for Joint PLDA is 200 and 96 respectively.

We adopt the LBP descriptors provided by [30] to do the following experiments. The dimen-

sion of their LBP descriptors for a image is 5900. We apply PCA to reduce the dimensions. We need to find the optimal PCA dimensions for the Bayesian Face algorithm and the Joint Bayesian Face algorithm. There are two sets of parameters for PLDA and Joint PLDA: (i) the reduced PCA dimensions and (ii) the number of basis functions for signal and noise component. The latter two parameters are always varied together in our experiments and are referred to as “subspace dimensions”.

We apply an empirical approach to obtain the optimal values for PCA dimension and subspace dimensions. Table 4.2 shows performance of PLDA and Joint PLDA under different combinations of PCA dimensions and subspace dimensions. The results are reported by the mean % accuracy and the standard error of the mean based on 10 cross-validation experiments. From the table we find the optimal PCA dimension and subspace dimensions for PLDA are 200 and 128 respectively; the optimal PCA dimension and subspace dimensions for Joint PLDA are 200 and 96 respectively.

We list performance of four Bayesian face recognition algorithms in Table 4.3 as we vary PCA dimensions. From the table we find that the optimal PCA dimensions for the Bayesian Face algorithm and the Joint Bayesian Face algorithm are 100 and 400 respectively. When all the algorithms apply the optimal parameters, Joint PLDA performs best among four algorithms, the Joint Bayesian Face algorithm produces slightly better performance than PLDA, and the Bayesian Face algorithm performs worst.

Algorithms \ PCA Dims	PCA Dims			
	100	200	400	600
Bayesian Face	76.950 ± 0.486	72.250 ± 0.431	72.483 ± 0.654	65.467 ± 0.911
Joint Bayesian Face	81.967 ± 0.583	84.017 ± 0.725	84.067 ± 0.637	80.017 ± 0.672
PLDA	81.367 ± 0.659	83.350 ± 0.800	82.650 ± 0.618	81.000 ± 0.704
Joint PLDA	81.917 ± 0.633	84.317 ± 0.836	83.783 ± 0.588	82.717 ± 0.595

Table 4.3: **The performance of four Bayesian face recognition algorithms in the LFW database using the LBP image descriptors provided by [30] as we vary PCA dimensions.** The performance is shown by the mean % correct and the standard error of the mean based on 10 cross-validation experiments. For PLDA and Joint PLDA, the optimal subspace dimensions have been applied. Numbers with red colors indicate the best performance of the algorithm. Joint PLDA produces the best performance.

We also extracted our own LBP image descriptors and apply them to do face verification in the LFW database. To extract LBP descriptors, we divide a face image into several non-overlapping regions. We vary the size of regions to extract different LBP descriptors. We term the LBP descriptors with the extraction regions of the size 8×8 pixels, 10×10 pixels, 12×12 pixels, 14×14 pixels as LBP8,

LBP10, LBP12, LBP14. Other parameters to extract LBP descriptors are as follows: the radius to form neighborhood over a pixel location is set to 3, the number of neighbor points is set to 8. Uniform binary patterns are applied. We normalize the histograms in each region to unit length, then truncate their value at 0.2, then normalize again to unit length.

The verification results of four Bayesian face algorithms using LBP8, LBP10, LBP12, LBP14 are shown in Table 4.4. We set the PCA dimensions to 400 for the Bayesian Face algorithm and the Joint Bayesian Face algorithm. We set the PCA dimensions and subspace dimensions to 200 and 128 respectively for PLDA and Joint PLDA. From the table we find that Joint PLDA always performs best among four Bayesian face recognition algorithms for all different LBP descriptors. The best performance of Joint PLDA is obtained using the LBP12 descriptors. Compared with the LBP descriptors provided by [30], our LBP descriptors produce significantly better performance. The reason might be that we normalized the LBP histograms.

Algorithms Descriptors	Bayesian Face	PLDA	Joint Bayesian	Joint PLDA
LBP Provided by [30]	76.950 ± 0.486	83.350 ± 0.800	84.067 ± 0.637	84.317 ± 0.836
LBP8 [102]	78.050 ± 0.617	85.117 ± 0.502	85.950 ± 0.488	86.183 ± 0.434
LBP10 [102]	82.283 ± 0.608	87.333 ± 0.394	88.217 ± 0.343	88.267 ± 0.402
LBP12 [102]	82.067 ± 0.526	87.600 ± 0.451	87.617 ± 0.512	88.000 ± 0.442
LBP14 [102]	81.150 ± 0.432	86.600 ± 0.468	87.550 ± 0.428	87.733 ± 0.393
SIFT Provided by [58]	80.717 ± 0.554	86.317 ± 0.416	86.600 ± 0.590	87.333 ± 0.453
HOG [38]	78.717 ± 0.661	84.283 ± 0.491	84.217 ± 0.467	85.067 ± 0.472
TPLBP [140]	76.550 ± 0.520	82.933 ± 0.339	83.850 ± 0.423	83.933 ± 0.447
FPLBP [140]	75.500 ± 0.626	81.317 ± 0.637	82.033 ± 0.567	82.333 ± 0.619

Table 4.4: **The verification performance of four Bayesian face recognition algorithms using different image descriptors.** Numbers with red colors indicate the best performance. Joint PLDA performs best for all the descriptors. The best performance is achieved by using LBP descriptors. The LBP8, LBP10, LBP12, and LBP14 descriptors mean that we divide a face image into several regions with the size 8×8 pixels, 10×10 pixels, 12×12 pixels and 14×14 pixels respectively to extract LBP descriptors.

We also used the SIFT descriptors provided by [58] to represent face images to compare the performance of four Bayesian face recognition algorithms. They detected 9 fiducial points of each image and extracted SIFT descriptors from each fiducial point. A face image is represented by a concatenated vector of the SIFT descriptors from the 9 points. We set the PCA dimensions to 400 for the Bayesian Face algorithm and the Joint Bayesian Face algorithm. We set the PCA dimensions and subspace dimensions to 200 and 128 respectively for PLDA and Joint PLDA. The verification results of four Bayesian face recognition algorithms using the SIFT descriptor are also shown in Table 4.4. From the table we find that the Joint PLDA algorithm performs best among four Bayesian face recognition algorithms when SIFT descriptors are used to represent images. Compared with the performance using the LBP descriptors extracted by us, the performance using the SIFT descriptor is slightly worse.

We also used HOG descriptors [39] to represent images to compare the performance of four Bayesian face recognition algorithms. We use the following settings to extract the HOG descriptors from images: the cell size is set to 10×10 pixels, there are 2×2 cells in a block, the overlap rate among blocks is set to 0.5, the angle range is set to $0^\circ \sim 180^\circ$, and the bin number is set to 9. We set the PCA dimensions to 400 for the Bayesian Face algorithm and the Joint Bayesian Face algorithm. We set the PCA dimensions and subspace dimensions to 200 and 128 respectively for PLDA and Joint PLDA. The verification results of four Bayesian face algorithms using HOG descriptors are also shown in Table 4.4. From the table we find that when HOG descriptors are used to represent images the Joint PLDA algorithm performs best. We also find that the performance using HOG descriptors is worse than the performance using the SIFT and our LBP descriptors.

The TPLBP and FPLBP descriptors are also used to represent face images to compare the performance of four algorithms. We follow the settings in [140] to extract TPLBP and FPLBP descriptors. For both the two descriptors, we set the PCA dimensions to 400 for the Bayesian Face algorithm and the Joint Bayesian Face algorithm; we set the PCA dimensions and subspace dimensions to 200 and 128 respectively for PLDA and Joint PLDA. The verification results of four algorithms using TPLBP and FPLBP descriptors are also shown in Table 4.4. From the table we find that Joint PLDA algorithm still performs best when images are represented by TPLBP and FPLBP descriptors. We also find that the performance using TPLBP descriptors is worse than the performance using FPLBP descriptors. Moreover, the performance using the two descriptors is much worse than the performance using other descriptors.

Among the four Bayesian face recognition algorithms, the Bayesian Face algorithm always performs worst for all the image descriptors. The results suggest that modeling the probability of image difference captures less discriminatory information than modeling the joint probability of two images. From Table 4.4, we find that the performance of the Joint Bayesian Face algorithm is slightly better than PLDA for all the image descriptors although the difference is quite marginal. It might demonstrate that estimating covariance matrices directly from high dimensional data has a weaker advantage than using subspace method. Joint PLDA always performs best, which supports our argument that Joint PLDA can improve the verification performance by combining the advantages of PLDA and the Joint Bayesian Face algorithm.

Among all the image descriptors in Table 4.4, the best performance is obtained using LBP descriptors. It shows LBP descriptors might capture more discriminatory information than other image descriptors. The performance varies when we extract LBP descriptors from regions with different size, which indicates the size of region affects the verification performance and we need to find a optimal value.

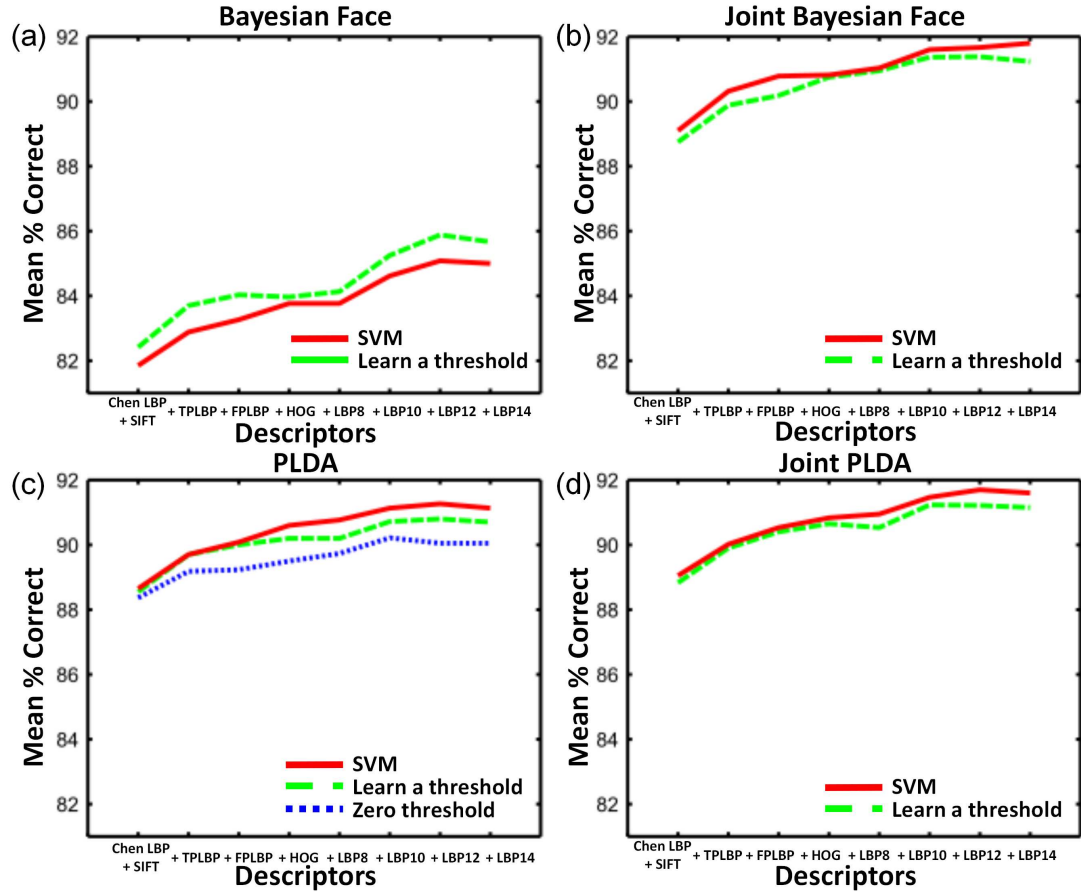


Figure 4.3: **Comparison of three combination approaches.** (a) The SVM combination approach is worse than the ‘learn-a-threshold’ combination method for the Bayesian Face algorithm. However, the SVM combination approach performs best for the Joint Bayesian Face algorithm in (b), PLDA in (c) and Joint PLDA in (d).

4.5.3 Experiments Combining Multiple Image Descriptors

As Wolf et al. [140] and Li et al. [82] demonstrated, combining multiple image descriptors produces better performance than using single descriptor. In this section, we firstly compare three approaches to combine multiple descriptors and then show the performance of four Bayesian face recognition algorithms combining multiple descriptors.

Our methods to combine multiple descriptors are to combine match scores of LFW test images pairs. We use the match scores obtained in section 4.5.2. The match score of the Bayesian Face Recognition algorithm for a pair of images is the difference between the match log likelihood and the non-match

likelihood. For the Bayesian Face algorithm, the match likelihood is computed by the equation 4.7 and the non-match likelihood is calculated by the equation 4.8. For PLDA, the match likelihood is computed by the equation 4.20 and the non-match likelihood is calculated by the equation 4.17. For the Joint Bayesian Face algorithm, the match likelihood is computed by the equation 4.26 and the non-match likelihood is calculated by the equation 4.27. For Joint PLDA, the match likelihood and the non-match likelihood are computed by the same functions as the Joint Bayesian Face algorithm. We use the match scores of different image descriptors obtained in section 4.5.2 to compare the performance of the following three combination approaches:

- We treat each image descriptor independently and the final match score for a image pair is the sum of the match scores using each descriptor. Two images are considered to match if the final match score is bigger than zero.
- We treat each image descriptor independently and the final match score is the sum of the match scores using each descriptor. Two images are considered to match if the final match score is bigger than a threshold, which we learn from training images.
- We create a $n \times d$ match score matrix from n training image pairs and d image descriptors. We train a Linear SVM classifier [36] based on the score matrix. We use the trained SVM classifier to predict two images matching or not matching.

The first combination approach is adopted by PLDA in [82] but is not suitable for the Bayesian Face algorithm, the Joint Bayesian Face algorithm, and Joint PLDA because a match threshold has to be learnt for the three algorithms. The second and third approaches can be applied to all the four algorithms.

In Figure 4.3 we compare the performance of different combination approaches for the Bayesian Face algorithm in (a), the Joint Bayesian Face algorithm in (b), PLDA in (c) and Joint PLDA in (d). From Figure 4.3 we find that the performance of the three combination approaches increases when more descriptors are combined. The results confirm the conclusion of [140] and [82]. We also find that the SVM combination approach performs best except for the Bayesian Face algorithm.

We use the SVM combination approach to combine multiple descriptors and compare the performance of four algorithms in Figure 4.4. We find that the performance of Joint Bayesian Face algorithm and Joint PLDA is very close to each other. The Bayesian Face algorithm performs worst.

Chen et al. [30] claimed that the Joint Bayesian Face algorithm performed better than PLDA in the LFW database when combining LBP, SIFT, TPLBP, and FPLBP descriptors. We use the LBP descriptors provided by [30], the SIFT descriptors provided by [58], and our own implementation of TPLBP, and FPLBP descriptors to duplicate their experiments. We use the SVM approach to combine the four descriptors. Our experiment results in Table 4.5 agree with their conclusion that PLDA performs slightly worse than the Joint Bayesian Face algorithm when the four descriptors are combined. Joint PLDA performs better than PLDA but worse than the Joint Bayesian Face algorithm.

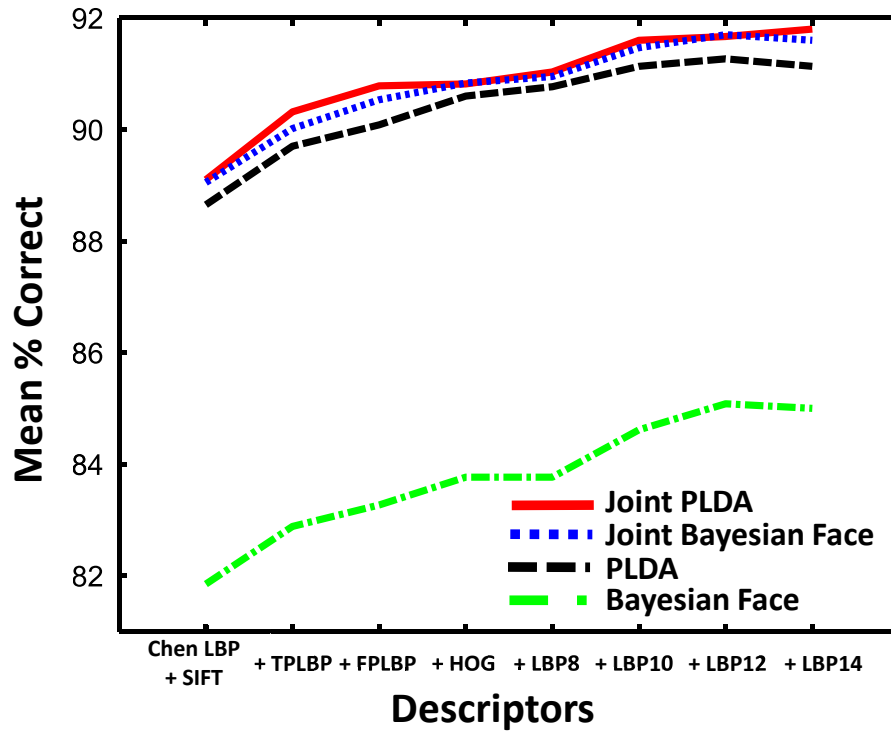


Figure 4.4: We compare the performance of four Bayesian face recognition algorithms when the SVM combination approach is used to combine multiple descriptors. The performance of Joint PLDA is slightly better than the Joint Bayesian Face algorithm.

Results \ Algorithms	Bayesian Face	PLDA	Joint Bayesian Face	Joint PLDA
Results provided by [30]		90.07	90.90	
Our experiment results	83.267 ± 0.413	90.080 ± 0.365	90.783 ± 0.360	90.583 ± 0.335

Table 4.5: We duplicate the experiments in [30] using the SVM approach to combine LBP, SIFT, TPLBP and FPLBP. Our results agree with the conclusion in [30] that PLDA performs slightly worse than the Joint Bayesian Face algorithm.

We notice that our LBP descriptors perform better than the LBP descriptors provided by [30] in section 4.5.2. Therefore, we are motivated to compare the performance of four Bayesian face recognition algorithms using our own LBP descriptors. As shown in Table 4.6, when we combine LBP(LBP8, LBP10, LBP12, LBP14), SIFT, TPLBP and FPLBP descriptors, Joint PLDA produces the best performance 91.367 ± 0.448 , which is nearly the same as the performance 91.30 ± 0.003 of the commercial face recognition application face.com [63] at that time.

Results \ Algorithms	Bayesian Face	PLDA	Joint Bayesian Face	Joint PLDA
LBP (LBP8, LBP10, LBP12, LBP14), SIFT, TPLBP, FPLBP	84.455 ± 0.575	90.717 ± 0.522	91.217 ± 0.466	91.367 ± 0.448

Table 4.6: **We duplicate the experiments in [30] using our own LBP image descriptors.** Joint PLDA performs best.

4.6 Conclusion

In this chapter we compared the Bayesian Face algorithm, PLDA, and the Joint Bayesian Face algorithm by analyzing their commonalities and differences. We find that modeling two images jointly can capture more discriminatory information than modeling the image difference. PLDA and the Joint Bayesian Face algorithm model the joint distribution of two images and produce good performance in the LFW database. PLDA and the Bayesian Face algorithm have their own advantages but they also have some disadvantages. We are motivated to propose Joint PLDA to combine the advantages of PLDA and the Bayesian Face algorithm. Joint PLDA applies a strict EM algorithm to guarantee likelihood increases and can also estimate the covariance matrix from the high dimensional data directly.

We compared the performance of the Bayesian Face algorithm, the Joint Bayesian Face algorithm, PLDA and Joint PLDA in the LFW database. Our experiments demonstrate that Joint PLDA performs best when a single descriptor is used. We also compare the performance of the four Bayesian face recognition algorithms when we combine multiple image descriptors. When we combine LBP, SIFT, TPLBP and FPLBP descriptors, Joint PLDA can achieve $91.367\% \pm 0.448$ correct in the LFW database, which is comparable to $91.300\% \pm 0.003$ correct of the commercial face recognition system face.com [126].

Joint PLDA has connections to metric learning algorithms [58] [42], which aim to learn a metric to make two classes separable. Metric learning algorithms generally learn a Mahalanobis distance to separate two classes:

$$(\mathbf{x}_1 - \mathbf{x}_2)^T \Psi (\mathbf{x}_1 - \mathbf{x}_2), \quad (4.47)$$

where Ψ is a positive definite matrix.

If we compare equation 4.44 and equation 4.47, we find both the metric learning algorithms and Joint PLDA learn metrics. However, metric learning algorithms model the image difference and hence have the same drawback as the Bayesian Face algorithm. Modeling image differences might reduce the separability and capture less discriminatory information than modeling two images jointly.

In this chapter we only explore the performance of the four Bayesian face recognition algorithms using global image descriptors, which means that we extract visual features from the whole image. In future work we hope to investigate the performance of the four algorithms using local descriptors, which means we extract visual features from fiducial points (for example, the corners of the eyebrows, the

corners of the eyes). As demonstrated in [31], local descriptors are generally more robust to image variation if fiducial points can be detected precisely. Therefore, using local descriptors to represent face images might improve performance.

In the following chapter, we will argue large pose variation is the challenge for the Bayesian face recognition algorithms of this chapter and propose new algorithms to overcome the challenge.

Chapter 5

Tied Bayesian Face Recognition Algorithms for Pose Variation

5.1 Introduction

After decades of research to automatic face recognition, many face recognition algorithms and benchmarks have been generated. However, face recognition in uncontrolled environments is still an unsolved problem. In the previous chapter we have shown that Bayesian face recognition algorithms can produce good performance in the evaluation benchmark for face recognition under uncontrolled environments: the LFW database. In this chapter we will argue that large pose changes are the challenge for improving the verification performance in the LFW database and propose new algorithms to overcome this challenge.

How to deal with large pose changes has been a popular research topic for many years. Among existing face recognition algorithms across pose, Tied PLDA [82] produces good performance and is computationally fast. Tied PLDA is a version of PLDA [111]. In this context, a ‘Tied’ means that images from the same person but with two different poses have a common hidden variable but different generation processes. Tied PLDA estimates the mapping between two poses and performs well in the controlled XM2VTS [95] and FERET database [106] when large pose variation exists. We hypothesize that Tied PLDA can also deal with large pose variation in uncontrolled databases such as the LFW database [65].

It is also interesting to investigate a tied version of the Joint Bayesian Face algorithm [30] and Joint PLDA. We propose the Tied Joint Bayesian Face algorithm and Tied Joint PLDA. Throughout this chapter we refer to PLDA, the Joint Bayesian Face algorithm, and Joint PLDA as Bayesian face recognition algorithms. We refer to tied PLDA, the Tied Joint Bayesian Face algorithm, and Tied Joint PLDA as Tied Bayesian face recognition algorithms. We will demonstrate that Tied Bayesian face recognition algorithms have an advantage in dealing with large pose variation.

Tied Bayesian face recognition algorithms assign images into pre-defined horizontal pose categories and model the relationship of images under different pose categories. Therefore, to use tied models, sufficient training images are required in each pose category. However, the images in the LFW database were collected using the Viola-Jones frontal face detector [134], so there are few images in the non-

frontal pose categories. To address this issue, our first solution is to use the Multi-PIE database to train Tied Bayesian face recognition algorithms. This database contains 755,370 images from 345 individuals with 6 expressions and 19 lighting conditions and 15 poses under four sessions. Therefore, the Multi-PIE database can provide sufficient training images to train tied models. However, one possible disadvantage of this approach is that images in the Multi-PIE database were collected in the laboratory. These controlled Multi-PIE images might not include an equivalent amount of image variation as for the test images from the uncontrolled LFW database. Therefore, our second solution to address the problem of insufficient training images is to collect a new database from the internet but make sure there are sufficient training images in each pose category. We called the new database the UCL Multi-Pose.

The structure of this chapter is as follows: In section 5.2, we analyse the verification results of three Bayesian face recognition algorithms in the LFW database to find the challenge of improving performance. We then review existing face recognition algorithms across pose in section 5.3. In section 5.4 we propose two new Tied algorithms: the Tied Joint Bayesian Face algorithm and Tied Joint PLDA. In section 5.5, we introduce the UCL Multi-Pose database. To compare the performance of three Tied Bayesian face recognition algorithms to deal with pose variation in a controlled face database, we train and test the three algorithms in the Multi-PIE database in section 5.6.2. To investigate the performance difference of three algorithms to handle pose variation in an uncontrolled database, a similar comparison will also be applied using the UCL Multi-Pose database in section 5.6.3. To compare the verification performance of three Tied algorithms in the LFW database, we train the three Tied models in the Multi-PIE database and test in the LFW database in section 5.6.4. To verify whether an uncontrolled training database improves the performance of three tied models in the LFW database, we apply cross-database experiments by training in the UCL Multi-Pose database and testing in the LFW database in section 5.6.5. To obtain the best recognition performance in the LFW database, we add a switching mechanism to switch recognition algorithms based on the poses of two images in section 5.6.6. Finally, we draw a conclusion in section 5.7.

5.2 Performance of three Bayesian Face Recognition Algorithms under pose variation

In this section we will analyse the ability of three Bayesian face recognition algorithms to deal with pose variation in the LFW database. Since there are few LFW images with vertical pose variation, we restrict our consideration to pose variation to horizontal pose variation in this chapter.

We manually assign each image of the LFW database to a pre-defined pose category $\{-60^\circ, -45^\circ, -30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$, where the negative poses denote images that are left facing and the positive poses denote images that are right facing. To reduce the number of pose categories, we swap all left facing images to right facing images and change the pose value from a negative number to a positive number. After that each LFW image is with a pose from the set $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$. Although flipping all the left facing images might decrease the accuracy, the image variation caused by flipping is relatively minor if we consider all the images are captured under completely uncontrolled

environments and images include large variation.

Instead of using image intensities, we use the same method described in section 4.5.2 to extract LBP descriptors to represent images. We use the aligned LFW images provided by [127]. We first crop the central 160×80 pixels from each LFW image to obtain the face region; then we extract LBP12 image descriptors (see the detail of this descriptor in section 4.5.2), which are obtained by dividing each image into a grid of 12×12 non-overlapping regions and concatenating all the LBP histograms from each region. The histograms in each region are normalised to unit length, then are truncated at 0.2, and then are normalised again to unit length. In the end each image is described by a LBP vector with 7552 dimensions.

All three Bayesian face recognition algorithm apply PCA to reduce the dimensionality of the data vector. As in section 4.5.2 the PCA dimensions for the Joint Bayesian Face algorithm are set to 400. The PCA dimensions and subspace dimensions for both PLDA and Joint PLDA are set to 200 and 128 respectively. We adopt the ‘unrestricted configuration’ to use the LFW training data, which means identity labels associated with images are allowed to be used. For the Joint Bayesian Face algorithm we apply 6 iterations of an EM-like algorithm to train its model parameters, which are initialized to random values. For PLDA and Joint PLDA, we apply 25 iterations of an EM algorithm to train their model parameters, which are initialized to random values.



Figure 5.1: We assign each LFW test pair into one of a pre-defined pair groups based on the poses of the two images. We show several examples of matched pairs and non-matched pairs in pair group $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$, $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$.

To compare different algorithms properly, the LFW database designers established an evaluation

protocol. Images are divided into 10 subsets which are mutually exclusive in terms of identities and images. The experiments are performed 10 times by applying a leave-one-out validation scheme. In each experiment, one subset is selected for testing and the remainder of the 9 subsets are used for training. In each test set there are 300 matched pairs and 300 non-matched pairs. Based on the poses of the two images in a pair, we assign each pair into one of the pair groups, which comprise $\{0^\circ-0^\circ\}$, $\{15^\circ-15^\circ\}$, $\{30^\circ-30^\circ\}$, $\{45^\circ-45^\circ\}$, $\{0^\circ-15^\circ\}$, $\{15^\circ-30^\circ\}$, $\{30^\circ-45^\circ\}$, $\{0^\circ-30^\circ\}$, $\{15^\circ-45^\circ\}$, $\{30^\circ-60^\circ\}$, $\{0^\circ-45^\circ\}$, $\{15^\circ-60^\circ\}$, $\{0^\circ-60^\circ\}$. Since there are not pairs belonging to $\{45^\circ-60^\circ\}$ and $\{60^\circ-60^\circ\}$, we do not list these two groups. Figure 5.1 shows examples of matched pairs and non-matched pairs in pair group $\{0^\circ-15^\circ\}$, $\{0^\circ-30^\circ\}$, $\{0^\circ-45^\circ\}$, $\{0^\circ-60^\circ\}$. For each pair group we collect the relevant image pairs in each test set and compute the % correct verification decisions. We repeat the experiments in the ten test sets and report the final verification performance by the mean of 10 experiment results and the standard error of the mean. Table 5.1 shows the performance of three Bayesian face recognition algorithms for each pair group.

Pose Difference	Pair Groups	PLDA	Joint Bayesian Face	Joint PLDA
0	0-0	88.676±0.866	88.732±0.791	89.242±0.639
	15-15	88.471±1.116	88.486±1.158	88.724±1.457
	30-30	85.819±1.689	88.002±1.335	87.472±1.418
	45-45	30.000±15.275	30.000±15.275	30.000±15.275
15	0-15	90.029±0.873	89.231±0.782	89.492±0.664
	15-30	87.048±1.223	87.362±1.202	87.656±1.084
	30-45	70.433±5.233	73.167±6.093	76.944±4.672
30	0-30	84.256±1.103	86.487±0.887	86.629±0.793
	15-45	82.662±2.985	77.765±3.000	81.523±3.614
	30-60	51.667±15.000	56.167±15.176	66.667±14.907
45	0-45	76.916±6.943	75.566±5.804	75.197±5.552
	15-60	38.333±14.490	38.333±14.498	38.333±14.498
60	0-60	51.071±12.644	61.071±12.099	61.071±12.099

Table 5.1: **Performance of PLDA, the Joint Bayesian Face algorithm and Joint PLDA for different pair groups in the LFW database.** All three Bayesian face recognition algorithms produce good performance for near frontal pair groups, in which both two images are with pose 0° , 15° , 30° . However, they all perform badly when either image of a pair is with pose 45° , 60° . Note there are no pairs belonging to $\{45^\circ-60^\circ\}$ and $\{60^\circ-60^\circ\}$ group, so we do not show them in the table.

From Table 5.1 we find that the three Bayesian face recognition algorithms perform well when the two images are near frontal, which means images are with pose 0° , 15° , 30° . For example, Joint

PLDA achieved 89.242%, 89.492%, 86.629% correct for pair group $\{0^\circ - 0^\circ\}$, $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$ respectively. However, the performance drops significantly if either of two images is with pose $45^\circ, 60^\circ$. For example, Joint PLDA only achieved 75.197%, 61.071% correct for pair group $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$ respectively. Therefore, we can draw a conclusion that large pose variation is the challenge for improving performance in the LFW database. For pair groups with large pose difference, we notice that the performance of three Bayesian face recognition algorithms for pair group $\{0^\circ - 60^\circ\}$ are better than for pair group $\{15^\circ - 60^\circ\}$. The reason is because that there are less training images for pair group $\{15^\circ - 60^\circ\}$ than for pair group $\{0^\circ - 60^\circ\}$ as shown in table 5.2.

5.3 Existing Face Recognition Algorithms Across Pose

5.3.1 Previous Work

Existing face recognition algorithms across pose can be classified into two categories: 3D algorithms and 2D algorithms. Since human heads are 3D objects and pose variation is essentially caused by the motion of head in a 3D space, many 3D algorithms are motivated to be proposed to handle pose variation. The key of the 3D algorithms is 3D models, which might be a single model [50] or a deformable model in the format of parameters [17]. Existing 3D algorithms can be divided into three categories according to the way that the 3D model is used:

- Frontalization. Probe images, which are normally non-frontal, are transformed into frontal view. Gallery images are normally frontal. Then a match is decided between a frontalized probe image and a frontal gallery image. A example is [8].
- Synthesis. A 3D model is applied to generate several virtual images at several poses based on the frontal gallery image. Then the generated gallery image and the probe image with the same pose is compared to make a match decision. A example is [151].
- 3D model parameters. All the gallery and probe images are fitted into a 3D model to obtain a set of model parameters as a unique signature for each image. Then model parameters of the gallery and the probe image are compared to decide whether two images match. A example is [17].

As demonstrated in [8] [151] [17], 3D algorithms typically require several minutes to recognize an image and the recognition performance depends heavily on the precision of the 3D models and the optimization algorithms.

Compared with the 3D algorithms, the 2D algorithms lack one degree of freedom. However, the 2D algorithms can apply statistical learning method to estimate the relationship of images at different poses. The 3D transformation caused by pose difference can be approximated by some statistical learning strategies. The learning process to pose transformation can be conducted in image space or feature space. Examples in image space include Active appearance models [34], Linear Shape model [70], Eigen Light Fields [54], etc. Examples in feature space include Kernel PCA [86], Kernel FDA [144], Correlation Filters [80], Local Linear Regression [27], etc.

Compared with the 3D algorithms, the 2D algorithms have advantages in speed and simplicity of implementation. However, the recognition performance of these algorithms has historically not been as good as 3D algorithms. Recently Arashloo et al. [6] proposed an MRF-based classification method, which used the energy of the established match between a pair of images to decide the match assignment. They measured textural and structural similarities between two images. The main advantages of their algorithm is that it does not need to know the poses of probe images and does not need use non-frontal images in training. Their algorithm produced good performance when pose variation exists. However, the computation cost to recognize an image is expensive. Prince et al. [111] proposed a variant of PLDA called Tied PLDA, which demonstrated better performance than the 3D algorithms on several constrained databases. Their experiments showed that Tied PLDA can handle large pose variation well. I will describe the detail of the Tied PLDA in the next section.

5.3.2 Tied PLDA

In Tied PLDA face images are considered as generated from two underlying variables: the hidden identity variable and the hidden noise variable. The hidden identity variable describes identity and is constant for a given identity. The hidden noise variable explains within-individual variation of images at the same pose. Images from the same person at different poses are considered to be generated from the same hidden identity variable but using different pose-dependent linear transformations. The image generation process is described by the following equation:

$$\mathbf{x}_{ijk} = \boldsymbol{\mu}_k + \mathbf{F}_k \mathbf{h}_i + \mathbf{G}_k \mathbf{w}_{ijk} + \boldsymbol{\epsilon}_{ijk}, \quad (5.1)$$

where \mathbf{x}_{ijk} denotes the k^{th} pose of the j^{th} image of the i^{th} individual, $\boldsymbol{\mu}_k$ represents the mean image at pose k , \mathbf{F}_k is a matrix containing the between-individual basis functions in columns for pose k . The term \mathbf{h}_i represents the hidden identity variable which is constant for all the images of the i^{th} individual. The matrix \mathbf{G}_k is a matrix containing the within-individual basis functions in columns for pose k . The term \mathbf{w}_{ijk} denotes the hidden noise variable which is different for each image. The term $\boldsymbol{\epsilon}_{ijk}$ represents a stochastic noise.

More formally, the generative process can be described in terms of conditional probabilities:

$$Pr(\mathbf{x}_{ijk} | \mathbf{h}_i, \mathbf{w}_{ijk}) = \mathcal{G}_{\mathbf{x}}[\boldsymbol{\mu}_k + \mathbf{F}_k \mathbf{h}_i + \mathbf{G}_k \mathbf{w}_{ijk}, \boldsymbol{\Sigma}_k] \quad (5.2)$$

$$Pr(\mathbf{h}_i) = \mathcal{G}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}] \quad (5.3)$$

$$Pr(\mathbf{w}_{ijk}) = \mathcal{G}_{\mathbf{w}}[\mathbf{0}, \mathbf{I}], \quad (5.4)$$

where $\mathcal{G}_{\mathbf{o}}[\boldsymbol{\rho}, \boldsymbol{\varsigma}]$ denotes a Gaussian in \mathbf{o} with mean $\boldsymbol{\rho}$ and covariance $\boldsymbol{\varsigma}$.

Learning

Given training images \mathbf{x} with different poses, an EM algorithm is applied to learn the parameters $\boldsymbol{\theta} = \{\mathbf{F}_k, \mathbf{G}_k, \boldsymbol{\Sigma}_k\}$ for each pose. In the Expectation Step, we fix the parameters $\boldsymbol{\theta}$ and compute the full posterior distribution over the latent variables \mathbf{h}_i and \mathbf{w}_{ijk} . In the Maximization Step, we use the images at pose k to optimize the corresponding model parameters $\{\mathbf{F}_k, \mathbf{G}_k, \boldsymbol{\Sigma}_k\}$ of pose k .

Verification

In Tied PLDA face verification is treated as a model selection problem. We make the verification decision

by comparing the likelihood of two generative models: $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)$ and $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s)$. Model \mathcal{M}_d indicates that two images are from different people and model \mathcal{M}_s indicates that two images are from the same person.

If we have two images \mathbf{x}_1 and \mathbf{x}_2 , from which have been subtracted the relevant mean image \mathbf{m}_1 and \mathbf{m}_2 of the corresponding pose, and we assume they are independent, the likelihood of two images from different people $Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)$ is

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d) &= Pr(\mathbf{x}_1|\mathcal{M}_d)Pr(\mathbf{x}_2|\mathcal{M}_d) \\ &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_d] \\ &= \mathcal{G}_{\mathbf{x}} \begin{bmatrix} \mathbf{0}, & \mathbf{F}_1\mathbf{F}_1^T + \mathbf{G}_1\mathbf{G}_1^T + \Sigma_1 & \mathbf{0} \\ & \mathbf{0} & \mathbf{F}_2\mathbf{F}_2^T + \mathbf{G}_2\mathbf{G}_2^T + \Sigma_2 \end{bmatrix}, \end{aligned} \quad (5.5)$$

where the term \mathbf{x} is the concatenation of \mathbf{x}_1 and \mathbf{x}_2 ; the term Σ_d is non-match covariance matrix; the term $\{\mathbf{F}_1, \mathbf{G}_1, \Sigma_1\}$ is the model parameters at the pose of image \mathbf{x}_1 ; the term $\{\mathbf{F}_2, \mathbf{G}_2, \Sigma_2\}$ is the model parameters at the pose of image \mathbf{x}_2 .

If two images match (Model \mathcal{M}_s), the likelihood of two images is

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s) &= Pr(\mathbf{x}|\mathcal{M}_s) \\ &= \mathcal{G}_{\mathbf{x}}[\mathbf{0}, \Sigma_s] \\ &= \mathcal{G}_{\mathbf{x}} \begin{bmatrix} \mathbf{0}, & \mathbf{F}_1\mathbf{F}_1^T + \mathbf{G}_1\mathbf{G}_1^T + \Sigma_1 & \mathbf{F}_1\mathbf{F}_2^T \\ & \mathbf{F}_1\mathbf{F}_2^T & \mathbf{F}_2\mathbf{F}_2^T + \mathbf{G}_2\mathbf{G}_2^T + \Sigma_2 \end{bmatrix}, \end{aligned} \quad (5.6)$$

where the term Σ_s is the match covariance matrix.

With the above two likelihoods, we make the match decision by the log likelihood ratio $r(\mathbf{x}_1, \mathbf{x}_2)$ between the two models \mathcal{M}_s and \mathcal{M}_d .

$$\begin{aligned} r(\mathbf{x}_1, \mathbf{x}_2) &= \log \frac{Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s)}{Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d)} \\ &= \log Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_s) - \log Pr(\mathbf{x}_1, \mathbf{x}_2|\mathcal{M}_d) \\ &= \kappa + \mathbf{x}^T \Sigma_d^{-1} \mathbf{x} - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x}, \end{aligned} \quad (5.7)$$

where κ is a constant.

5.4 Tied Bayesian Face Recognition Algorithms

In this section, we apply the same idea as Tied PLDA to the Joint Bayesian Face algorithm and Joint PLDA. We propose two new algorithms: the Tied Joint Bayesian Face algorithm in section 5.4.1 and Joint PLDA in section 5.4.2.

5.4.1 The Tied Joint Bayesian Face Algorithm

Face Image Representation

We assume \mathbf{x}_1 is an image at pose 1 and \mathbf{x}_2 represents an image at pose 2. The image \mathbf{x}_1 has been subtracted with the mean of all training images at pose 1. The image \mathbf{x}_2 has been subtracted with the mean of all training images at pose 2. In the Tied Joint Bayesian Face algorithm, image \mathbf{x}_1 can be

described as the sum of the identity component α_1 and the within-individual variation component β_1 , image \mathbf{x}_2 can be described as the sum of the identity component α_2 and the within-individual variation component β_2 , so we have

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \quad (5.8)$$

or

$$\mathbf{x}' = \alpha' + \beta'. \quad (5.9)$$

The terms α_1 , α_2 , β_1 , and β_2 follow Gaussian distributions as

$$Pr(\alpha_1) = \mathcal{G}_{\alpha_1}[\mathbf{0}, \Sigma_{\alpha_1}] \quad (5.10)$$

$$Pr(\alpha_2) = \mathcal{G}_{\alpha_2}[\mathbf{0}, \Sigma_{\alpha_2}] \quad (5.11)$$

$$Pr(\beta_1) = \mathcal{G}_{\beta_1}[\mathbf{0}, \Sigma_{\beta_1}] \quad (5.12)$$

$$Pr(\beta_2) = \mathcal{G}_{\beta_2}[\mathbf{0}, \Sigma_{\beta_2}], \quad (5.13)$$

where Σ_{α_1} and Σ_{β_1} is the covariance matrix of the identity component and the covariance matrix of the within-individual variation component respectively for images at pose 1; Σ_{α_2} and Σ_{β_2} is the covariance matrix of the identity component and the covariance matrix of the within-individual variation component respectively for images at pose 2.

The joint distribution of an image pair \mathbf{x}' consisting of images from two poses can be written as

$$\begin{aligned} Pr(\mathbf{x}') &= Pr\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}\right) \\ &= \mathcal{G}_{\mathbf{x}'}[\mathbf{0}, \Sigma_{12}] \\ &= \mathcal{G}_{\mathbf{x}'}\left[\mathbf{0}, \begin{bmatrix} \Sigma_{\alpha_1} + \Sigma_{\beta_1} & \Sigma_{\alpha_{12}} \\ \Sigma_{\alpha_{12}}^T & \Sigma_{\alpha_2} + \Sigma_{\beta_2} \end{bmatrix}\right], \end{aligned} \quad (5.14)$$

where $\Sigma_{\alpha_{12}}$ is the covariance matrix of the identity component across pose 1 and pose 2.

Learning

Following [30], we develop an EM-like algorithm to learn covariance matrices $\xi = \{\Sigma_{\alpha_1}, \Sigma_{\alpha_2}, \Sigma_{\alpha_{12}}, \Sigma_{\beta_1}, \Sigma_{\beta_2}\}$. In the E-Step of our EM-like algorithm we estimate the identity component and the within-individual variation component of each training image. In the M-Step of the EM-like algorithm we update covariance matrices ξ .

For a given identity with m images at pose 1, n images at pose 2, the relationship between the images $\mathbf{x}' = [\mathbf{x}_{11} \cdots \mathbf{x}_{1m}, \mathbf{x}_{21} \cdots \mathbf{x}_{2n}]$ and the latent variables $\mathbf{y}' = [\alpha_1, \beta_{11} \cdots \beta_{1m}, \alpha_2, \beta_{21} \cdots \beta_{2n}]$ can be written as

$$\mathbf{x}' = \mathbf{P}\mathbf{y}' \quad (5.15)$$

where

$$\begin{aligned}
 \Sigma_{x'} &= \mathbf{P}\Sigma_{y'}\mathbf{P}^T \\
 &= \begin{bmatrix} \Sigma_{\alpha 1} + \Sigma_{\beta 1} & \Sigma_{\alpha 1} & \cdots & \Sigma_{\alpha 1} & \Sigma_{\alpha 12} & \Sigma_{\alpha 12} & \cdots & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 1} & \Sigma_{\alpha 1} + \Sigma_{\beta 1} & \cdots & \Sigma_{\alpha 1} & \Sigma_{\alpha 12} & \Sigma_{\alpha 12} & \cdots & \Sigma_{\alpha 12} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\alpha 1} & \Sigma_{\alpha 1} & \cdots & \Sigma_{\alpha 1} + \Sigma_{\beta 1} & \Sigma_{\alpha 12} & \Sigma_{\alpha 12} & \cdots & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 21} & \Sigma_{\alpha 21} & \cdots & \Sigma_{\alpha 21} & \Sigma_{\alpha 2} + \Sigma_{\beta 2} & \Sigma_{\alpha 2} & \cdots & \Sigma_{\alpha 2} \\ \Sigma_{\alpha 21} & \Sigma_{\alpha 21} & \cdots & \Sigma_{\alpha 21} & \Sigma_{\alpha 2} & \Sigma_{\alpha 2} + \Sigma_{\beta 2} & \cdots & \Sigma_{\alpha 2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\alpha 21} & \Sigma_{\alpha 21} & \cdots & \Sigma_{\alpha 21} & \Sigma_{\alpha 2} & \Sigma_{\alpha 2} & \cdots & \Sigma_{\alpha 2} + \Sigma_{\beta 2} \end{bmatrix} \\
 &= \begin{bmatrix} \bar{\Sigma}_1 & \bar{\Sigma}_{\alpha 12} \\ \bar{\Sigma}_{\alpha 21} & \bar{\Sigma}_2 \end{bmatrix}. \tag{5.20}
 \end{aligned}$$

In this part we describe **the E-Step of the EM-like algorithm**. For each individual, we estimate the distribution of latent variables \mathbf{y}' given all the images \mathbf{x}' associated with that individual and the parameters ξ^{t-1} at the previous iteration:

$$\begin{aligned}
 Pr(\mathbf{y}'|\mathbf{x}', \xi^{t-1}) &\propto Pr(\mathbf{x}'|\mathbf{y}', \xi^{t-1})Pr(\mathbf{y}') \\
 &= \mathcal{G}_{\mathbf{x}'}[\mathbf{P}\mathbf{y}', \Sigma_{x'}]\mathcal{G}_{\mathbf{y}'}[\mathbf{0}, \Sigma_{y'}] \\
 &\propto \mathcal{G}_{\mathbf{y}'}[\Sigma_{y'}\mathbf{P}^T(\mathbf{P}\Sigma_{y'}\mathbf{P}^T)^{-1}\mathbf{x}', \Sigma_{y'}]. \tag{5.21}
 \end{aligned}$$

According to the equation 5.18 and 5.19, we know that $\Sigma_{x'} = \mathbf{P}\Sigma_{y'}\mathbf{P}^T$, so the equation 5.21 can be written as

$$Pr(\mathbf{y}'|\mathbf{x}', \xi^{t-1}) \propto \mathcal{G}_{\mathbf{y}'}[\Sigma_{y'}\mathbf{P}^T\Sigma_{x'}^{-1}\mathbf{x}', \Sigma_{y'}]. \tag{5.22}$$

The expectation of the hidden variable \mathbf{y}' is

$$\begin{aligned}
 E(\mathbf{y}'|\mathbf{x}') &= \Sigma_{y'}\mathbf{P}^T\Sigma_{x'}^{-1}\mathbf{x}' \\
 &= \begin{bmatrix} \Sigma_{\alpha 1} & \cdots & \Sigma_{\alpha 1} & \Sigma_{\alpha 12} & \cdots & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 21} & \cdots & \Sigma_{\alpha 21} & \Sigma_{\alpha 2} & \cdots & \Sigma_{\alpha 2} \\ \Sigma_{\beta 1} & & & & & \\ & \ddots & & & & \\ & & \Sigma_{\beta 1} & & & \\ & & & \Sigma_{\beta 2} & & \\ & & & & \ddots & \\ & & & & & \Sigma_{\beta 2} \end{bmatrix} \Sigma_{\mathbf{x}'}^{-1} \begin{bmatrix} \mathbf{x}_{11} \\ \vdots \\ \mathbf{x}_{1m} \\ \mathbf{x}_{21} \\ \vdots \\ \mathbf{x}_{2n} \end{bmatrix}. \tag{5.23}
 \end{aligned}$$

It is expensive to compute the term $\Sigma_{x'}^{-1}$ of the equation 5.23. Fortunately the computation complexity can be reduced by taking the advantage of the block-wise structure of the matrix. We can follow

Shur's lemma as described in section 3.3.1 to compute the inversion term :

$$\begin{aligned}\Sigma_{x'}^{-1} &= \begin{bmatrix} \bar{\Sigma}_1 & \bar{\Sigma}_{\alpha 12} \\ \bar{\Sigma}_{\alpha 21} & \bar{\Sigma}_2 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} & -(\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \\ -\bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} & \bar{\Sigma}_2^{-1} + \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \end{bmatrix}.\end{aligned}$$

After we obtain the the expectation of the hidden variable $E(\mathbf{y}'|\mathbf{x}')$ defined in the equation 5.23, we can extract the identity component and the within-individual variation component of each image. The identity component $\{\alpha_{11}, \dots, \alpha_{1m}\}$ and the within-individual variation component $\{\beta_{11}, \dots, \beta_{1m}\}$ for images at pose 1 can be obtained by:

$$[\alpha_{11}^T \dots \alpha_{1m}^T]^T = \begin{bmatrix} \Sigma_{\alpha 1} & \dots & \Sigma_{\alpha 1} & \Sigma_{\alpha 12} & \dots & \Sigma_{\alpha 12} \end{bmatrix} \Sigma_x^{-1} [\mathbf{x}_{11}^T \dots \mathbf{x}_{1m}^T \mathbf{x}_{21}^T \dots \mathbf{x}_{2n}^T]^T \quad (5.24)$$

$$\begin{aligned}[\beta_{11}^T \dots \beta_{1m}^T]^T &= \text{diag}[\Sigma_{\beta 1} \dots \Sigma_{\beta 1}] [(\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \\ &\quad - (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1}] [\mathbf{x}_{11}^T \dots \mathbf{x}_{1m}^T \mathbf{x}_{21}^T \dots \mathbf{x}_{2n}^T]^T \quad (5.25)\end{aligned}$$

The identity component $\{\alpha_{21}, \dots, \alpha_{2n}\}$ and the within-individual variation component $\{\beta_{21}, \dots, \beta_{2n}\}$ for images at pose 2 can be obtained by:

$$[\alpha_{21}^T \dots \alpha_{2n}^T]^T = \begin{bmatrix} \Sigma_{\alpha 21} & \dots & \Sigma_{\alpha 21} & \Sigma_{\alpha 2} & \dots & \Sigma_{\alpha 2} \end{bmatrix} \Sigma_x^{-1} [\mathbf{x}_{11}^T \dots \mathbf{x}_{1m}^T \mathbf{x}_{21}^T \dots \mathbf{x}_{2n}^T]^T \quad (5.26)$$

$$\begin{aligned}[\beta_{21}^T \dots \beta_{2n}^T]^T &= \text{diag}[\Sigma_{\beta 2} \dots \Sigma_{\beta 2}] [-\bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \\ &\quad \bar{\Sigma}_2^{-1} + \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T (\bar{\Sigma}_1 - \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1} \bar{\Sigma}_{\alpha 12}^T)^{-1} \bar{\Sigma}_{\alpha 12} \bar{\Sigma}_2^{-1}] [\mathbf{x}_{11}^T \dots \mathbf{x}_{1m}^T \mathbf{x}_{21}^T \dots \mathbf{x}_{2n}^T]^T.\end{aligned} \quad (5.27)$$

In the **M-Step of the EM-like algorithm**, we update the parameters $\xi = \{\Sigma_{\alpha 1}, \Sigma_{\alpha 2}, \Sigma_{\alpha 12}, \Sigma_{\beta 1}, \Sigma_{\beta 2}\}$ by

$$\begin{bmatrix} \Sigma_{\alpha 1} & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 21} & \Sigma_{\alpha 2} \end{bmatrix} = \text{cov} \left(\begin{bmatrix} \alpha_a \\ \alpha_b \end{bmatrix} \right) \quad (5.28)$$

$$\Sigma_{\beta 1} = \text{cov}(\beta_a) \quad (5.29)$$

$$\Sigma_{\beta 2} = \text{cov}(\beta_b), \quad (5.30)$$

where α_a and α_b are the identity components of training images at pose 1 and pose 2 respectively; the term β_a and β_b are the within-individual variation component of training images at pose 1 and pose 2 respectively.

Verification

Similarly to the Joint Bayesian Face algorithm described in section 4.2.3, we make the match decision based on the likelihood ratio between two generative models: $Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d)$ and $Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s)$. Model \mathcal{M}_d denotes two images are from different people and model \mathcal{M}_s means two images are from the same person. If we assume image \mathbf{x}_1 at pose 1 and image \mathbf{x}_2 at pose 2 are from the same identity

and are independent, then the joint probability of two images can be derived as

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s) &= \mathcal{G}_{\mathbf{x}} [\mathbf{0}, \boldsymbol{\Sigma}_s] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma}_{\alpha 1} + \boldsymbol{\Sigma}_{\beta 1} & \boldsymbol{\Sigma}_{\alpha 12} \\ \boldsymbol{\Sigma}_{\alpha 12}^T & \boldsymbol{\Sigma}_{\alpha 2} + \boldsymbol{\Sigma}_{\beta 2} \end{bmatrix} \right], \end{aligned} \quad (5.31)$$

where $\boldsymbol{\Sigma}_s$ is the match covariance matrix.

When two images are from different people and are assumed to be generated independently, we have

$$\begin{aligned} Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d) &= \mathcal{G}_{\mathbf{x}} [\mathbf{0}, \boldsymbol{\Sigma}_d] \\ &= \mathcal{G}_{\mathbf{x}} \left[\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma}_{\alpha 1} + \boldsymbol{\Sigma}_{\beta 1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\alpha 2} + \boldsymbol{\Sigma}_{\beta 2} \end{bmatrix} \right], \end{aligned} \quad (5.32)$$

where $\boldsymbol{\Sigma}_d$ is the non-match covariance matrix.

The final matching decision is based on the log likelihood ratio $r(\mathbf{x}_1, \mathbf{x}_2)$ between two model \mathcal{M}_s and \mathcal{M}_d :

$$\begin{aligned} r(\mathbf{x}_1, \mathbf{x}_2) &= \log \left[\frac{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_s)}{Pr(\mathbf{x}_1, \mathbf{x}_2 | \mathcal{M}_d)} \right] \\ &\propto \mathbf{x}^T \boldsymbol{\Sigma}_d^{-1} \mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_s^{-1} \mathbf{x}. \end{aligned} \quad (5.33)$$

5.4.2 Tied Joint PLDA

In this section we will describe the tied version of Joint PLDA.

Face Representation

In the Tied Joint PLDA a face image \mathbf{x}_{ijk} can be represented as the sum of the identity component $\boldsymbol{\alpha}_{ik}$ and the within-individual variation component $\boldsymbol{\beta}_{ijk}$:

$$\mathbf{x}_{ijk} = \boldsymbol{\alpha}_{ik} + \boldsymbol{\beta}_{ijk} \quad (5.34)$$

$$\boldsymbol{\alpha}_{ik} = \mathbf{F}_k \mathbf{h}_i \quad (5.35)$$

$$\boldsymbol{\beta}_{ijk} = \mathbf{G}_k \mathbf{w}_{ijk} + \boldsymbol{\epsilon}_{ijk}, \quad (5.36)$$

where the term \mathbf{x}_{ijk} denotes the j^{th} image of the i^{th} individual at pose k with the mean of all face images subtracted; the identity component $\boldsymbol{\alpha}_{ik}$ is equivalent to the term $\mathbf{F}_k \mathbf{h}_i$ of Tied PLDA and the within-individual variation component $\boldsymbol{\beta}_{ijk}$ is equivalent to the term $\mathbf{G}_k \mathbf{w}_{ijk} + \boldsymbol{\epsilon}_{ijk}$ of Tied PLDA. Therefore, it can also be described in terms of conditional probabilities:

$$Pr(\mathbf{x}_{ijk}) = \mathcal{G}_{\mathbf{x}_{ijk}} [\mathbf{F}_k \mathbf{h}_i + \mathbf{G}_k \mathbf{w}_{ijk}, \boldsymbol{\Sigma}_k] \quad (5.37)$$

$$Pr(\mathbf{h}_i) = \mathcal{G}_{\mathbf{h}_i} [\mathbf{0}, \mathbf{I}] \quad (5.38)$$

$$Pr(\mathbf{w}_{ijk}) = \mathcal{G}_{\mathbf{w}_{ijk}} [\mathbf{0}, \mathbf{I}], \quad (5.39)$$

where the term \mathbf{F}_k contains the basis functions of the between-individual subspace in columns for pose k ; the term \mathbf{h}_i denotes the hidden identity variable which is constant for all images at different poses from a identity; \mathbf{G}_k contains the basis functions of the within-individual subspace in columns for pose

\mathbf{w}_{ijk} denotes the hidden noise variable which is different for each image; Σ_k is a diagonal covariance matrix for stochastic noise of images at pose k .

We assume we have image \mathbf{x}_{ij1} at pose 1 and \mathbf{x}_{ij2} at pose 2. According to the equation 5.34, images \mathbf{x}_{ij1} and \mathbf{x}_{ij2} can be described as:

$$\mathbf{x}_{ij1} = \boldsymbol{\alpha}_{i1} + \boldsymbol{\beta}_{ij1} \quad (5.40)$$

$$\mathbf{x}_{ij2} = \boldsymbol{\alpha}_{i2} + \boldsymbol{\beta}_{ij2}. \quad (5.41)$$

The terms $\boldsymbol{\alpha}_{i1}$, $\boldsymbol{\beta}_{ij2}$, $\boldsymbol{\alpha}_{i2}$, and $\boldsymbol{\beta}_{ij2}$ follow Gaussian distribution:

$$Pr(\boldsymbol{\alpha}_{i1}) = \mathcal{G}_{\boldsymbol{\alpha}_{i1}}[\mathbf{0}, \Sigma_{\boldsymbol{\alpha}1}] \quad (5.42)$$

$$Pr(\boldsymbol{\beta}_{ij1}) = \mathcal{G}_{\boldsymbol{\beta}_{ij1}}[\mathbf{0}, \Sigma_{\boldsymbol{\beta}1}] \quad (5.43)$$

$$Pr(\boldsymbol{\alpha}_{i2}) = \mathcal{G}_{\boldsymbol{\alpha}_{i2}}[\mathbf{0}, \Sigma_{\boldsymbol{\alpha}2}] \quad (5.44)$$

$$Pr(\boldsymbol{\beta}_{ij2}) = \mathcal{G}_{\boldsymbol{\beta}_{ij2}}[\mathbf{0}, \Sigma_{\boldsymbol{\beta}2}], \quad (5.45)$$

where the terms $\Sigma_{\boldsymbol{\alpha}1}$ and $\Sigma_{\boldsymbol{\beta}1}$ are the covariance matrix of the identity component and the covariance matrix of the within-individual variation component respectively for images at pose 1; the terms $\Sigma_{\boldsymbol{\alpha}2}$ and $\Sigma_{\boldsymbol{\beta}2}$ are the covariance matrix of the identity component and the covariance matrix of the within-individual variation component respectively for images at pose 2.

The joint distribution of an image pair \mathbf{x}' follows the Gaussian distribution:

$$Pr(\mathbf{x}') = Pr\left(\begin{bmatrix} \mathbf{x}_{ij1} \\ \mathbf{x}_{ij2} \end{bmatrix}\right) \quad (5.46)$$

$$= \mathcal{G}_{\mathbf{x}'}[\mathbf{0}, \Sigma_{12}] \quad (5.47)$$

$$= \mathcal{G}_{\mathbf{x}'}\left[\mathbf{0}, \begin{bmatrix} \Sigma_{\boldsymbol{\alpha}1} + \Sigma_{\boldsymbol{\beta}1} & \Sigma_{\boldsymbol{\alpha}12} \\ \Sigma_{\boldsymbol{\alpha}12}^T & \Sigma_{\boldsymbol{\alpha}2} + \Sigma_{\boldsymbol{\beta}2} \end{bmatrix}\right], \quad (5.48)$$

where $\Sigma_{\boldsymbol{\alpha}12}$ is the covariance matrix of the identity component across the pose 1 and pose 2.

Learning

We attempt to estimate the covariance matrices $\{\Sigma_{\boldsymbol{\alpha}1}, \Sigma_{\boldsymbol{\beta}1}, \Sigma_{\boldsymbol{\alpha}2}, \Sigma_{\boldsymbol{\beta}2}, \Sigma_{\boldsymbol{\alpha}12}\}$ from training images. We first apply the EM algorithm of Tied PLDA to estimate the optimal model parameters $\boldsymbol{\theta} = \{\mathbf{F}_k, \mathbf{G}_k, \Sigma_k\}$, then we apply the E-Step of Tied PLDA training method defined in [82] to estimate the expectation of the hidden variable \mathbf{y}_{ijk} for each training image \mathbf{x}_{ijk} :

$$E[\mathbf{y}_{ijk}] = (\mathbf{A}_k^T \Sigma_k'^{-1} \mathbf{A}_k + \mathbf{I})^{-1} \mathbf{A}_k^T \Sigma_k'^{-1} \mathbf{x}_{ijk}, \quad (5.49)$$

where

$$\mathbf{y}_{ijk} = \begin{bmatrix} \mathbf{h}_i \\ \mathbf{w}_{ijk} \end{bmatrix} \quad (5.50)$$

$$\mathbf{A}_k = [\mathbf{F}_k \quad \mathbf{G}_k] \quad (5.51)$$

$$\Sigma_k' = \begin{bmatrix} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_k \end{bmatrix}. \quad (5.52)$$

Then we can obtain the identity component α_{ij1} and the within-individual variation component β_{ij1} for each image \mathbf{x}_{ij1} at pose 1 by

$$\alpha_{ij1} = \mathbf{F}_1 \mathbf{h}_i \quad (5.53)$$

$$\beta_{ij1} = \mathbf{x}_{ij1} - \mathbf{F}_1 \mathbf{h}_i, \quad (5.54)$$

where the term \mathbf{F}_1 contains the basis functions of the between-individual subspace in columns at pose 1; the term \mathbf{h}_i is the hidden identity variables for identity i .

Using a similar method we obtain the identity component α_{ij2} and the within-individual variation component β_{ij2} for each image \mathbf{x}_{ij2} at pose 2:

$$\alpha_{ij2} = \mathbf{F}_2 \mathbf{h}_i \quad (5.55)$$

$$\beta_{ij2} = \mathbf{x}_{ij2} - \mathbf{F}_2 \mathbf{h}_i, \quad (5.56)$$

where \mathbf{F}_2 contains the basis functions of the between-individual subspace in columns at pose 2.

Lastly, we update parameters $\{\Sigma_{\alpha 1}, \Sigma_{\alpha 2}, \Sigma_{\alpha 12}, \Sigma_{\beta 1}, \Sigma_{\beta 2}\}$ by

$$\begin{bmatrix} \Sigma_{\alpha 1} & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 21} & \Sigma_{\alpha 2} \end{bmatrix} = \text{cov} \left(\begin{bmatrix} \alpha_a \\ \alpha_b \end{bmatrix} \right) \quad (5.57)$$

$$\Sigma_{\beta 1} = \text{cov}(\beta_a) \quad (5.58)$$

$$\Sigma_{\beta 2} = \text{cov}(\beta_b), \quad (5.59)$$

where the terms α_a and α_b are the identity components of training images at pose 1 and pose 2 respectively; the terms β_a and β_b are the within-individual variation components of training images at pose 1 and pose 2 respectively.

Verification

Similar to the Tied Joint Bayesian Face algorithm, the verification decision is made based on the likelihood ratio:

$$r(\mathbf{x}_1, \mathbf{x}_2) \propto \mathbf{x}^T \Sigma_d^{-1} \mathbf{x} - \mathbf{x}^T \Sigma_s^{-1} \mathbf{x},$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

$$\Sigma_d = \begin{bmatrix} \Sigma_{\alpha 1} + \Sigma_{\beta 1} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\alpha 2} + \Sigma_{\beta 2} \end{bmatrix}$$

$$\Sigma_s = \begin{bmatrix} \Sigma_{\alpha 1} + \Sigma_{\beta 1} & \Sigma_{\alpha 12} \\ \Sigma_{\alpha 12}^T & \Sigma_{\alpha 2} + \Sigma_{\beta 2} \end{bmatrix}.$$

5.5 New Database

In this section will first analyze the pose distribution of the training images of the LFW database to explain the motivation to collect a new database. Then we will describe the collection scheme for the new database.

5.5.1 Motivation

In the LFW database, images are divided into 10 subsets without overlapping in identities and images. A leave-one-out validation scheme is applied to evaluate the recognition performance. When one subset is selected for testing and the remainder of the 9 subsets can be used for training. The final performance is evaluated by repeating the verification experiments in each of 10 test subsets.

Test folds	Category	0-15	0-30	0-45	0-60	15-30	15-45	15-60	30-45	30-60	45-60
1	People	730	217	35	7	158	27	7	18	3	1
	Pairs	3494	1928	1099	74	682	294	24	64	5	1
2	People	735	221	40	3	153	29	4	22	2	0
	Pairs	3553	1790	1105	17	612	289	8	71	2	0
3	People	739	218	42	7	155	31	6	21	3	1
	Pairs	3671	1968	1197	74	684	310	23	71	5	1
4	People	748	224	37	7	159	29	7	22	3	1
	Pairs	3660	1869	1014	74	659	273	24	70	5	1
5	People	763	222	39	7	157	31	7	21	3	1
	Pairs	361	1903	1199	74	695	320	24	70	5	1
6	People	735	217	38	7	159	30	7	21	3	1
	Pairs	3610	1903	1101	74	683	298	24	68	5	1
7	People	721	209	37	7	148	28	7	20	3	1
	Pairs	3294	1717	1159	74	609	293	24	67	5	1
8	People	751	211	41	6	150	31	6	22	2	1
	Pairs	3671	1906	1201	73	657	322	22	74	4	1
9	People	748	211	41	6	157	32	6	21	3	1
	Pairs	3637	1904	1216	64	668	325	20	72	5	1
10	People	744	219	37	6	161	29	6	19	2	1
	Pairs	3640	1929	707	68	639	210	23	48	4	1

Table 5.2: **Identity number and possible training pairs for each pair group in each of 10 cross-validation experiments.** In the LFW database images are divided into 10 folds. One fold is used for testing and the rest 9 folds are used for training. The final performance is reported based on 10 cross-validation experiments. We label each LFW image by a pose of pose list $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$. When one fold is chosen as the test fold, we list the number of training identities and possible training pairs for each pair category, to which each training pair is assigned based the poses of the two images.

As we described in section 5.2, we assign each LFW image to a pre-defined pose category $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$. By analyzing the image pairs of the LFW test sets, we find that we need to train Tied Bayesian models for the following pair groups, which comprise $\{0^\circ-0^\circ\}$, $\{15^\circ-15^\circ\}$, $\{30^\circ-30^\circ\}$, $\{45^\circ-45^\circ\}$, $\{0^\circ-15^\circ\}$, $\{15^\circ-30^\circ\}$, $\{30^\circ-45^\circ\}$, $\{0^\circ-30^\circ\}$, $\{15^\circ-45^\circ\}$, $\{30^\circ-60^\circ\}$, $\{0^\circ-45^\circ\}$, $\{15^\circ-60^\circ\}$, $\{0^\circ-60^\circ\}$. We follow the ‘unrestricted configuration’ to use the LFW training images. We list the available identities and image pairs that can be used in training for each pair group in 10

cross-validation experiments in Table 5.2.

From the table 5.2 we find that there are insufficient training images for pair groups with large pose changes, especially those with over 30° difference. For example, there is 0 or only 1 training pair for pair group $\{15^\circ - 60^\circ\}$ in 10 experiments, it is impossible to train a Tied Bayesian model for this pair group. Therefore, to train Tied Bayesian face recognition algorithms, it is necessary to have more training image pairs with large pose differences.

The Multi-PIE database [56] contains 755,370 images from 345 individuals with 6 expressions and 19 lighting conditions and 15 poses under four sessions. Therefore, it can provide enough training images for any pair groups. However, images of the Multi-PIE database are collected in the controlled environments. We hypothesize that the Multi-PIE images might not provide an equivalent amount of image variation to the uncontrolled test images of the LFW database. Therefore, we deemed it necessary to collect a new database.

5.5.2 The UCL Multi-Pose Face Database

To obtain sufficient training images with large pose variation, we use the same approach as for the LFW database and collect images from the internet. The new database is called the UCL Multi-Pose. The collection protocol was as follows: we first collect a list of celebrities' names without overlapping with the LFW database, then we use the Google image search engine to obtain images of each celebrity. Since these images are captured in completely uncontrolled environments, they contain large variation as the LFW images include. We swap all left facing images to right facing. Then we check the pose of images manually and make sure there are at least three images at each pose of a list $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$ for each celebrity. Last we label four fiducial points (the left eye corner of the left eye, the nose bridge, the right eye corner of the right eye, the mouth top) and apply a similarity transformation to register each image into a pre-defined template. Similar to the LFW database, identity information is provided for each image. Figure 5.2 shows several examples from the UCL Multi-Pose database.

Although the collection spirit is the same for both the LFW database and the UCL Multi-Pose database, there are some significant differences:

- The LFW database applies Viola-Jones face detector [134] to collect images so most of images are near frontal. The UCL Multi-Pose database is designed to collect more non-frontal images to train Tied Bayesian face recognition algorithms: there are more non-frontal images than the LFW database.
- The LFW database contains 13,233 images from 5,749 people while the UCL Multi-Pose database includes 7,485 images from 153 people. Therefore, the LFW database is much broader (more people) than the UCL Multi-Pose database. The image number per identity varies from 1 to 530 in the LFW database while the image number varies from 53 to 76 in the UCL Multi-Pose database, so the LFW database is much shallower (less images per person) than the UCL Multi-Pose database.



Figure 5.2: **Several sample images from the UCL Multi-Pose database.** The database consists of 7,450 images from 153 people. To provide sufficient training images with large pose difference, we collected at least 3 images at each pose of a list $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$ for each person. In this database all the left facing images have been swapped into right facing.

5.6 Experiments

In this section we compare the performance of the three Tied Bayesian face recognition algorithms. To verify their performance in dealing with pose variation in a controlled face database, we first compare the performance in the Multi-PIE database [56], in which multiple images at different poses are collected for each identity in the laboratory. We describe the experiment detail in section 5.6.2. To compare the performance of the three algorithms in uncontrolled database, we also conduct experiments in the UCL Multi-Pose database, in which at least three images at each pose of a pre-defined pose category are captured for each identity from the internet. The experiment detail is described in section 5.6.3.

We also do two cross-data experiments. To solve the problem that the LFW database cannot provide sufficient training images with large pose difference, we train Tied Bayesian face recognition algorithms in the Multi-PIE database and test in the LFW database in section 5.6.4. Images of the Multi-PIE database are collected in the laboratory and might not capture sufficient within-individual variation as the uncontrolled LFW images contains, we also train Tied Bayesian face recognition algorithms in the UCL Multi-Pose database and test in the LFW database in section 5.6.5. Figure 5.3 illustrates the structure of experiments in this section.

5.6.1 Data Preprocessing

In this chapter we use three databases: the Multi-PIE database, the UCL Multi-Pose database, and the LFW database. Figure 5.4 shows several samples from the three databases. Instead of using image intensities, we extract Local Binary Patterns (LBP) descriptors [102] from the three databases to conduct

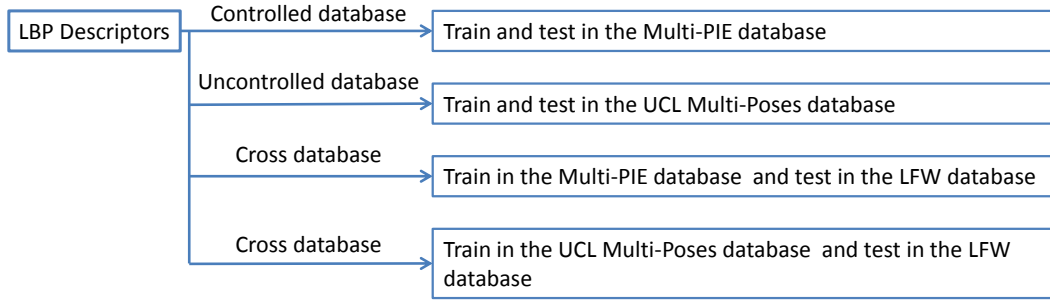


Figure 5.3: **Structure of experiments in this chapter.** We conduct four experiments using LBP image descriptors. To compare three Tied Bayesian face recognition algorithms in a controlled database, we do experiments in the Multi-PIE database. To compare the performance in an uncontrolled database, we also conduct experiments in the UCL Multi-Pose database. To provide sufficient training images for the three Tied Bayesian face recognition algorithms from a controlled database, we train the three algorithms in the Multi-PIE database and test them in the LFW database. To provide sufficient training images for the three Tied algorithms from an uncontrolled database, we train three Tied algorithms in the UCL Multi-Pose database and test in the LFW database.

our experiments. We will describe the extraction process in the three databases in turn.

For the Multi-PIE database we use frontal lit images with rightward horizontal pose 0° , 15° , 30° , 45° , 60° , 75° , 90° . Under this constraint each of 337 people has several images at each pose of a finite pose category $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$. We label four fiducial points (the left corner of the left eye, the nose bridge, the right corner of the right eye, the mouth top) of each image by hand and then apply a similarity transformation to register each image to a pre-defined template based on four fiducial points; Next we crop the central 160×80 pixels of each image to obtain the face region; Finally, we extract LBP descriptors from face regions by the following settings: we divide each 160×80 cropped image into several non-overlapping 12×12 patches, the radius to form neighborhood over each pixel location is set to 3, the number of neighbor points is set to 8, uniform binary patterns are applied. After we obtain the LBP histograms for all the patches, we normalize the histograms in each patch to unit length and truncate their values at 0.2, then normalize again to unit length. The face image is represented by the concatenation of all the LBP histograms from all patches.

In the UCL Multi-Pose database, all the images are right facing and have been registered. We crop the central 160×80 pixels of each image to obtain the face region and extract LBP descriptors using the same method as in the Multi-PIE database.

For the LFW database, we use the original 250×250 images without any alignment and swap all the left facing images as right facing. Then we assign each image to a pre-defined pose category $\{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ\}$ by hand. Next we use the same method to extract LBP descriptors as in the Multi-PIE database: we label four fiducial points of each image by hand, apply a similarity transformation to register images based the four fiducial points, crop the central 160×80 pixels of each image, extract LBP descriptors.



Figure 5.4: **Several image examples from (a) the Multi-PIE database, (b) the UCL Multi-Pose database, (c) the LFW database.** All the images in the three databases have been registered to a pre-defined template by applying a similarity transformation based on four manually labeled fiducial points (the left eye corner of the left eye, the nose bridge, the right eye corner of the right eye, the mouth top).

In all the following experiments, for Tied PLDA and Tied Joint PLDA, we always use 25 iterations of the EM algorithm to train the model parameters $\{\mathbf{F}_k, \mathbf{G}_k, \Sigma_k\}$, which are initialized to random values. For Tied Joint Bayesian Face algorithm we always use 5 iterations of the EM-like algorithm to train the model parameters $\{\Sigma_{\alpha_1}, \Sigma_{\alpha_2}, \Sigma_{\alpha_{12}}, \Sigma_{\beta_1}, \Sigma_{\beta_2}\}$, which are initialized to random values.

5.6.2 Train and Test in the Multi-PIE Database

In this section we compare the performance of three Tied Bayesian face recognition algorithms in the controlled Multi-PIE database. We train and test three algorithms only using the Multi-PIE images.

We compare face verification performance of three Tied Bayesian face recognition algorithms for the following pair groups: $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$, $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$, $\{0^\circ - 75^\circ\}$, and $\{0^\circ - 90^\circ\}$. For each pair group we use images of the first 237 people in training and images of the remaining 100 identities in test. There is no overlap in identities and images between the training set and the test set. In test, for each pair group, we collect 1458 matched pairs using images of each test identity. We also collect 6021 non-matched pairs by combining images of each test identity with images from 5 other random test identities. In total we verify 7479 pairs for each pair group.

For all the three Tied Bayesian Face algorithms we first apply PCA to reduce the dimensions. For Tied PLDA and Tied Joint PLDA we set the PCA dimensions to 200, subspace dimensions to 128. For the Tied Joint Bayesian Face algorithm we set the PCA dimensions to 400. These experiment settings are obtained using an empirical approach.

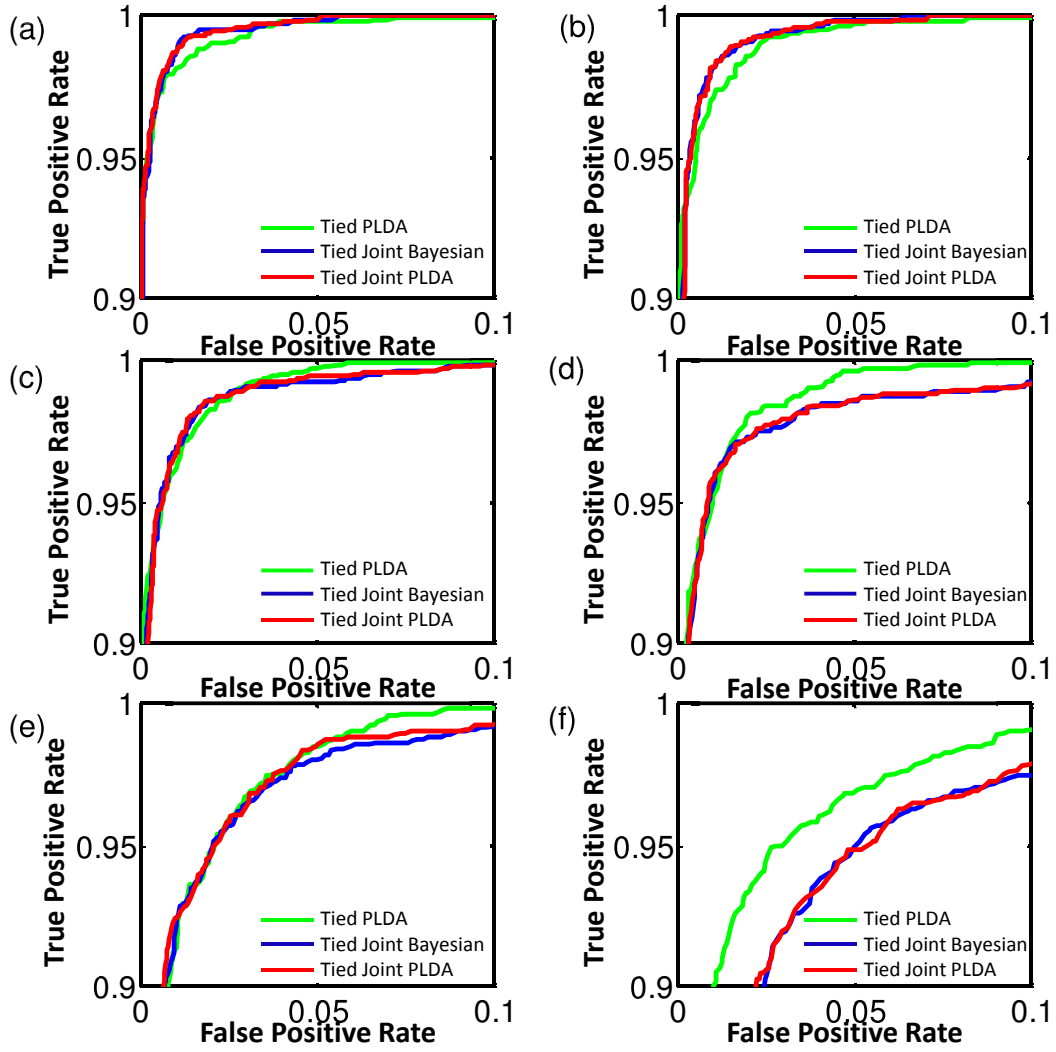


Figure 5.5: The performance of three Tied Bayesian face recognition algorithms for 6 pair groups is shown by ROC curves when trained and tested in the Multi-PIE database. The performance of three algorithms for pair groups $\{0^\circ - 15^\circ\}$ is illustrated in (a), $\{0^\circ - 30^\circ\}$ in (b), $\{0^\circ - 45^\circ\}$ in (c), $\{0^\circ - 60^\circ\}$ in (d), $\{0^\circ - 75^\circ\}$ in (e) and $\{0^\circ - 90^\circ\}$ in (f).

Algorithm \ Results	0-15	0-30	0-45	0-60	0-75	0-90
Tied PLDA	0.9993	0.9989	0.9988	0.9984	0.9969	0.9952
Tied Joint Bayesian Face	0.9994	0.9991	0.9985	0.9971	0.9855	0.9910
Tied Joint PLDA	0.9994	0.9990	0.9985	0.9971	0.9958	0.9912

Table 5.3: Area under the ROC curve of Figure 5.5. Larger area means better verification performance.

The performance of three Tied Bayesian face recognition algorithms are reported by the Receiver

Operator Characteristic (ROC) curves in Figure 5.5. More details regarding ROC curve are described in section 2.3.2. To show the performance difference among the three algorithms more clearly, we also list area under the ROC curve in Table 5.3. From Figure 5.5 and Table 5.3 we find that the performance of Tied Joint PLDA and the Tied Joint Bayesian Face algorithm is very similar for all six pair groups. We also find that Tied Joint PLDA and the Tied Joint Bayesian Face algorithm performs slightly better than Tied PLDA when pose difference is less than 45 degrees. However, Tied PLDA produces better performance when large pose variation exists. For example, for pair group $\{0^\circ - 90^\circ\}$, the performance of Tied PLDA is much better than the performance of Tied Joint PLDA and the Tied Joint Bayesian Face algorithm.

5.6.3 Train and Test in the UCL Multi-Pose Database

In this part we compare the performance of three Tied Bayesian Face algorithms in the uncontrolled UCL Multi-Pose database. We train and test our models only using the UCL Multi-Pose images. We verify the performance of the three algorithms for the following pair groups: $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$, $\{0^\circ - 45^\circ\}$, and $\{0^\circ - 60^\circ\}$. For each pair group we use images from the first 152 people in training and images of the remaining 101 individuals in test. There is no overlap in identities and images between the training set and the test set. In test, for each pair group, we collect 1990 matched pairs using images of each test identity. We also collect 7560 non-matched pairs by combining images of each test identity with images from 10 random other test identities. In total we verify 9469 pairs for each pair group.

For all the three Tied Bayesian face recognition algorithms, we apply PCA to reduce the dimensions. For Tied PLDA and Tied Joint PLDA, we set the PCA dimensions to 100, subspace dimensions to 32. For the Tied Joint Bayesian Face algorithm we set the PCA dimensions to 100. These settings are obtained by an empirical approach.

We report the performance of the three algorithms by ROC curves in Figure 5.6. To show the performance difference of the three algorithms more clearly, we also list area under the ROC curve in Table 5.4. From Figure 5.6 and Table 5.4 we find that the Tied Joint Bayesian Face algorithm and Tied Joint PLDA produce better performance than Tied PLDA for all pair groups. The reason might be as our conclusion in chapter 4 that more discriminatory information might be captured when estimating covariance matrix without making low dimension assumption. From Table 5.4 we also find that the performance of Tied Joint PLDA and the Tied Joint Bayesian Face algorithm are very similar.

We compare Figure 5.6 and Figure 5.5 and find that the performance of all the three algorithms in the UCL Multi-Pose database is much worse than the results in the Multi-PIE database: face verification in the uncontrolled UCL Multi-Pose database is probably more difficult than in the controlled Multi-PIE database.

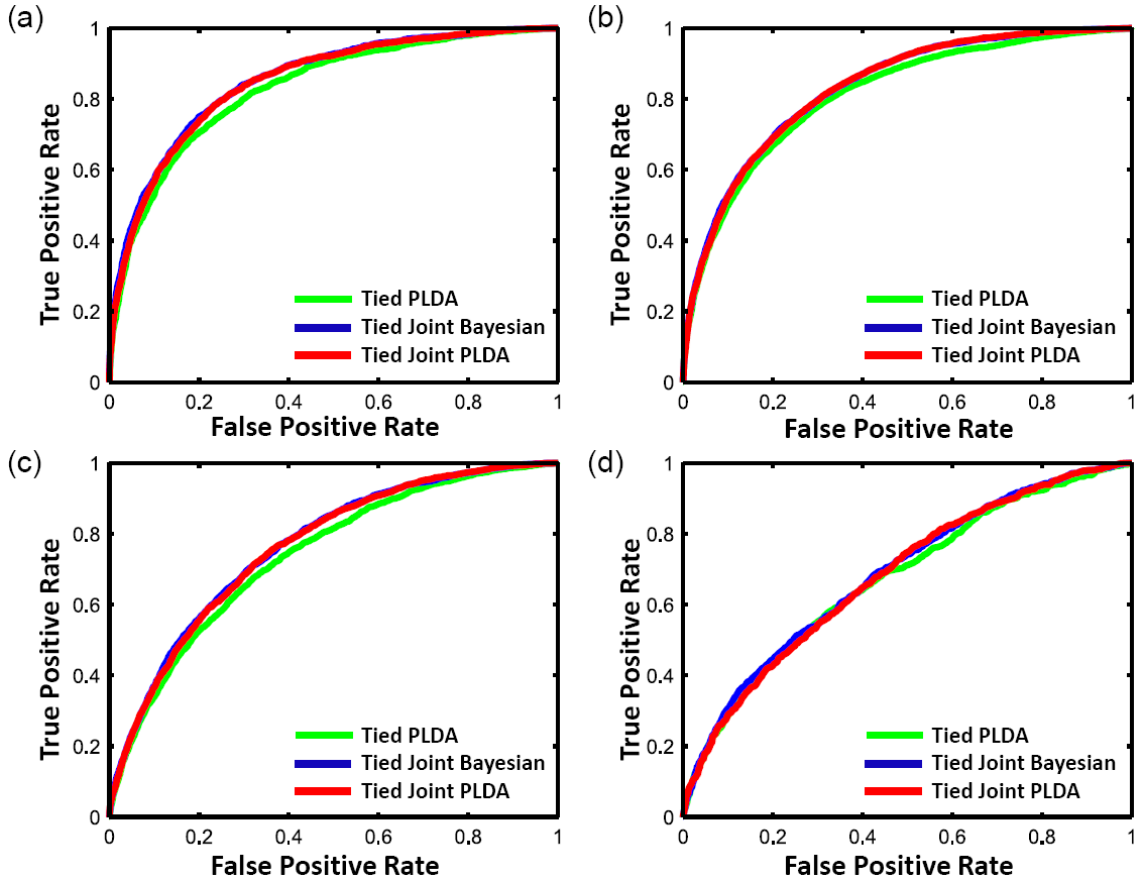


Figure 5.6: The face verification performance of the three Tied Bayesian face recognition algorithms for 4 pair groups is reported by ROC curves when trained and tested in the UCL Multi-Pose database. The performance of three algorithms for pair groups $\{0^\circ - 15^\circ\}$ is illustrated in (a), $\{0^\circ - 30^\circ\}$ in (b), $\{0^\circ - 45^\circ\}$ in (c), $\{0^\circ - 60^\circ\}$ in (d).

Algorithm \ Results	0-15	0-30	0-45	0-60
Tied PLDA	0.8306	0.8142	0.7395	0.6749
Tied Joint Bayesian Face	0.8528	0.8333	0.7643	0.6867
Tied Joint LDA	0.8488	0.8324	0.7619	0.6816

Table 5.4: Area under the ROC curve of Figure 5.6. Larger area means better verification performance.

5.6.4 Train in the Multi-PIE Database and Test in the LFW Database

In this part we compare the performance of three Tied Bayesian Face algorithms when trained in the Multi-PIE database but tested in the LFW database. We compare the performance of the three Tied

algorithms for the following pair groups: $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$, $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$, $\{15^\circ - 30^\circ\}$, $\{15^\circ - 45^\circ\}$, $\{15^\circ - 60^\circ\}$, $\{30^\circ - 45^\circ\}$, and $\{30^\circ - 60^\circ\}$. There are no LFW test pairs belong to pair group $\{45^\circ - 60^\circ\}$, so we do not list it.

For each pair group, we train and test Tied models as follows: we use all the relevant Multi-PIE images at the two poses to train Tied models. In test, the LFW evaluation protocol applies a leave-one-out cross validation scheme. Images are divided into 10 subsets. Each subset is selected as the test fold in turn and the final performance is based on the test results of ten experiments, each of them uses a different test fold. To obtain the result for each of 10 test folds, we collect the image pairs belonging to the target pair group and compute the % correct. We repeat the experiments in 10 test folds and report the final performance for the specified pair group by the mean % correct in 10 test folds and the standard error of the mean.

Pose Difference	Pair Groups	Bayesian face recognition algorithms trained and tested in the LFW database			Tied Bayesian face recognition algorithms trained in the Multi-PIE database and tested in the LFW database		
		PLDA	Joint Bayesian Face	Joint PLDA	Tied PLDA	Tied Joint Bayesian Face	Tied Joint PLDA
15	0-15	90.029 ±0.873	89.231 ±0.782	89.492 ±0.664	78.593 ±0.860	78.913 ±0.931	79.251 ±0.903
	15-30	87.048 ±1.223	87.362 ±1.202	87.656 ±1.084	77.950 ±1.268	78.151 ±1.225	79.434 ±1.168
	30-45	70.433 ±5.233	73.167 ±6.093	76.944 ±4.672	74.433 ±5.648	74.571 ±4.918	76.433 ±5.630
30	0-30	84.256 ±1.103	86.487 ±0.887	86.629 ±0.793	73.429 ±1.238	74.525 ±1.229	74.617 ±1.425
	15-45	82.662 ±2.985	77.765 ±3.000	81.523 ±3.614	66.228 ±5.070	69.451 ±7.091	70.570 ±7.219
	30-60	51.667 ±15.000	56.167 ±15.176	66.667 ±14.907	50.000 ±18.898	50.000 ±18.898	50.000 ±18.898
45	0-45	76.916 ±6.943	75.566 ±5.804	75.197 ±5.552	71.692 ±4.934	73.800 ±4.709	72.670 ±7.159
	15-60	38.333 ±14.498	38.333 ±14.498	38.333 ±14.498	61.111 ±15.316	52.778 ±13.205	52.778 ±13.205
60	0-60	51.071 ±12.644	61.071 ±12.099	61.071 ±12.099	69.643 ±11.655	69.643 ±10.395	69.643 ±10.395

Table 5.5: **Verification results when trained in the Multi-PIE database and tested in the LFW database.** We compare the performance of Bayesian face recognition algorithms (PLDA, the Joint Bayesian Face algorithm, and Joint PLDA) trained in the LFW database with the performance of Tied Bayesian face recognition algorithms (Tied PLDA, the Tied Joint Bayesian Face algorithm, and Tied Joint PLDA) trained in the Multi-PIE database. The tests are all conducted in the LFW database. We find that the performance of Bayesian face recognition algorithms is better than the performance of Tied Bayesian face recognition algorithm. Note: there is no result for pair group $\{45^\circ - 60^\circ\}$ because no LFW test image pairs exist in that pair group.

For all the three Tied Bayesian face recognition algorithms we apply PCA to reduce the dimensions.

For Tied PLDA and Tied Joint PLDA we set the PCA dimensions to 200, subspace dimensions to 128. For the Tied Joint Bayesian Face algorithm we set the PCA dimensions to 200. These optimal settings are obtained by an empirical approach.

In Table 5.5 we compare the performance of Bayesian face recognition algorithms trained in the LFW database with the performance of Tied Bayesian face recognition algorithms trained in the Multi-PIE database for different pair groups. The approach to train and test Bayesian face recognition algorithms for each pair group in the LFW database is described in section 5.2. From Table 5.5 we find that Tied Bayesian face recognition algorithms perform worse than Bayesian face recognition algorithms. The reason might be that images of the Multi-PIE database are collected in well controlled environments while images of the LFW database are collected from totally uncontrolled environments. These controlled Multi-PIE training images do not include an equivalent amount of image variation as the uncontrolled LFW test images.

From Table 5.5 we also find that Tied Joint PLDA performs best among three Tied algorithms. The reason might be that Tied Joint PLDA can estimate better covariance matrix by combining the advantages of Tied PLDA and the Tied Joint Bayesian Face algorithm. In Table 5.5 we also find that the results of the three Tied Bayesian face recognition algorithms are all exactly $50.000\% \pm 18.898$. It is because there are only $2 \sim 5$ image pairs in each test fold for pair group $\{30^\circ - 60^\circ\}$ as shown in table 5.2. The three Tied Bayesian face recognition algorithms all failed for this pair group.

5.6.5 Train in the UCL Multi-Pose Database and Test in the LFW Database

In this section we compare the performance of three Tied Bayesian face recognition algorithms when trained in the UCL Multi-Pose database and tested in the LFW database. The UCL Multi-Pose database was collected using the same method as for the LFW database. Therefore, compared to the Multi-PIE database, images of the UCL Multi-Pose database include more variation. The UCL Multi-Pose database is potentially better than the Multi-PIE database as a training database for face recognition in uncontrolled environments. As in section 5.6.4 we compare the performance of the three Tied algorithms for the following pair groups: $\{0^\circ - 15^\circ\}$, $\{0^\circ - 30^\circ\}$, $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$, $\{15^\circ - 30^\circ\}$, $\{15^\circ - 45^\circ\}$, $\{15^\circ - 60^\circ\}$, $\{30^\circ - 45^\circ\}$, and $\{30^\circ - 60^\circ\}$.

As in section 5.6.4, for each pair group, we use all the relevant UCL Multi-Pose images at the two poses to train Tied models; In test, for each of 10 test folds, we compute % correct of the image pairs belonging to the test pair group, then we report the final performance by the mean of % correct in 10 test folds and the standard error of the mean.

For all the three Tied Bayesian Face algorithms we apply PCA to reduce the dimensions. For Tied PLDA and Tied Joint PLDA we set the PCA dimensions to 200, subspace dimensions to 128. For the Tied Joint Bayesian Face algorithm we set the PCA dimensions to 200. The optimal settings are obtained by an empirical approach.

In Table 5.6 we compare the performance of three Bayesian face recognition algorithms trained in the LFW database with the performance of three Tied Bayesian face recognition algorithms trained in the UCL Multi-Pose database for each pair group. The way to train and test Bayesian face recognition

algorithms for each pair group in the LFW database is described in section 5.2. From Table 5.6 we find that Tied Bayesian face recognition algorithms perform better than Bayesian face recognition algorithms for non-frontal pair groups with large pose differences. Examples include $\{15^\circ - 45^\circ\}$, $\{30^\circ - 60^\circ\}$, $\{0^\circ - 45^\circ\}$, $\{15^\circ - 60^\circ\}$, and $\{0^\circ - 60^\circ\}$. The experiment results confirm our assumption that Tied Bayesian face recognition algorithms can increase performance when large pose variation exists. We also noticed that Bayesian face recognition algorithms perform better than Tied Bayesian face recognition algorithms for near frontal pair groups, in which both two images are near frontal. Examples include $\{0^\circ - 15^\circ\}$, $\{15^\circ - 30^\circ\}$, and $\{0^\circ - 30^\circ\}$. The reason might be that our UCL Multi-Pose database provides less frontal training images than the LFW database.

From Table 5.6 we also find that the Tied Joint Bayesian Face algorithm and Tied Joint PLDA produce better performance than Tied PLDA. The reason might be that estimating the covariance matrix without making low dimension assumption can capture more discriminatory information as we concluded in chapter 4.

Pose difference	Pair groups	Bayesian face recognition algorithms trained and tested in the LFW database			Tied Bayesian face recognition algorithms trained in the UCL Multi-Poses database and tested in the LFW database		
		PLDA	Joint Bayesian Face	Joint PLDA	Tied PLDA	Tied Joint Bayesian Face	Tied Joint PLDA
15	0-15	90.029 ±0.873	89.231 ±0.782	89.492 ±0.664	81.810 ±0.755	83.163 ±1.508	83.646 ±0.907
	15-30	87.048 ±1.223	87.362 ±1.202	87.656 ±1.084	82.961 ±1.610	85.815 ±0.908	85.481 ±1.008
	30-45	70.433 ±5.233	73.167 ±6.093	76.944 ±4.672	79.448 ±5.651	87.028 ±3.715	86.671 ±3.858
30	0-30	84.256 ±1.103	86.487 ±0.887	86.629 ±0.793	81.428 ±1.369	84.318 ±1.309	84.468 ±1.791
	15-45	82.662 ±2.985	77.765 ±3.000	81.523 ±3.614	79.848 ±4.452	85.126 ±3.260	83.645 ±4.295
	30-60	51.667 ±15.000	56.167 ±15.176	66.667 ±14.907	77.381 ±13.934	76.190 ±11.419	88.095 ±7.897
45	0-45	76.916 ±6.943	75.566 ±5.804	75.197 ±5.552	79.204 ±4.493	79.812 ±2.437	76.669 ±2.641
	15-60	38.333 ±14.498	38.333 ±14.498	38.333 ±14.498	77.381 ±13.934	75.000 ±11.180	80.556 ±9.044
60	0-60	51.071 ±12.644	61.071 ±12.099	61.071 ±12.099	67.143 ±12.919	63.571 ±12.664	72.143 ±10.830

Table 5.6: **Verification results when trained in the UCL Multi-Pose database and tested in the LFW database.** We compare the performance of Bayesian face recognition algorithms trained in the LFW database with the performance of Tied Bayesian face recognition algorithm trained in the UCL Multi-Pose database. The tests are all conducted in the LFW database. We find that Tied Bayesian face recognition algorithms perform better when large pose differences exist. Note:there is no result for pair group $\{45^\circ - 60^\circ\}$ because no image pairs exist in that pair group.

We compare Table 5.6 with Table 5.5 and find that the performance of Tied Bayesian face recognition algorithms trained in the UCL Multi-Pose database is better than the results trained in the Multi-PIE

database. These results support our hypothesis that it is better to use images from uncontrolled database in training for a uncontrolled test database.

5.6.6 Switching Mechanism

From Table 5.6 we find that Bayesian face recognition algorithms perform better than Tied Bayesian face recognition algorithms for near-frontal pairs. However, Tied Bayesian face recognition algorithms perform better when large pose differences exist. Therefore, we make a conjecture: the best performance is achieved if we switch between Bayesian face recognition algorithms and Tied Bayesian face recognition algorithms based on pose difference of a image pair.

We apply a simple switching model as follows. We apply Tied Bayesian face recognition algorithms to make match decision for the following pair groups: $\{0^\circ - 45^\circ\}$, $\{0^\circ - 60^\circ\}$, $\{15^\circ - 45^\circ\}$, $\{15^\circ - 60^\circ\}$, $\{30^\circ - 45^\circ\}$, and $\{30^\circ - 60^\circ\}$. We apply Bayesian face recognition algorithms for other pair groups. The matching decision is assigned by a switching model:

$$D = \begin{cases} D_{Tied} & \{0^\circ - 45^\circ\}, \{0^\circ - 60^\circ\}, \{15^\circ - 45^\circ\}, \{15^\circ - 60^\circ\}, \{30^\circ - 45^\circ\}, \{30^\circ - 60^\circ\} \\ D_{Bayesian} & \text{Otherwise} \end{cases} \quad (5.60)$$

In Table 5.7 we compare the performance of Bayesian face recognition algorithms in the LFW database with the performance of applying a switching mechanism to combine the advantages of Bayesian face recognition algorithms and Tied Bayesian face recognition algorithms. Here three Bayesian face recognition algorithms are trained using LBP descriptors extracted from the LFW images as in section 5.2; three Tied Bayesian face recognition algorithms are trained using LBP descriptors extracted from the UCL Multi-Pose database as in section 5.6.5.

From Table 5.7 we find that the switching mechanism can improve the verification performance in the LFW database although the improvement is slight. The reason might be that there are not many test pairs with large pose differences in the LFW database, although Tied Bayesian face recognition algorithms improve the performance for pair groups with large pose variation.

Algorithm	% correct	Algorithm	% correct	Algorithm	% correct
PLDA	87.350 ± 0.433	Joint Bayesian Face	87.617 ± 0.512	Joint PLDA	88.000 ± 0.442
Switching between PLDA and Tied PLDA	87.583 ± 0.383	Switching between Joint Bayesian Face and Tied Joint Bayesian Face	87.821 ± 0.457	Switching between Joint PLDA and Tied Joint PLDA	88.167 ± 0.453

Table 5.7: **The effect of the switching mechanism.** We compare the performance of Bayesian face recognition algorithms in the LFW database with the performance of using a switching Mechanism to combine advantages of Bayesian face recognition algorithms and Tied Bayesian face recognition algorithms. The switching Mechanism improves the performance.

5.7 Conclusion

In this chapter we demonstrated that large pose variation is the challenge for Bayesian face recognition algorithms (PLDA, the Bayesian Face algorithm, and Joint PLDA). To address this issue, we proposed two new algorithms: the Tied Joint Bayesian Face algorithm and Tied Joint PLDA. To train tied models, sufficient training images are required for each pose. However, the LFW database cannot satisfy this requirement. We introduced the UCL Multi-Pose database to solve this problem.

We first compare Tied PLDA, the Tied Joint Bayesian Face algorithm, and Tied Joint PLDA in the controlled Multi-PIE database and the uncontrolled UCL Multi-Pose database respectively. Then we conduct two cross-database experiments: trained in the Multi-PIE database and tested in the LFW database; trained in the UCL Multi-Pose database and tested in the LFW database. Our experimental results show that the performance of the three Tied Bayesian face recognition algorithms trained in the uncontrolled UCL Multi-Pose database is better than the performance trained in the controlled Multi-PIE database. Our experiment results also demonstrated that Tied Bayesian face recognition algorithms improve the performance for pair groups with large pose difference. Among the three Tied algorithms, we find that Tied Joint PLDA performs best. However, for near-frontal pairs, Bayesian face recognition algorithms perform better than Tied Bayesian face recognition algorithms.

To combine the advantages of Bayesian face recognition algorithms and Tied Bayesian face recognition algorithms, we introduced a switching mechanism: we apply Tied Bayesian face recognition algorithms for pair groups with large pose differences and apply Bayesian face recognition algorithms for other pair groups. Our experimental results show that the switching mechanism improves performance in the LFW database.

Our algorithm has connections to the learned Bayesian Face algorithm [89], which applies Manifold Relevance Determination [40] to learn the identity subspace. The commonality between their model and our tied models is that these algorithms are all Bayesian models and the match assignment for two images is decided by comparing the match likelihood and non-match likelihood. The difference is that we apply Gaussian latent variable models while they use Gaussian Process latent variable models. They demonstrated that their Gaussian Process based model is flexible to fit complex data and improve the verification performance in uncontrolled environments. The performance might be improved if we combine the learned Bayesian Face algorithm and our tied models.

Currently we only use our tied model to deal with horizontal pose variation. The tied models can be also used to handle vertical pose variation and lighting variation.

Chapter 6

Conclusion

In this report, we proposed a series of probabilistic face recognition algorithms to improve recognition performance in uncontrolled environments. The motivation for these algorithms is based on two limitations of the existing algorithms: (i) many algorithms do not perform well in uncontrolled environments; (ii) most existing algorithms cannot handle large pose variation in uncontrolled environments. To overcome the first limitation, we proposed Multi-Scale PLDA and Joint PLDA and show that they improve performance in the benchmark database of face recognition under uncontrolled environments: the Labeled Faces in the Wild database [65]. To resolve the second limitation, we collected a new database and proposed the Tied Joint Bayesian Face algorithm and Tied PLDA.

In this chapter, we will firstly summarize our findings in each chapter in section 6.1. Then we discuss limitations and future work in section 6.2.

6.1 Summary and Contributions

In chapter 3, we proposed Multi-Scale PLDA to combine patch-based face representation methods and Probabilistic Linear Discriminant Analysis (PLDA) [111]. In Multi-Scale PLDA, face images are described as a sum of the signal component and the noise component. The signal component describes the between-individual variation and is a weighted combination of the basis functions of the between-individual subspace. The noise component explains the within-individual variation and is a weighted combination of the basis functions of the within-individual subspace. We break both the signal and noise into regular grids of non-overlapping patches. We change the patch configuration of the signal component to vary the spatial support of the basis functions of the between-individual subspace. We change the grid resolution of the noise component to vary the degree of the localization of the basis functions of the within-individual subspace.

We applied Multi-scale PLDA in four controlled databases and one uncontrolled database. We find that we can obtain the best performance in three constrained databases when the signal component of Multi-scale PLDA is treated locally and the noise component is treated globally. We achieved 100% correct performance for face identification in the XM2VTS frontal database [95] using an optimal combination of local signal and global noise models, which is a significant improvement compared to 91% correct of PLDA and 84% correct of Dual Space LDA [137]. However, performance did not increase in

the fourth constrained database when we treat the signal more locally. We attributed this difference to the pose changes that are not present in the three controlled databases but are present in the fourth database. If there are pose changes between two images, the corresponding facial features will not appear in corresponding patches. We also applied Multi-Scale PLDA in the uncontrolled database: the LFW database. Since the unconstrained face database contains large pose variation, Multi-scale PLDA does not perform well when intensities are used to represent images.

The main disadvantage of Multi-Scale PLDA is that it is sensitive to pose variation. We hope to address this problem by extending the shiftmap representation [113] to estimate the corresponding patches for two images with different poses in future work. Another disadvantage of Multi-Scale PLDA is that the training process is slower than PLDA since it applies more basis functions and requires more calculation.

In chapter 4 we proposed Joint PLDA to combine the advantages of PLDA and the Joint Bayesian Face algorithm [30]. PLDA and the Joint Bayesian Face algorithm are two state of the art algorithms and produce a good performance in the LFW database. The advantage of PLDA is that it uses an EM training method to estimate model parameters and guarantees that the likelihood increases at each iteration. The disadvantage of PLDA is that it uses a subspace method to project high dimensional face data into a low dimensional subspace and may discard some discriminative information. The advantage of the Joint Bayesian Face algorithm is that it does not make the low dimension assumption and can estimate the match and non-match covariance matrix from high dimensional data directly. Its disadvantage is that it uses an EM-like algorithm and cannot guarantee that the likelihood increases at each iteration. We proposed Joint PLDA to combine the two algorithms. Joint PLDA uses a strict EM algorithm to guarantee likelihood increases and can also estimate the covariance matrix from the high dimensional data directly.

Our experiments show that Joint PLDA always produces better performance than PLDA and the Joint Bayesian Face algorithm in the LFW database when a single descriptor is used. When we combine four image descriptors, Joint PLDA can achieve $91.367\% \pm 0.448$ in the LFW database, which is comparable to $91.300\% \pm 0.003$ of the commercial face recognition system face.com [126].

One drawback of Joint PLDA is that it requires more computation cost to do face verification than PLDA because it does not make low dimensional assumption. Moreover, Joint PLDA cannot pre-process images offline as PLDA, so the speed of Joint PLDA to do face identification is slower than PLDA.

In chapter 5 we proposed the Tied Joint Bayesian Face algorithm and Tied Joint PLDA to handle large pose variation in uncontrolled environments. We assign each LFW image to one of a set of pre-defined horizontal pose categories and allocate each LFW test image pair to one of a set of pair groups based on the poses of the two images. We analyse the verification performance of PLDA and the Joint Bayesian Face algorithm for each pair group and find that both algorithms perform very badly for pair groups with large pose variation. To handle the pose changes in the LFW database, we attempt to use Tied PLDA [82], which has been demonstrated to be able to handle pose variation in controlled databases well. We also proposed Tied Joint Bayesian Face algorithm and Tied Joint PLDA to address the issue.

We refer to Tied PLDA, Tied Joint Bayesian Face algorithm and Tied Joint PLDA as Tied Bayesian face recognition algorithms. We refer to PLDA, the Joint Bayesian Face algorithm and Joint PLDA as Bayesian face recognition algorithms.

To train Tied Bayesian face recognition algorithms, sufficient training images are required for each pair group but the LFW database cannot provide that. To have sufficient training images, we used the Multi-PIE database and also introduced our own UCL Multi-Pose database as training datasets. When we train the three Tied Bayesian face recognition algorithms in the Multi-PIE database and test in the LFW database, we find that the performance of the Tied Bayesian algorithms is worse than the performance of the Bayesian face recognition algorithms, which are trained only using the LFW images. The reason might be that the images of the controlled Multi-PIE database do not include an equivalent amount of image variation as the uncontrolled LFW test images. When we train the Tied Bayesian face recognition algorithms in the UCL Multi-Pose database and test in the LFW database, our experiments show that the performance of the Tied Bayesian face recognition algorithms is better than the performance of the Bayesian face recognition algorithms for pair groups with large pose variation. Among the three Tied Bayesian face recognition algorithms, Tied Joint PLDA performs best. However, the Bayesian face recognition algorithms perform better than the Tied Bayesian face recognition algorithms for near frontal pair groups. To combine the advantages of Tied Bayesian face recognition algorithms and Bayesian face recognition algorithms, we proposed a switching mechanism to apply different algorithms based on the poses of the two images. Our experiments show that the switching mechanism can improve performance in the LFW database.

Currently we have only applied tied models to handle horizontal pose variation. In the future we can extend the application of tied model to deal with vertical pose variation and lighting variation.

6.2 Limitations and Future Work

Gaussian Model

In this report we assumed that the marginal density of the data is a multivariate Gaussian. The drawback of this assumption is that our models might be sensitive to outliers in the training images. To address this limitation, we will propose a more robust probability model in the future, in which we assume the marginal density of the data is distributed as a multivariate t-distribution. Compared with a Gaussian, a t-distribution has heavier tails, which helps improving the robustness to outliers as demonstrated in [74]. The t-distributed models will be more general than the Gaussian models. In fact, a t-distribution is equivalent to a Gaussian when the degree of freedom approaches infinity. The Gaussian models can be treated as the special case of the t-distributed models. We have great interest to transfer the models of this thesis to the version using t-distribution and investigate whether the new models improve the robustness to image variations in uncontrolled environments. One drawback of the t-distributed models is that the training will be slower than the Gaussian models since more computation will be required to find an optimal value for the degree of freedom.

Large Training Data

Along with the development of image search engines and social networks, it has become easier and

easier to collect a large number of training images. We are interested to investigate the performance if we use more training images. Recently, Taigman et al. [128] have demonstrated that the neural network is a good model to leverage the huge volume of training data. The deep and large networks can be learned effectively from big training data and can be applied to form a very compact representation to face images. The process to learn deep and large networks is termed deep learning. Taigman et al. used deep learning to extract features from images and applied a simple classifier to make the verification assignment. In the future we will use the deep learning method to extract features from images and use the proposed algorithms in this report to do face verification in the LFW database. We hypothesize this combination will improve performance compared to [128]. However, deep learning has its disadvantages: it requires a very large amount of memory to store the networks and the computation cost to recognize an image is very expensive. It might take several seconds to identify a face image. Therefore, it might not be suitable for some mobile devices which have limited memory and low computation capability.

Image Descriptors

In this report we used global image descriptors to do face verification in the LFW database. Global image descriptors denote that we extract visual features from the whole image. In future work we hope to investigate the performance of all our algorithms using local descriptors, which means we extract visual features from fiducial points (they are a set of salient facial parts and usually locate on the corners of the eyebrows, the corners of the eyes, the tip of the nose, the corners of the lips, etc). It has been demonstrated in [31] [112] that local image descriptors are more robust to pose variation and help improve the recognition performance. Another advantage of local image descriptors is that it is easy to form a dense representation of the face image by increasing the number of fiducial points and collecting descriptors from a pyramid of patches with different size over a fiducial point. This type of dense representation has been demonstrated in [31] [124] and shown to capture more discriminative information and improve the performance. Currently, we do not have a good fiducial points detector, so we did not use a dense representation to encode the LFW images. In the future we will use high dimensional data vectors obtained from dense representation methods to conduct verification experiments in the LFW database.

Glossary

Notations

Υ	A face image represented by a pixel intensity array
\mathbf{x}	A face image represented by a pixel intensity vector obtained by concatenating the columns of pixels in the image Υ
$\boldsymbol{\mu}$	The mean vector of all the training images
Φ	A matrix containing the basis functions of the Eigenfaces subspace in its columns
$\boldsymbol{\omega}$	A coefficient vector in the Eigenfaces subspace
\mathbf{x}_{ijk}	The k^{th} pose of the j^{th} image of the i^{th} individual
$\boldsymbol{\mu}_k$	The mean vector at pose k
\mathbf{F}	A matrix containing the basis functions of the between-individual subspace in its columns
\mathbf{F}_k	A matrix containing the basis functions of the between-individual subspace for pose k in its columns
\mathbf{F}^p	A matrix containing the basis functions of the between-individual subspace for the p^{th} patch of images in its columns
\mathbf{G}	A matrix containing the basis functions of the within-individual subspace in its columns
\mathbf{G}_k	A matrix containing the basis functions of the within-individual subspace for pose k in its column
\mathbf{G}^q	A matrix containing the basis functions of the within-individual subspace for the q^{th} patch of images in its column
\mathbf{h}_i	A hidden identity variable for all the images of the i^{th} individual
\mathbf{h}_i^p	A hidden identity variable for the p^{th} patch of all the images of the i^{th} individual
\mathbf{w}_{ij}	A hidden noise variable for the j^{th} image of the i^{th} individual
\mathbf{w}_{ij}^q	A hidden noise variable for the q^{th} patch of the j^{th} image of the i^{th} individual
ϵ_{ij}	A stochastic noise of the j^{th} image of the i^{th} individual
ϵ_{ijk}	A stochastic noise of the j^{th} image of the i^{th} individual at pose k
Σ	The diagonal covariance matrix for the stochastic noise of images
Σ_k	The diagonal covariance matrix for the stochastic noise of images at pose k
θ	The model parameters
\mathbf{I}	The identity matrix
P	The patch number for the signal component
Q	The patch number for the noise component

P_{IR}	The identification rate
P_{FA}	The false alarm rate
τ	The threshold
η	The similarity score of two images
S	The scatter matrix
S_B	The scatter matrix for the between-individual variation
S_W	The scatter matrix for the within-individual variation
\hat{W}	A matrix containing the basis functions of the Fisherfaces subspace in its columns
p_i	The percentage of correct assignment for the test group i of the LFW database
$\hat{\mu}$	The mean accuracy
S_E	The standard error of the mean
$\hat{\sigma}$	The estimate of the standard deviation
κ	A constant
Δ	The difference of two images
\mathcal{M}_s	The model two images match
\mathcal{M}_d	The model two images do not match
Σ_d	The non-match covariance matrix
Σ_s	The match covariance matrix
Λ_s	The eigenvalues of the within-individual covariance matrix
V_s	The eigenvectors of the within-individual covariance matrix
Λ_d	The eigenvalues of the between-individual covariance matrix
V_d	The eigenvectors of the between-individual covariance matrix
α	The identity component of a face image
β	The within-individual variation component of a face image
Σ_α	The covariance matrix for the between-individual variation
Σ_β	The covariance matrix for the within-individual variation
$\Sigma_{\alpha 1}$	The covariance matrix for the between-individual variation at pose 1
$\Sigma_{\beta 1}$	The covariance matrix for the within-individual variation at pose 1
$\Sigma_{\alpha 2}$	The covariance matrix for the between-individual variation at pose 2
$\Sigma_{\beta 2}$	The covariance matrix for the within-individual variation at pose 2
$\Sigma_{\alpha 12}$	The covariance matrix for the between-individual variation across pose 1 and pose 2
α_a	The identity component of training images at pose 1
α_b	The identity component of training images at pose 2
β_a	The within-individual variation component of training images at pose 1
β_b	The within-individual variation component of training images at pose 2
$\mathcal{G}_o[\varrho, \varsigma]$	A Gaussian in \mathcal{O} with mean ϱ and covariance ς
$r(\mathbf{x}_1, \mathbf{x}_2)$	The log likelihood ratio of two images \mathbf{x}_1 and \mathbf{x}_2

Acronyms

CCD	Charge-coupled Device
MAP	Maximum a Posteriori
EM	Expectation maximization
NC	Nearest Centroid
NN	Nearest Neighbors
PCA	Principal Component Analysis
LFA	Local Feature Analysis
EBGM	Elastic Bunch Graph Matching
LDA	Linear Discriminant Analysis
ASM	Active Shape Model
AAM	Active Appearance Model
SVM	Support Vector Machine
PLDA	Probabilistic Linear Discriminant Analysis
SLDA	Smooth Linear Discriminant Analysis
OSS	One-Shot Similarity
LDML	Logistic Discriminant Base Metric Learning
MKNN	Marginalized k-nearest-neighbour
LLDA	Locally Linear Discriminant Analysis
CSML	Cosine Similarity Metric Learning
DML-EIG	Distance Metric Learning with Eigenvalue Optimization
CMD	Covariance Matrix Descriptors
SUB-SML	Similarity Metric Learning over the Intra-personal Subspace
LBP	Local Binary Patterns
TPLBP	Three-Patch LBP
FPLBP	Four-Patch LBP
LE	Learning-based
LARK	Locally Adaptive Regression Kernel
LQP	Local Quantized Patterns
SIFT	Scale Invariant Feature Transform
HOG	Histogram of Oriented Gradients
OCLBP	Over-Complete Local Binary Patterns
FERET	Face Recognition Technology Test
FRVT	Face Recognition Vendor Test
ORL	Olivetti Research Ltd
AR	Alex Martghnez and Robert Benavente
PIE	Pose, Illumination, and Expression
XM2VTS	Extended Multi Modal Verification for Teleservices and Security applications

KFDB	Korean Face Database
FRGC	Face Recognition Grand Challenge
WDRef	Wide and Deep Reference
PUBFIG	Public Figure
LFW	Labeled Faces in the Wild
ROC	Receiver Operator Characteristic
CMC	Cumulative Match Characteristic

Bibliography

- [1] Facial recognition technology safeguards beijing olympics. *Bulletin of the Chinese Academy of Sciences*, 3:131–132, 2008.
- [2] Y. Adini, Y. Moses, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:721–732, 1997.
- [3] J. Aghajanian and S. Prince. Mosaicfaces: a discrete representation for face recognition. In *Workshop on Applications of Computer Vision*, 2008.
- [4] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision*, pages 469–481. 2004.
- [5] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.
- [6] S. R. Arashloo, J. Kittler, and W. J. Christmas. Pose-invariant face recognition by matching on multi-resolution mrfs linked by supercoupling transform. *Computer Vision and Image Understanding*, 115(7):1073–1083, 2011.
- [7] A. B. Ashraf, S. Lucey, and T. Chen. Learning patch correspondences for improved viewpoint invariant face recognition. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [8] A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3d pose normalization. In *International Conference on Computer Vision*, pages 937–944, 2011.
- [9] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz. Fast high dimensional vector multiplication face recognition. In *International Conference on Computer Vision*, pages 1960–1967, 2013.
- [10] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linearprojection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.
- [11] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Computer Vision and Pattern Recognition*, pages 545–552, 2011.
- [12] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14:585–591, 2001.
- [13] T. Berg and P. N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *British Machine Vision Conference*, volume 1, page 5, 2012.
- [14] D. J. Beymer. Face recognition under varying pose. In *Computer Vision and Pattern Recognition*, pages 756–761, 1994.
- [15] V. Blanz, P. Grother, P. J. Phillips, and T. Vetter. Face recognition based on frontal views generated from non-frontal images. In *Computer Vision and Pattern Recognition*, volume 2, pages 454–461, 2005.

- [16] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Computer graphics and interactive techniques*, pages 187–194, 1999.
- [17] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.
- [18] H. Bon-Woo, H. Byun, R. Myoung-Cheol, and L. Seong-Whan. Performance evaluation of face recognition algorithms on the asian face database, kfdb. In *Audio-and Video-Based Biometric Person Authentication*, pages 557–565, 2003.
- [19] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, 2006.
- [20] R. Brunelli and T. Poggio. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
- [21] M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistics. In *First International Workshop in Automatic Face and Gesture Recognition*, pages 154–159, 1995.
- [22] D. Cai, X. He, J. Han, and H.-J. Zhang. Orthogonal laplacianfaces for face recognition. *Image Processing*, 15(11):3608–3614, 2006.
- [23] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning a spatially smooth subspace for face recognition. In *Computer Vision and Pattern Recognition*, pages 1–7, 2007.
- [24] P. Campadelli, R. Lanzarotti, and G. Lipori. Automatic facial feature extraction for face recognition. *Face Recognition*, pages 31–58, 2007.
- [25] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.
- [26] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Computer Vision and Pattern Recognition*, pages 2707–2714, 2010.
- [27] X. Chai, S. Shan, X. Chen, and W. Gao. Locally linear regression for pose-invariant face recognition. *Image Processing, IEEE Transactions on*, 16(7):1716–1725, 2007.
- [28] H. Chan and W. Bledsoe. A man-machine facial recognition system: some preliminary results. *Panoramic Reserch Inc.*, 1965.
- [29] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.
- [30] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision*, pages 566–579, 2012.
- [31] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Computer Vision and Pattern Recognition*, pages 3025–3032, 2013.
- [32] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.
- [33] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61:38–59, 1995.
- [34] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685, 2001.
- [35] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002.

- [36] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [37] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.
- [39] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, pages 428–441, 2006.
- [40] A. Damianou, C. Ek, M. Titsias, and N. Lawrence. Manifold relevance determination. *arXiv preprint arXiv:1206.4610*, 2012.
- [41] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(7):1169–1179, 1988.
- [42] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Machine learning*, pages 209–216. ACM, 2007.
- [43] A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.
- [44] L. Ding and A. M. Martinez. Precise detailed detection of faces and facial features. In *Computer Vision and Pattern Recognition*, pages 1–7, 2008.
- [45] M. Eckhardt, I. Fasel, and J. Movellan. Towards practical facial feature detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(03):379–400, 2009.
- [46] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition*, pages 300–305, 1998.
- [47] R. A. Facial, J. Phillips, A. Martin, and C. L. Wilson. An introduction to evaluating biometric systems. *National Institute of Standards and Technology*, 33(2):56–63, 2000.
- [48] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. *arXiv preprint arXiv:1403.2802*, 2014.
- [49] Y. Fu and S. Prince. Investigating the spatial support of signal and noise in face recognition. In *ICCV workshop on subspace methods*, pages 131–138, 2009.
- [50] K. Fukui and O. Yamaguchi. Face recognition using multi-viewpoint patterns for robot vision. In *Robotics Research*, pages 192–201. Springer, 2005.
- [51] W. Gao, B. Cao, S. Shan, et al. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38, 2008.
- [52] A. S. Georghiadis, P. N. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.
- [53] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.
- [54] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449–465, 2002.
- [55] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):449–465, 2004.
- [56] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

- [57] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*, pages 413–426, 2008.
- [58] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision*, pages 498–505, 2009.
- [59] G. Guo, S. Z. Li, and K. L. Chan. Face recognition by support vector machines. In *Automatic Face and Gesture Recognition*, pages 196–201, 2000.
- [60] T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 155–176, 1996.
- [61] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.
- [62] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.
- [63] G. B. Huang. Face verification results in lfw. <http://vis-www.cs.umass.edu/lfw/results.html>.
- [64] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [65] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [66] S. U. Hussain, T. Napoléon, F. Jurie, et al. Face recognition using local quantized patterns. In *British Machine Vision Conference*, 2012.
- [67] J. Ilonen, J.-K. Kamarainen, P. Paalanen, M. Hamouz, J. Kittler, and H. Kalviainen. Image feature localization by multiple hypothesis testing of gabor features. *IEEE Transactions on Image Processing*, 17(3):311–325, 2008.
- [68] S. Ioffe. Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542, 2006.
- [69] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In *Audio- and video-based biometric person authentication*, pages 90–95, 2001.
- [70] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):640–649, 2007.
- [71] T. Kanade. *computer recognition of human faces*. PhD thesis, 1973.
- [72] T. Kanade and A. Yamada. Multi-subregion based probabilistic approach toward pose-invariant face recognition. In *Computational Intelligence in Robotics and Automation*, volume 2, pages 954–959, 2003.
- [73] M. D. Kelly. *Visual identification of people by computer*. PhD thesis, 1971.
- [74] Z. Khan and F. Dellaert. Robust generative subspace modeling: The subspace t distribution. *Georgia Institute of Technology*, 2004.
- [75] O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. In *Similarity-Based Pattern Recognition*, pages 31–45. Springer, 2011.
- [76] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *International Conference on Computer Vision*, pages 365–372, 2010.

- [77] A. Lanitis, C. Taylor, and T. Cootes. An automatic face identification system using flexible appearance models. In *British Machine Vision Conference*, pages 65–74, 1994.
- [78] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.
- [79] E. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.
- [80] M. D. Levine and Y. Yu. Face recognition subject to variations in facial expression, illumination and pose using correlation filters. *Computer Vision and Image Understanding*, 104(1):1–15, 2006.
- [81] A. Li, S. Shan, X. Chen, and W. Gao. Maximizing intra-individual correlations for face recognition across pose differences. In *Computer Vision and Pattern Recognition*, pages 605–611, 2009.
- [82] P. Li, Y. Fu, U. Mohammed, J. Elder, and S. Prince. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, 2012.
- [83] Y. Li, S. Gong, and H. Liddell. Constructing facial identity surfaces in a nonlinear discriminating space. In *Computer Vision and Pattern Recognition*, volume 2, pages 258–265, 1993.
- [84] H.-C. Lian and B.-L. Lu. Multi-view gender classification using local binary patterns and support vector machines. In *Advances in Neural Networks*, pages 202–209. 2006.
- [85] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. In *European Conference on Computer Vision*, pages 72–85. 2008.
- [86] C. Liu. Gabor-based kernel pca with fractional power polynomial models for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):572–581, 2004.
- [87] E. LOCOCO. Airport switches face-recognition systems over accuracy concerns. *Bloomberg News*, May 2002.
- [88] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [89] C. Lu and X. Tang. Learning the face prior for bayesian face recognition. In *European Conference on Computer Vision*, pages 119–134. Springer, 2014.
- [90] S. Lucey and T. Chen. Learning patch dependencies for improved pose mismatched face verification. In *Computer Vision and Pattern Recognition*, volume 1, pages 17–22, 2006.
- [91] W. Ma and H. Zhang. An indexing and browsing system for home video. In *European Signal Processing Conference*, pages 5–8, 2000.
- [92] A. M. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.
- [93] A. M. Martinez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:748–763, 2002.
- [94] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.
- [95] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, 1999.
- [96] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *European Conference on Computer Vision*, pages 504–513. 2008.
- [97] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, 2000.

- [98] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:696–710, 1997.
- [99] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Asian Conference on Computer Vision*, pages 709–720, 2010.
- [100] M. Nishiyama, T. Kozakaya, and O. Yamaguchi. Illumination normalization using quotient image-based techniques. *Recent Advances in Face Recognition*, pages 97–108, 2008.
- [101] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [102] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
- [103] P. Penev and J. Atick. Local feature analysis: a general statistical theory for object representation. *Network: computation in neural systems*, 7(3):477–500, 1996.
- [104] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition*, pages 84–91, 1994.
- [105] P. Phillips, W. Scruggs, A. OToole, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. *National Institute of Standards and Technology*, 7408, 2007.
- [106] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.
- [107] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition*, volume 1, pages 947–954, 2005.
- [108] P. J. Phillips, P. Grother, and R. Micheals. *Evaluation methods in face recognition*. Springer, 2011.
- [109] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone. Face recognition vendor test 2002. In *Analysis and Modeling of Faces and Gestures*, page 44. IEEE, 2003.
- [110] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.
- [111] S. Prince and J. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [112] S. Prince, J. Warrell, J. Elder, and F. Felisberti. Tied factor analysis for face recognition across large pose differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):970–984, 2008.
- [113] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *International Conference on Computer Vision*, pages 151–158, 2009.
- [114] R. Ramamoorthi. Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1322–1333, 2002.
- [115] S. A. Rizvi, P. J. Phillips, and H. Moon. A verification protocol and statistical performance analysis for face recognition algorithms. In *Computer Vision and Pattern Recognition*, pages 833–838, 1998.
- [116] S. A. Rizvi, P. J. Phillips, and H. Moon. A verification protocol and statistical performance analysis for face recognition algorithms. In *Computer Vision and Pattern Recognition*, pages 833–838, 1998.
- [117] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *European Conference on Computer Vision*, pages 3–19. 2002.

- [118] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
- [119] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *the Second IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.
- [120] J. M. Saragih, S. Lucey, and J. F. Cohn. Face alignment through subspace constrained mean-shifts. In *Computer Vision*, pages 1034–1041, 2009.
- [121] H. J. Seo and P. Milanfar. Face verification using the lark representation. *IEEE Transactions on Information Forensics and Security*, 6(4):1275–1286, 2011.
- [122] A. Shashua and T. Riklin-Raviv. The quotient image: class-based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:129–139, 2001.
- [123] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.
- [124] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, volume 1, page 7, 2013.
- [125] H. W. Stan, H. Wang, S. Z. Li, and Y. Wang. Generalized quotient image. In *Computer Vision and Pattern Recognition*, pages 498–505, 2004.
- [126] Y. Taigman and L. Wolf. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition. *arXiv preprint arXiv:1108.1122*, 2011.
- [127] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference*, pages 1–12, 2009.
- [128] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deep-face: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition*, 2014.
- [129] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Lecture Notes in Computer Science*, 4778:168, 2007.
- [130] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.
- [131] A. T. Tokuhiko and B. J. Vaughn. Initial test results of the omron face cue entry system at the university of missouri-rolla reactor. *Journal of Nuclear Science and Technology*, 41(4):502–510, 2004.
- [132] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.
- [133] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition*, page 511, 2001.
- [134] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [135] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1692–1698, 2005.
- [136] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma. Toward a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):372–386, 2012.
- [137] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *Computer Vision and Pattern Recognition*, pages 564–569, 2004.

- [138] A. H. Watt and F. Policarpo. *The Computer Image*. Addison Wesley, 1998.
- [139] L. Wiskott, J.-M. Fellous, N. Krger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997.
- [140] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in European Conference on Computer Vision*, volume 6, 2008.
- [141] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.
- [142] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.
- [143] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. J. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*, 16:212–220, 2007.
- [144] J. Yang, A. F. Frangi, J.-y. Yang, D. Zhang, and Z. Jin. Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):230–244, 2005.
- [145] 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 Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.
- [146] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *Computer Vision and Pattern Recognition*, pages 497–504, 2011.
- [147] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Computer Vision and Pattern Recognition*, page 158, 1997.
- [148] C. Zhan, W. Li, P. Ogunbona, and F. Safaei. Real-time facial feature point extraction. In *Advances in Multimedia Information Processing*, pages 88–97. 2007.
- [149] B. Zhang, Y. Gao, S. Zhao, and J. Liu. Local derivative pattern versus local binary pattern face recognition with high order local pattern descriptor. *Image Processing*, 19(2):533–544, 2010.
- [150] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 2002.
- [151] X. Zhang, Y. Gao, and M. K. Leung. Recognizing rotated faces from frontal and side views: An approach toward effective use of mugshot databases. *IEEE Transactions on Information Forensics and Security*, 3(4):684–697, 2008.
- [152] W. Zhao and R. Chellappa. SFS based view synthesis for robust face recognition. In *Automatic Face and Gesture Recognition*, pages 285–292, 2000.
- [153] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE, 2012.